

Análisis de datos de microarrays

M. Carme Ruíz de Villa
Alex Sánchez-Pla

PID_00192743

Ninguna parte de esta publicación, incluido el diseño general y la cubierta, puede ser copiada, reproducida, almacenada o transmitida de ninguna forma, ni por ningún medio, sea éste eléctrico, químico, mecánico, óptico, grabación, fotocopia, o cualquier otro, sin la previa autorización escrita de los titulares del copyright.

Índice

1. El proceso de análisis de datos de microarrays.....	7
1.1. Introducción	7
1.2. Tipos de estudios	7
1.2.1. Comparación de grupos o <i>class comparison</i>	7
1.2.2. Predicción de clase	8
1.2.3. Descubrimiento de clases	8
1.2.4. Otros tipos de estudios	9
1.3. Algunos ejemplos concretos	9
1.3.1. Estudio de procesos regulados por citoquinas	9
1.3.2. Clasificación molecular de la leucemia	10
1.3.3. Efecto del estrógeno y el tiempo de administración	10
1.3.4. Efecto del CCL4 en la expresión génica	10
1.3.5. Análisis de patrones en el ciclo celular	11
1.3.6. Recapitulación	11
1.4. El proceso de análisis de microarrays	12
2. Diseño de experimentos de microarrays.....	15
2.1. Fuentes de variabilidad	15
2.2. Algunas definiciones básicas	16
2.3. Principios básicos en el diseño del experimento	17
2.3.1. Replicación	17
2.3.2. Aleatorización	20
2.3.3. Bloquear	20
2.4. Diseños experimentales para microarrays de dos colores	21
3. Exploración de los datos, control de calidad y preprocesado	23
3.1. Introducción	23
3.1.1. Nivel de análisis y tipo de microarray	24
3.1.2. Datos de partida	24
3.2. Gráficos para la exploración y control de calidad	27
3.2.1. Control de calidad con gráficos estadísticos generales	28
3.2.2. Gráficos de diagnóstico para microarrays de dos colores	31
3.2.3. Gráficos de diagnóstico para microarrays de un color ..	33
3.3. Normalización de arrays de dos colores	37
3.3.1. Normalización global	37
3.3.2. Normalización dependiente de la intensidad	38
3.4. Resumen (sumarización) y normalización de microarrays de Affymetrix	38
3.4.1. Métodos originales de Affymetrix	39
3.4.2. El método RMA	40

3.5. Filtraje no específico	42
4. Selección de genes diferencialmente expresados.....	43
4.1. Introducción	43
4.1.1. Medidas naturales para comparar dos muestras	44
4.1.2. Selección de genes diferencialmente expresados	46
4.1.3. Potencia y tamaño muestral	50
4.1.4. El problema de la multiplicidad de tests	51
4.2. Modelos lineales para la selección de genes: <code>limma</code>	53
4.2.1. El modelo lineal general	53
4.2.2. Ejemplos de situaciones modelizables linealmente	54
4.2.3. Ejemplo 2: Comparación de tres grupos	55
4.2.4. Estimación e inferencia con el modelo lineal	61
4.2.5. Modelos lineales para microarrays	62
4.2.6. Implementación y ejemplos	63
5. Después de la selección: análisis de listas de genes.....	67
5.1. Introducción	67
5.2. Análisis comparativo de listas de genes	67
5.3. Anotación de los resultados	68
5.4. Visualización de los perfiles de expresión	71
5.5. Análisis de significación biológica	72
5.5.1. Análisis de enriquecimiento	72
5.5.2. Análisis de enriquecimiento con Bioconductor	73
6. Caso resuelto: Selección de genes diferencialmente expresados.....	75
6.1. Introducción	75
6.1.1. El ejemplo	75
6.1.2. Directorios y opciones de trabajo	76
6.2. Obtención y lectura de los datos	76
6.2.1. Los datos	76
6.2.2. Lectura de los datos	77
6.3. Exploración, control de calidad y normalización	78
6.3.1. Exploración y visualización	78
6.3.2. Control de calidad	80
6.3.3. Normalización y filtraje	82
6.4. Selección de genes diferencialmente expresados	85
6.4.1. Análisis basado en modelos lineales	85
6.4.2. Comparaciones múltiples	88
6.4.3. Anotación de resultados	89
6.4.4. Visualización de los perfiles de expresión	90
7. Métodos avanzados: Busca de patrones y predicción.....	92
7.1. Descubrimiento de grupos en datos de microarrays	92
7.1.1. Principios y métodos para el descubrimiento de clases	94

7.1.2. Consistencia de la agrupación	96
7.2. Diagnósticos moleculares y métodos de clasificación	98
7.2.1. La selección de variables para la clasificación	103
8. Caso resuelto: Descubrimiento de clases.....	106
8.1. <i>Clustering</i> y visualización	106
8.1.1. Preprocesamiento de los datos	106
8.1.2. Agrupación jerárquica de las muestras	108
8.1.3. Agrupación de genes por el método de las k-means.....	110
8.1.4. Anotación de resultados	113
Resumen.....	115
Bibliografía.....	117

1. El proceso de análisis de datos de microarrays

1.1. Introducción

Este capítulo es una corta transición entre el módulo “Preliminares” del curso, en el que se han presentado los conceptos y herramientas básicos, y el presente módulo, donde se presentan por separado y con mayor detalle los métodos de análisis de datos de microarrays (MDA).

Su objetivo, por tanto, es ofrecer una visión de conjunto que sirva de guía (*roadmap*) para los apartados siguientes de modo que, sin perder el detalle de cada uno de ellos, tengamos conciencia de en qué punto del proceso general nos encontramos.

El capítulo se estructura en dos partes. En la primera se presentan brevemente algunos de los problemas que típicamente se pueden querer estudiar con microarrays u otras técnicas similares de análisis de datos de alto rendimiento. A continuación se trata lo que se ha llamado el proceso de análisis de microarrays. Finalmente, se introducen algunos casos reales que, a modo de ejemplo, se utilizarán en los apartados siguientes.

1.2. Tipos de estudios

Los microarrays y otras tecnologías de alto rendimiento se han aplicado a multitud de investigaciones, de tipos muy diversos que van desde el estudio del cáncer ([1], [20], [47]) al de la germinación y la maduración del tomate ([35]). A pesar de ello, no resulta complicado clasificar los estudios realizados en algunos de los grandes bloques que se describen a continuación. La clasificación está basada en el excelente texto de Simon y otros ([41]), y aunque se origina en problemas de microarrays, se puede aplicar fácilmente a estudios de genómica o ultrasecuenciación.

1.2.1. Comparación de grupos o *class comparison*

El objetivo de los estudios comparativos es determinar si los perfiles de expresión génica difieren entre grupos previamente identificados. También se conocen estos estudios como de selección de genes diferencialmente expresados, y son sin duda los más habituales. Los grupos pueden representar una gran variedad de condiciones, desde distintos tejidos a distintos tratamientos o múltiples combinaciones de factores experimentales.

El análisis de este tipo de experimentos, que se describe en el apartado 4 sobre selección de genes diferencialmente expresados, utiliza herramientas estadísticas, como las pruebas de comparación de grupos paramétricas (*t* de Student) o no (test de Mann-Whitney) o diversos métodos de análisis de la varianza. Entre los ejemplos del subapartado 1.3, los casos 1.3.1, 1.3.3 o 1.3.4 hacen referencia a estudios comparativos.

1.2.2. Predicción de clase

La predicción de clase (en inglés, *class prediction*) puede confundirse con la selección de genes en tanto que disponemos de clases predefinidas, pero su objetivo es distinto, ya que no pretende simplemente buscar genes cuya expresión sea distinta entre dichos grupos, sino genes que puedan ser utilizados para identificar a qué clase pertenece un “nuevo” individuo dado cuya clase es *a priori* desconocida. El proceso de predicción suele empezar con una selección de genes informativos, que pueden ser, o no, los mismos que se obtendrían si aplicáramos los métodos del apartado 1.2.1., seguida de la construcción de un modelo de predicción y, lo que es más importante, de la verificación o validación de dicho modelo con unos datos nuevos independientes de los utilizados para el desarrollo del modelo.

Aunque el interés de la predicción de clase es muy alto, se trata de un procedimiento mucho más complejo y con más posibilidades de error que la simple selección de genes diferencialmente expresados. Entre los ejemplos del subapartado 1.3, el caso 1.3.2 trata de un problema de predicción y de uno de descubrimiento de clases.

1.2.3. Descubrimiento de clases

Un problema distinto a los descritos se presenta cuando no se conocen las clases en que se agrupan los individuos. En este caso, de lo que se trata es de encontrar grupos entre los datos que permitan reunir a los individuos más parecidos entre sí y distintos de los de los demás grupos. Los métodos estadísticos que se emplearán en estos casos se conocen como análisis de conglomerados (*cluster analysis*) y no son tan complejos como los de predicción de clase, aunque algunos aspectos, como por ejemplo la definición del número de grupos, no es sencillo. Entre los ejemplos del subapartado 1.3, tanto el caso `golub`, en parte, como el `breastTum` tratan problemas de descubrimiento de clases.

Una curiosidad del campo de la estadística es que el término clasificación aparece usado de forma indistinta para referirse a problemas de predicción de clase o de descubrimiento de clase (en inglés, *class discovery*).

1.2.4. Otros tipos de estudios

Además de los tipos de estudio reseñados, existen otros que no encajan en ninguno de los tipos descritos. Sin entrar en detalles, podemos señalar los estudios de evolución a lo largo del tiempo (*time course*), los de significación biológica (*gene enrichment analysis*, *gene set enrichment analysis*, etc.), los que buscan relaciones entre los genes (*network analysis* o *pathway analysis*). Para los objetivos de este documento con conocer e identificar los tres grandes bloques mencionados resultará más que suficiente.

1.3. Algunos ejemplos concretos

Una de las dificultades con las que se encuentra la persona que se inicia en el análisis de datos de microarrays es de dónde puede obtener ejemplos concretos con los que practicar las técnicas que está aprendiendo.

No es difícil encontrar datos de microarrays en Internet, por lo que se han seleccionado algunos conjuntos interesantes para utilizarlos como ejemplos a lo largo del curso. Algunos de estos son “populares” en el sentido de que han sido utilizados en diversas ocasiones y por lo tanto, se encuentran bien documentados. Otros se han escogido simplemente porque ilustran bien algunas de las ideas que se desea exponer o por su accesibilidad.

Todos los datos corresponden a investigaciones publicadas, por lo que no se describen exhaustivamente, sino que se expone brevemente el origen y objetivos del trabajo –incluyendo su clasificación según los grupos definidos en el subapartado anterior– y las características pertinentes para su análisis, como el tipo de microarrays, los grupos –si los hay– o cómo acceder a los datos.

1.3.1. Estudio de procesos regulados por citoquinas

Este conjunto de datos, que se denominará *celltypes*, corresponde a un estudio realizado por Chelvarajan y otros ([6]) que analizaron el efecto de la estimulación con lipopolisacáridos en la regulación por parte de citoquinas de ciertos procesos biológicos relacionados con la inflamación.

Este estudio es del tipo *class comparison*, es decir, su principal objetivo es la obtención de genes diferencialmente expresados entre dos o más condiciones.

Los datos se encuentran disponibles en la base de datos pública *caarray* mantenida por el National Institute of Health (NIH), pero pueden descargarse de la página de materiales del curso para garantizar su disponibilidad.

1.3.2. Clasificación molecular de la leucemia

A finales de los años noventa, Todd Golub y sus colaboradores ([20]) realizaron uno de los estudios más populares hasta el momento con datos de microarrays. En él utilizaron microarrays de oligonucleótidos para 6817 genes humanos con el objetivo de encontrar una forma de distinguir (clasificar) tumores de pacientes con leucemia linfoblástica aguda (ALL) de aquellos que sufrían de leucemia mieloide aguda (AML). Además se deseaba descubrir subgrupos de forma que pudieran definirse variantes de cada una de estas patologías a nivel molecular.

La diversidad de objetivos del estudio lleva a clasificarlo tanto entre los del tipo de predicción de clase como entre los que buscan descubrir nuevas clases o grupos en los datos.

Los datos de este estudio se encuentran disponibles en la web del Instituto Broad, donde se llevó a cabo. También se encuentra disponible un paquete de Bioconductor denominado `ALL` que permite utilizarlos directamente usando R y Bioconductor.

1.3.3. Efecto del estrógeno y el tiempo de administración

Scholtens y otros ([40]) describen un estudio sobre el efecto de un tratamiento con estrógenos y del tiempo transcurrido desde el tratamiento, en la expresión génica en pacientes con cáncer de mama. Los investigadores supusieron que los genes asociados con una respuesta temprana podrían considerarse dianas directas del tratamiento, mientras que los que tardaron más en hacerlo podrían considerarse objetivos secundarios correspondientes a dianas más alejadas en las vías metabólicas.

Estos datos han sido utilizados multitud de veces en los cursos de análisis de microarrays realizados por el proyecto Bioconductor y se encuentran disponibles en forma de paquete de R, el paquete `estrogen`. Una característica importante de este paquete es el hecho de que en lugar de los datos previamente procesados proporciona los datos originales (*raw data*) en forma de archivos .CEL de Affymetrix. Esto permite una mayor flexibilidad a la hora de reutilizarlos lo que explica su popularidad.

1.3.4. Efecto del CCL4 en la expresión génica

Holger y otros de la empresa LGC Ltd. en Teddington, Inglaterra, realizaron unos experimentos con microarrays de dos colores en los que trataron hepatocitos de ratón con tetraclorido de carbono (CCL4) o con dimetilsulfóxido (DMSO). El tetraclorido de carbono fue ampliamente utilizado en productos

de limpieza o refrigeración para el hogar hasta que se detectó que podía tener efectos tóxicos e incluso cancerígenos. El DMSO es un solvente similar, sin efectos tóxicos conocidos, que se utilizó como control negativo.

Los datos de este estudio no han sido publicados pero se encuentran disponibles en el paquete `CCL4` de Bioconductor. Su interés reside, por un lado, en que se trata de datos de microarrays de dos colores de la marca Agilent –en un momento en que la mayoría de estudios se realizan con datos de un color. Aparte de esto cabe resaltar el hecho de que el paquete incluye, de forma similar al anterior, los datos originales en forma de archivos de tipo Genepix, uno de los programas populares para escanear imágenes generadas con microarrays de dos colores.

Este estudio es también un estudio de comparación de clases que tienen el objetivo principal de la selección de genes cuya expresión se asocia al tratamiento con CCL4 o DMSO.

1.3.5. Análisis de patrones en el ciclo celular

Los datos de este ejemplo denominado `kidney` son datos ya normalizados referidos a la expresión de los genes en distintos momentos del ciclo celular de la levadura, es decir, desde que concluye la división celular hasta que se inicia la siguiente.

Los datos puede descargarse desde la página web del proyecto Yeast Cell Cycle Project (<http://genome-www.stanford.edu/cellcycle/data/rawdata/>).

1.3.6. Recapitulación

La tabla 1 resume la lista de ejemplos que se utilizan en este manual indicando el nombre con que nos referiremos de aquí en adelante a cada conjunto de datos así como algunas de sus características.

Tabla 1. Conjuntos de datos utilizados en este manual

Nombre	Tipo	N. Muestras	Tipo de estudio
<code>celltypes</code>	Un color (Affy, Mouse 4302)	12	Comparativo
<code>ALL</code>	Un color (Affy, HGU95A)	38	Clasificación
<code>estrogen</code>	Un color (Affy, HGU95A)	8	Comparativo
<code>CCL4</code>	Dos colores (Agilent, WG Rat Microarray)	16	Comparativo
<code>yeast</code>	texto	49	Clasificación (<i>clustering</i>)

Lista de conjuntos de datos para utilizar como ejemplos. Además del nombre que se le asigna, se indica el tipo y número de arrays y el tipo de estudio para el que el conjunto ha sido ideado.

1.4. El proceso de análisis de microarrays

Una vez descritos los tipos de análisis y algunos ejemplos, podemos pasar a describir el proceso de análisis de microarrays que se resume brevemente en la figura 1.

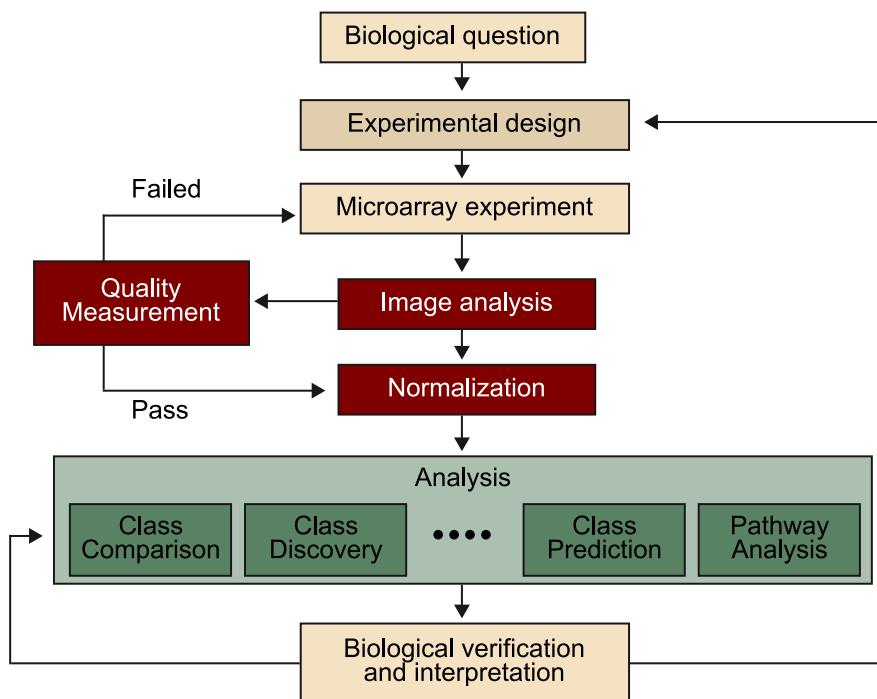


Figura 1. Proceso de análisis de microarrays
El análisis de microarrays puede ser fácilmente visualizado como un proceso que empieza por una pregunta biológica y concluye con una interpretación de los resultados de los análisis que, de alguna forma, confiamos que nos acerque un poco a la respuesta de la pregunta inicial.

El análisis de microarrays, como la mayoría de análisis, debe proceder de forma ordenada y siguiendo el método científico:

- La pregunta y su contexto nos servirán de guía para definir el diseño experimental adecuado.
- El experimento se deberá realizar siguiendo las pautas decididas en el diseño experimental y los datos obtenidos (*raw data*) deberán someterse a los controles de calidad adecuados antes de continuar con su análisis.
- Una vez decidido si la calidad de los datos es aceptable, pasaremos a prepararlos para el análisis, lo que puede incluir diversas formas de preprocesado, o de transformaciones que a menudo se incluyen de forma general bajo el paraguas del término *normalización*, aunque, como veremos, se trata de conceptos distintos.
- Los datos normalizados se utilizarán para los análisis estadísticos que hayamos decidido realizar durante el diseño experimental.

- Finalmente, los resultados de los análisis serán la base para una interpretación biológica de los resultados del experimento.

El proceso descrito es básicamente una forma razonable de proceder en general. Los microarrays y otros datos genómicos difieren en varios aspectos de los datos clásicos en torno a los cuales se han desarrollado la mayor parte de técnicas estadísticas. En consecuencia, en muchos casos ha sido necesario adaptar las técnicas existentes o desarrollar otras nuevas para adecuarse a las nuevas situaciones encontradas. Esto ha determinado que existan muchos métodos para cada una de los pasos descritos anteriormente, lo que da lugar a una grandísima cantidad de posibilidades.

En la práctica, lo que suele hacerse es optar por utilizar algunos de los métodos en los que hay un cierto consenso acerca de su calidad y utilidad para cada problema. Allison ([2]) repasa los puntos principales de este consenso dando una lista de puntos a tener en cuenta en cualquier estudio que utilice microarrays. Imbeaud y Auffray ([22]) citan una lista de hasta 39 puntos que deben seguirse en un experimento con microarrays para ser considerados de buenas prácticas.

Finalmente, Zhu y otros ([51]) utilizan un conjunto de arrays con valores de expresión conocidos para proponer los que, a su parecer, resultan los métodos más apropiados para cada etapa, desde la corrección del *background* hasta la selección de genes diferencialmente expresados. La figura 2 ilustra algunas de las opciones sugeridas por dichos autores.

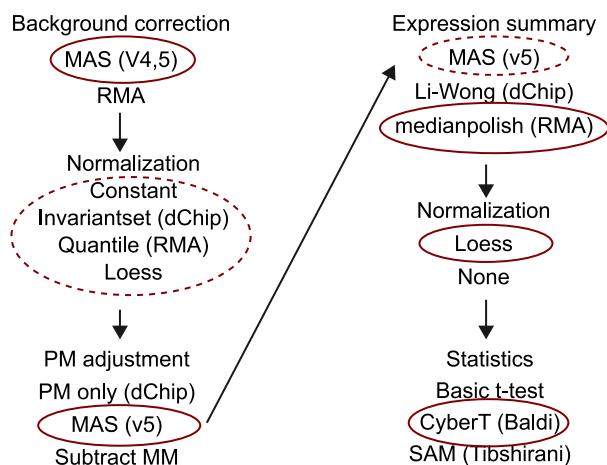


Figura 2. Representación esquemática de las opciones más habituales para cada etapa de preprocessado y de selección de genes diferencialmente expresados. Zhu y otros ([51]) han estudiado las distintas combinaciones de opciones usando un conjunto de datos preparado específicamente para su estudio y han concluido con una lista de recomendaciones para cada etapa (opciones encerradas con una línea continua).

Los apartados que siguen plantean los métodos en el orden del proceso descrito en la figura 2. Se empieza por tratar los principios del diseño de experimentos. A continuación se describen algunos métodos para el control de calidad, el preprocessado y la normalización de los datos. Se sigue con los métodos

de selección de genes y los métodos de clasificación, para concluir con una introducción a los métodos de análisis de la significación biológica de las listas de genes obtenidas de los procesos anteriores.

2. Diseño de experimentos de microarrays

En este apartado se examinan unas componentes clave del análisis de microarrays, el diseño de experimentos que, no solo es crucial para una buena recogida de información, sino que marca todo el proceso desde el preprocesado al análisis final.

2.1. Fuentes de variabilidad

Un aspecto muy importante en cualquier tipo de estudio son las fuentes de variación de los datos. Cuanto mayor es su número e intensidad, más complejo resulta extraer información útil de estudios comparativos.

En el caso de los datos genómicos, la variabilidad es particularmente elevada debido al gran número de posibles fuentes de variación que pueden presentarse (véase figura 3).

En el diseño de experimentos genómicos se tendrá que tener en cuenta cuáles son las principales fuentes que se desea tener en cuenta y del resultado de dicha decisión dependerá a menudo la posibilidad de extraer conclusiones de los experimentos.

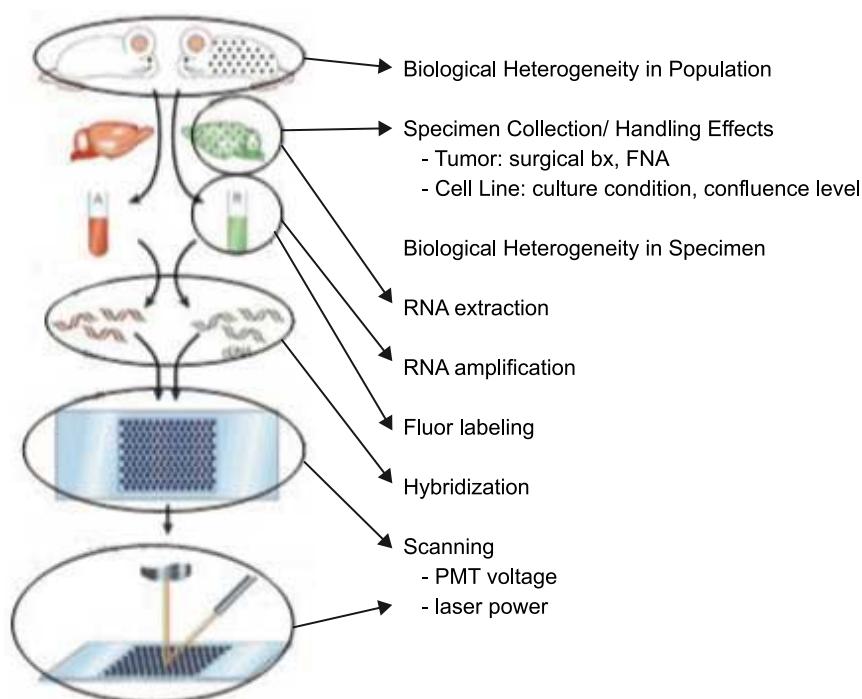


Figura 3. Fuentes de variabilidad en experimentos con microarrays. Los experimentos con microarrays presentan muchas posibles fuentes de variabilidad que tienen que ser controladas adecuadamente para asegurar la reproducibilidad de los estudios. Geschwind ([19]) ilustra algunas de estas fuentes.

Cuando se considera cómo controlar la variabilidad de los estudios, una distinción importante es la que se establece entre variabilidad sistemática y aleatoria.

La variabilidad sistemática se suele considerar principalmente debida a procesos técnicos, mientras que la variabilidad aleatoria suele ser atribuible tanto a razones técnicas como biológicas. Un ejemplo muy conocido de variabilidad sistemática en los microarrays de dos colores es el hecho de que, de los dos colorantes utilizados, Cy3 y Cy5, uno se absorbe más fácilmente que el otro, por lo que iguales cantidades de ARN marcadas con los dos colorantes darán lugar a intensidades diferentes.

Un ejemplo de variabilidad aleatoria es simplemente la variabilidad entre especímenes –ya sean pacientes, animales o cultivos– que participan en un experimento.

La manera natural de manejar la variabilidad aleatoria es mediante la utilización de un diseño experimental adecuado como, por un lado, contribuirá en lo posible a reducir el error experimental y, por el otro, facilitará la aplicación de los métodos de la inferencia estadística para extraer conclusiones estadísticamente válidas, es decir, que puedan ser extrapoladas a la población.

La manera habitual de tratar con la variabilidad sistemática es mediante el ajuste de los instrumentos –o de las condiciones experimentales– para que tengan en cuenta el posible error sistemático que pueda estar cometiéndose. Esto puede consistir en una calibración adecuada de los instrumentos o en técnicas para compensar los errores como el *dye-swap*, comentado más adelante para compensar la absorción diferencial de los colorantes.

2.2. Algunas definiciones básicas

Como la mayoría de los ámbitos, el diseño experimental tiene su propio lenguaje, que se utiliza para describir con propiedad y precisión los diseños experimentales realizados. A continuación se definen algunos términos que se utilizarán a lo largo del resto del documento.

- **Unidad experimental:** entidad física a la que se aplica un tratamiento de forma independiente al resto de unidades. En cada unidad experimental se pueden realizar una o varias medidas. Si en una misma unidad experimental se realizan varias medidas, nos referimos a cada una de ellas como unidades observacionales.

Por ejemplo, en el estudio que denominamos *celltypes*, sobre el efecto de la estimulación con LPS en la actividad de las citoquinas cada ratón es una unidad experimental.

- **Factor:** son las variables independientes, típicamente tratamientos o condiciones experimentales fijadas por el experimentador, las que pueden in-

fluir en la variabilidad de la variable de interés. En el estudio *celltypes* hay dos factores a tener en cuenta: el tratamiento (estimulación o no con LPS) y la edad de los animales (jóvenes o viejos).

- **Niveles:** cada una de las categorías o posibles valores de un factor. El conjunto de valores de un factor y su tipo definen la clasificación del factor:
 - Si el factor toma valores categóricos o numéricos discretos es un factor categórico. Si el factor toma valores numéricos continuos, es un factor numérico.
 - Si el factor toma todos posibles valores del conjunto de niveles, se habla de factor fijo. Si el factor toma algunos de los posibles valores del conjunto de niveles (es decir, si se tiene únicamente una muestra de los posibles niveles), se habla de factor aleatorio.

En el estudio *celltypes* el factor tratamiento es un factor fijo con dos niveles: "LPS" y "medio" y el factor edad es un factor fijo con dos niveles "joven" y "viejo".

- **Tratamiento:** es una combinación específica de los niveles de los factores en estudio. Son, por tanto, las condiciones experimentales que se desean comparar en el experimento. En un diseño con un único factor son los distintos niveles del factor y en un diseño con varios factores son las distintas combinaciones de niveles de los factores.

2.3. Principios básicos en el diseño del experimento

Al planificar un experimento hay tres principios básicos que se deben tener siempre en cuenta: la replicación, el control local o bloqueo y la aleatorización. Aplicados correctamente dichos principios garantizan diseños experimentales eficientes.

2.3.1. Replicación

La replicación o repetición de un experimento de forma idéntica en un número determinado de unidades, es la que permite la realización de un posterior análisis estadístico. La utilización de réplicas es importante para incrementar la precisión, obtener suficiente potencia en los tests y como base para los procedimientos de inferencia. Normalmente, distinguimos dos tipos de réplicas en el análisis de microarrays:

- **La replicación técnica** se da cuando se analiza la misma muestra en múltiples ocasiones. Puede ser tanto la replicación de *spots* en el mismo chip (es decir, si el chip contiene más de un *spot* con las mismas secuencias, como se hibridan con la misma muestra, estos *spots* son réplicas técnicas)

como la de diferentes alícuotas de la misma muestra hibridadas en diferentes microarrays.

- La **replicación biológica** se da cuando se analizan múltiples muestras del mismo tipo de tejido u organismo bajo las mismas condiciones experimentales.

La replicación técnica permite la estimación del error a nivel de medida, mientras que la replicación biológica permite estimar la variabilidad a nivel de población.

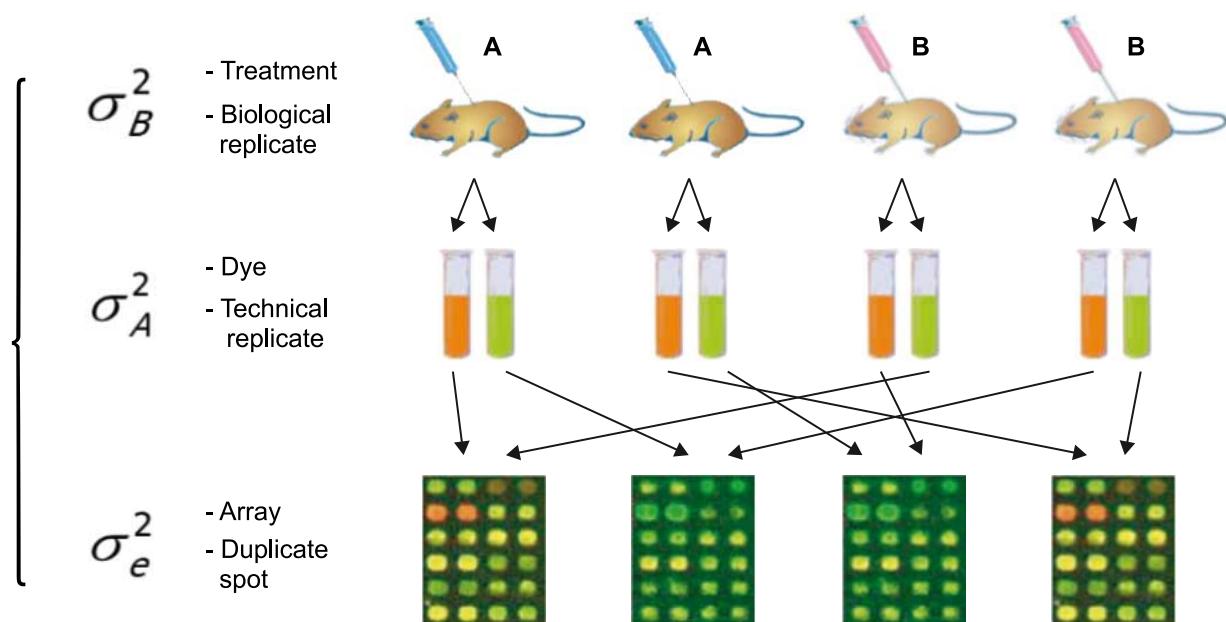


Figura 4. Réplicas biológicas frente a técnicas

Potencia y tamaño de la muestra

Sorprendentemente, los primeros experimentos de microarrays utilizaban pocas o ninguna réplica biológica. La principal explicación para este hecho -además de la falta de conocimiento estadístico- fue el alto coste de cada microarray. En pocos años, la necesidad de las réplicas ha llegado a ser indiscutible, y al mismo tiempo, el coste de los chips ha disminuido considerablemente. Actualmente, es normal la utilización de, al menos, tres o cinco réplicas por condición experimental, aunque a este consenso se ha llegado más por razones empíricas que por la disponibilidad de modelos apropiados para el análisis de la potencia y tamaño de la muestra.

Recientemente, se ha producido una importante afluencia de artículos describiendo métodos para el análisis de la potencia y tamaño de la muestra, véase por ejemplo Jung y otros [25], Lin y otros [33] y Van Iterson y otros [24].. A pesar de su variedad, ningún método aparece como candidato claro para ser utilizado en situaciones prácticas. Esto se debe, probablemente, a la complejidad de los datos de microarrays, básicamente porque los genes no son inde-

pendientes. Por tanto, se sabe que existen estructuras de correlación en los datos, pero la mayor parte de las dependencias son desconocidas, por lo que la estimación de estas estructuras es muy complicada.

Como indicaba Allison ([2]), aunque no hay consenso sobre qué procedimiento es mejor para determinar el tamaño de las muestras, sí que lo hay sobre la conveniencia de realizar el análisis de la potencia y, por supuesto, sobre el hecho de que un mayor número de réplicas generalmente proporcionan una mayor potencia. En la práctica, una buena opción es probar algunos de los métodos existentes y utilizarlos para forjarse una opinión de consenso sobre el tamaño muestral adecuado para una potencia y un error experimental fijados.

En Internet existen varias herramientas libres que permiten realizar estos cálculos. Entre las que se pueden considerar relativamente fiables, se encuentran:

- <http://bioinformatics.mdanderson.org/MicroarraySampleSize/>
- <http://sph.umd.edu/epib/faculty/mltlee/web-front-r.html>

Pooling

En el contexto de los microarrays, llamamos *pooling* a la combinación de mARN de diferentes casos en una única muestra. Hay dos razones a favor de ello. Una es que a veces no hay suficiente ARN disponible y esta es la única forma de conseguir suficiente material para construir los arrays. Otra más discutible, es la creencia de que la variabilidad entre arrays puede reducirse utilizando *pooling*. La justificación es que combinar muestras equivale a promediar expresiones, y como ya se conoce, el promedio es menos variable que los valores individuales. A pesar de la debilidad de este argumento, es verdad que en ciertas situaciones el *pooling* puede ser apropiado y, recientemente, muchos estadísticos han dedicado sus esfuerzos a tratar de responder a la pregunta *to pool or not to pool* ([27]). Por ejemplo, si la variabilidad biológica está altamente relacionada con el error en las medidas, y las muestras biológicas tienen un coste mínimo en comparación con el de los arrays, una estrategia apropiada de *pooling* puede servir para reducir los costes del estudio manteniendo su potencia.

En todo caso, el *pooling* no se debería usar en cualquier tipo de estudios. Si el objetivo es comparar expresiones medias, puede funcionar adecuadamente, pero se debería evitar claramente cuando el objetivo del experimento es construir predictores que se basen en características individuales.

2.3.2. Aleatorización

Se entiende por aleatorizar la asignación de todos los factores al azar a las unidades experimentales. Con ello se consigue disminuir el efecto de los factores no controlados por el experimentador en el diseño experimental y que podrían influir en los resultados.

Las ventajas de aleatorizar los factores no controlados son:

- Transforma la variabilidad sistemática no planificada en variabilidad no planificada o ruido aleatorio. Dicho de otra forma, aleatorizar previene contra la introducción de sesgos en el experimento.
- Evita la dependencia entre observaciones al aleatorizar los instantes de recogida muestral.
- Valida muchos de los procedimientos estadísticos más comunes.

2.3.3. Bloquear

Supongamos que un estudio quiere comparar dos tratamientos pero las muestras no son homogéneas: pueden estar formadas por individuos de los dos sexos, o puede saberse que los arrays se tendrán que procesar en tres lotes distintos. Si hay diferencias sistemáticas entre los bloques (un bloque = un grupo de muestras homogéneas como los individuos varones o los arrays de un día), los efectos de interés –diferencias entre tratamientos– pueden quedar confundidos por las diferencias entre los bloques. El bloqueo o control local ofrece una forma para minimizar el efecto de heterogeneidad en las muestras, debido a situaciones que no se pueden evitar (el tratamiento que recibe un individuo se puede decidir, su edad o el sexo no). A diferencia de lo que ocurre con los factores tratamiento, el experimentador no está interesado en investigar las posibles diferencias de la respuesta entre los niveles de los factores bloque.

El control local consiste en agrupar las unidades experimentales de forma que la variabilidad dentro de los grupos sea inferior a la variabilidad de todas las unidades antes de agrupar.

Esta estrategia permite evitar que las diferencias entre tratamientos se confundan con las diferencias entre unidades experimentales al reducir la variabilidad asociada a las diferencias entre unidades experimentales, atribuibles a las diferencias entre bloques.

La figura 5 ilustra las formas correcta e incorrecta de organizar la asignación de dos tratamientos a un grupo de 8 individuos en presencia de dos bloques asociados al sexo de los individuos y al día de hibridación de los arrays.

Muestra	Diseño erróneo			Muestra	Diseño correcto		
	Tratamiento	Sexo	Lote		Tratamiento	Sexo	Lote
1	A	Macho	1	1	A	Macho	1
2	A	Macho	1	2	A	Hembra	2
3	A	Macho	1	3	A	Macho	2
4	A	Macho	1	4	A	Hembra	1
5	B	Hembra	2	5	B	Macho	1
6	B	Hembra	2	6	B	Hembra	2
7	B	Hembra	2	7	B	Macho	2
8	B	Hembra	2	8	B	Hembra	1

Figura 5. Dos diseños alternativos para investigar el efecto de dos tratamientos en presencia de dos bloques atribuibles al sexo de los individuos y al lote de producción de los arrays. El diseño de la izquierda es incorrecto pues en él se confunden completamente los efectos de los tratamientos con los dos bloques. El diseño de la derecha es correcto con ambos bloques cruzados con los tratamientos.

2.4. Diseños experimentales para microarrays de dos colores

En arrays de dos colores, se aplican dos condiciones experimentales a cada array. Esto permite la estimación del efecto del array, como el efecto de bloqueo. En Affymetrix u otros microarrays de un canal, cada condición se aplica a un chip separado, imposibilitando la estimación del efecto de los microarrays, que, por otra parte, se considera que tiene una relación muy pequeña en el tratamiento de los efectos debido al proceso industrial utilizado para fabricar estos chips.

Como consecuencia de lo anterior, los experimentos que usan arrays de un canal se consideran estándar, en cuanto que se les pueden aplicar los conceptos y técnicas tradicionales de diseño de experimentos.

Los arrays de dos canales presentan una situación más complicada. Por una parte los dos colores no son simétricos, es decir, con la misma cantidad de material, un array hibridado con uno u otro color, sea Cy5 o Cy3, emitirá señales con diferente intensidad. Una forma de reducir este problema es mediante *dye swapping*, que consiste en utilizar para cada comparación dos arrays con los colorantes (*dyes*) intercambiados, es decir, si en el primer array la muestra 1 se marca con Cy3 y la muestra 2 con Cy5, con las muestras del segundo array se hace al revés (figura 6).

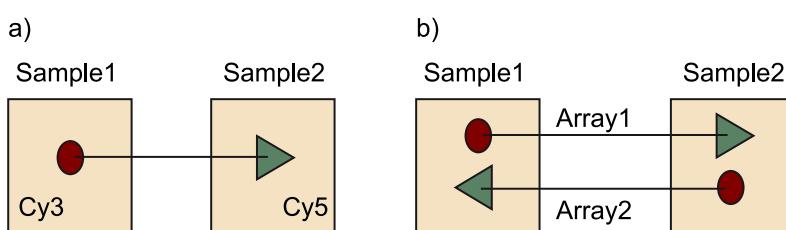


Figura 6. a. Representación simplificada de un diseño. Cada flecha representa un array de dos canales en el que el origen indica el marcaje con Cy3. b. Dye swapping.

Por otra parte, el hecho de que solo se puedan aplicar dos condiciones a cada array complica el diseño, ya sea porque normalmente hay más de dos condiciones, o porque no es recomendable hibridar directamente dos muestras concretas en un array, creando pares artificiales.

Como resultado de las complejidades que presenta realizar un diseño eficiente con arrays de dos colores cuando el número de factores y condiciones experimentales es alto, el problema de la construcción de diseños óptimos ha sido estudiado exhaustivamente. A continuación se describen brevemente algunos de los diseños más conocidos.

El diseño utilizado más comúnmente en la comunidad biológica es el *diseño de referencia* en el que cada condición de interés se compara con muestras tomadas de alguna referencia estándar común a todos los arrays (figura 7 a).

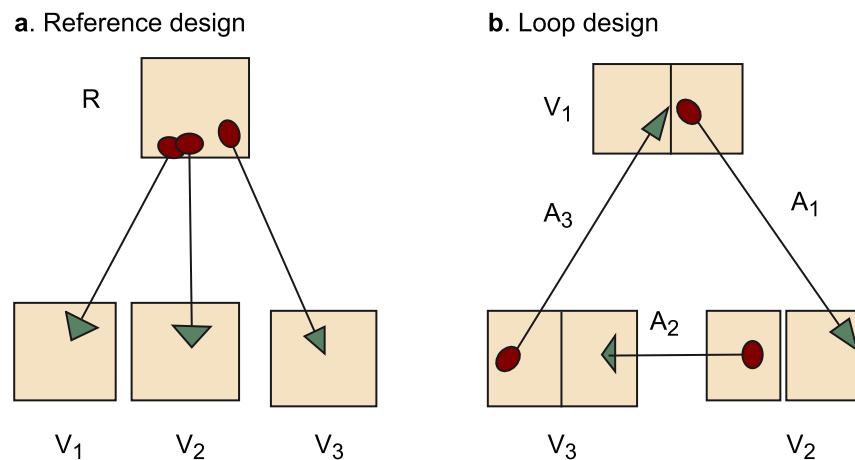


Figura 7. Diseño de arrays. a. Diseño de referencia. b. Diseño en *loop*

Los diseños de referencia permiten hacer comparaciones indirectas entre condiciones de interés. La crítica más importante a esta aproximación es que el 50% de las fuentes de hibridación se utilizan para producir la señal del grupo control, de un interés no intrínseco para los biólogos. En contraste, un diseño en bucle o diseño en *loop* compara dos condiciones a través de otra cadena de condiciones, por lo que elimina la necesidad de una muestra de referencia (figura 7 b).

3. Exploración de los datos, control de calidad y preprocesado

3.1. Introducción

Los estudios realizados con microarrays, sea cual sea la tecnología en la que se basan, tienen una característica común: generan grandes cantidades de datos a través de una serie de procesos, que hacen que su significado no siempre sea completamente intuitivo.

Como en todo tipo de análisis, antes de empezar a trabajar con los datos, debemos asegurarnos de que estos son fiables y completos y de que se encuentran en la escala apropiada para proporcionar la información que pretendemos obtener de ellos.

En el caso de los microarrays, solemos distinguir dos fases previas al análisis de los datos:

1) Exploración y control de calidad. Mediante gráficos y/o resúmenes numéricos estudiamos la estructura de los datos con el fin de decidir si parecen correctos o presentan anomalías que deben ser corregidas.

2) Preprocesado. Incluso si son correctos, los datos originales no sirven para el análisis sino que se necesita previamente:

a) Eliminar la parte de la señal no atribuible a la expresión, llamada *background* o ruido.

b) Realizar una *normalización* para hacerlos comparables entre muestras y eliminar posibles sesgos técnicos.

c) Resumir (también denominado sumarizar porque describe mejor el término original inglés *summarize*) la información de un gen en un solo valor cuando la tecnología de los microarrays lo haga necesario, como es el caso de los arrays de Affymetrix.

d) Eventualmente puede ser preciso realizar una transformación de los datos de forma que la escala sea razonable y facilite el análisis.

3.1.1. Nivel de análisis y tipo de microarray

Desde la generalización del uso de los microarrays se han desarrollado muchas formas para visualizar los datos y decidir acerca de su calidad. Algunas trabajan sobre los datos obtenidos del escáner, otras lo hacen con los datos normalizados. Algunas sirven tanto para arrays de dos colores como arrays de un color. Otras son específicas de la tecnología. Con el fin de organizar esta multitud de opciones podemos diferenciar:

- **Datos de bajo nivel.** Son los datos proporcionados por el escáner contenidos por ejemplo en archivos .gpr (dos colores) o .cel (un color). Estos últimos son binarios por lo que no es posible ni tan solo visualizarlos sin programas específicos.
- **Datos de alto nivel.** Son los datos resultantes del (pre)procesado de los datos de bajo nivel. Básicamente se corresponden con los datos de expresión ya resumidos (sumarizados), normalizados o no.

La exploración y el control de calidad pueden basarse en datos de bajo o de alto nivel. La tabla 2 muestra una clasificación de los distintos procedimientos según el tipo de datos y el tipo de microarray. Dichos procedimientos se explican en los subapartados.

Tabla 2. Procedimientos de visualización y control de calidad según el tipo de datos (de bajo o alto nivel) y el tipo de microarrays (de dos colores o un color). El significado de las abreviaturas es el siguiente: R: Rojo; G: Verde; Gr: Gráfico.

Tipo de microarrays. Nivel de análisis	General	Dos colores	Un color
Bajo nivel 1 color: (sondas) 2 colores: (1 canal R o G)	PCA Histograma Boxplot	R-G Scatterplots MA-plot Signal2Noise plot Imagen (R o G)	Imagen (Sondas) Gr. de degradación Gr. de densidad Gr. de probesets
Alto nivel 1 color: expresión absoluta 2 colores: expresión relativa	PCA Histogramas Boxplots	Imagen (R) Imagen (G) Imagen (R/G)	MA-Plots PLM-plots (NUSE, RLE, Residuos)

3.1.2. Datos de partida

La estructura de los datos de microarrays de un color o de dos colores difiere considerablemente, tanto a nivel físico (los chips y las imágenes que de ellos se obtiene) como informático, es decir en la forma en que se representan.

Arrays de dos colores

Tradicionalmente, los arrays de dos colores o de cDNA se realizaban de forma menos automatizada que los de un color o de Affymetrix. Esto implica que, tras obtener la imagen se efectuaba el escaneado del archivo .tiff resultante mediante un software independiente como Genepix. Este programa convier-

te las imágenes en números y genera un archivo de información (con extensión .gpr) a partir del cual pueden calcularse las expresiones relativas, así como valores de calidad para cada *spot* o punto escaneado en la imagen.

Para cada imagen (o sea para cada microarray) hay un archivo .gpr que contiene una fila por gen y varias columnas con distintos valores, por ejemplo, la intensidad para cada canal, valores resumen de las intensidades y controles de calidad (*FLAGS*).

La tabla 3 muestra lo que serían las primeras filas y columnas de un archivo .gpr, simplificado y con nombres imaginarios de los genes, obtenido mediante el programa Genepix.

Tabla 3. Ejemplo simplificado de las primeras filas y columnas de un archivo .gpr

Gen	Señal Cy5 (R)	Fondo Cy5 (Rb)	Señal Cy3 (G)	Fondo Cy3 (Gb)	Flag	Otros
gen-1	3.7547	1.8128	5.0672	1.8496	1	...
gen-2	0.8331	0.9175	1.1536	0.9995	0	...
gen-3	9.8254	0.2781	0.6921	0.5430	1	...
gen-4	9.1539	0.1918	3.8290	0.0014	0	...
gen-5	4.8603	0.2377	0.5338	0.3335	0	...
gen-6	7.8567	1.3941	1.3050	1.4876	1	...
gen-7	2.7619	0.8108	8.4916	1.6518	0	...
gen-8	3.6618	0.1918	0.3770	1.1842	0	...
gen-9	4.4300	0.5888	2.8161	0.8707	1	...
...

Los valores de intensidad se convierten en una única matriz de expresión que contiene una columna por chip con los valores de intensidad relativa obtenidas, por ejemplo, aplicando una fórmula del tipo:

$$\log \frac{R - Rb}{G - Gb}$$

y una fila por gen (mismas filas que los archivos .gpr).

La tabla 4 muestra lo que sería una matriz de expresión derivada de cuatro archivos .gpr como los de la tabla 3.

Tabla 4. Ejemplo simplificado de las primeras filas y columnas de una matriz de expresión relativa obtenida de cuatro archivos .gpr

	Array-1	Array-2	Array-3	Array-4
gen-1	1.65695	-0.10820	1.69515	8.25137

	Array-1	Array-2	Array-3	Array-4
gen-2	-1.82305	0.11350	1.58807	0.95676
gen-3	0.01561	0.47682	-27.88036	-1.39078
gen-4	0.42709	4.29319	-4.31366	30.96866
gen-5	0.04332	0.24126	-10.27675	0.26680
gen-6	-0.02825	0.68408	2.04163	0.99554
gen-7	3.50549	-8.04635	0.18286	0.22647
gen-8	-0.23261	-1.08477	0.19582	1.11561
gen-9	0.50643	1.52147	0.99092	0.23441
...

Arrays de un color

El resultado de escanear la imagen de un array de un color (o array de Affymetrix) es un archivo de extensión .CEL que, a diferencia de los arrays de dos colores, está en formato binario, es decir, que solo puede ser leído con programas específicos para ello.

De forma similar a los arrays de dos colores, existe un archivo .CEL por cada microarray chip, que contiene los valores PM (*perfect match*) y MM (*mismatch*) para cada sonda, o, en arrays de nueva generación que ya no contienen Mismatch, los valores de intensidad de cada sonda.

A partir de las intensidades de los archivos .CEL se genera la matriz de expresión que contiene una columna por chip con los valores de intensidad absoluta y una fila por grupo de sondas. En el caso de arrays de affymetrix existe una gran variedad de algoritmos de sumarización y según cuál se utilice se obtendrá unos u otros valores de expresión pero estos serán siempre medidas absolutas, es decir, independientes del resto de muestras.

La tabla 5 muestra las primeras filas de una matriz de expresión sumarizada correspondiente a los primeros genes del caso resuelto 1.

Tabla 5. Ejemplo simplificado de las primeras filas y columnas de una matriz de expresión absoluta obtenida de cuatro archivos ".CEL"

ID_REF	GSM188013	GSM188014	GSM188016	GSM188018
1007_s_at	15630.2	17048.8	13667.5	15138.8
1053_at	3614.4	3563.22	2604.65	1945.71
117_at	1032.67	1164.15	510.692	5061.2
121_at	5917.8	6826.67	4562.44	5870.13
1255_g_at	224.525	395.025	207.087	164.835

ID_REF	GSM188013	GSM188014	GSM188016	GSM188018
...

Datos de ejemplo

Los ejemplos de este apartado se basarán en dos de los conjuntos de datos descritos en el apartado 1. Los resultados obtenidos en estos ejemplos, como los del resto de la asignatura, son el resultado de ejecutar las instrucciones adecuadas de R. Con el fin de no romper la continuidad del texto en general, no se mostrarán dichas instrucciones –o al menos no de forma que se puedan reproducir. En los apéndices se proporcionará el código que permite reproducir cualquier ejemplo o cálculo llevado a cabo.

- Por un lado los datos del conjunto `estrogen` referidos al efecto del tratamiento con estrógenos en cáncer de mama descrito en el subapartado 1.3.3. La exploración se basará en los datos normalizados, pero también en los datos originales, de bajo nivel, contenidos en los archivos `.CEL`. Estos datos pueden obtenerse directamente del paquete de Bioconductor `estrogen`. Un aspecto interesante de este conjunto de datos es que, junto con 8 muestras correctas, se proporciona un array defectuoso, denominado `bad.cel`, que facilita el ver cómo aparecen ciertos gráficos cuando hay problemas.
- Los datos del conjunto `CC14` relativos al efecto del tratamiento con CCL4 en la expresión de los hepatocitos. Los datos resultan accesibles al instalar el paquete `CC14`.

3.2. Gráficos para la exploración y control de calidad

Los gráficos son útiles para comprobar la calidad de los datos de microarrays, obtener información sobre cómo se deben preprocesar los datos y comprobar, finalmente, que el preprocesado se haya realizado correctamente.

Siguiendo el esquema presentado en la **tabla 2**, se presentan a continuación los distintos gráficos utilizados con una breve descripción de lo que representa cada uno y cómo interpretarlos adecuadamente. El código para generar los gráficos aquí utilizados se presenta al final del apartado como un apéndice.

3.2.1. Control de calidad con gráficos estadísticos generales

Histogramas y gráficos de densidad

Estos gráficos permiten hacerse una idea de si las distribuciones de los distintos arrays son similares en forma y posición. La figura 8 muestra los histogramas correspondientes a los 9 arrays del conjunto de datos estrogen.

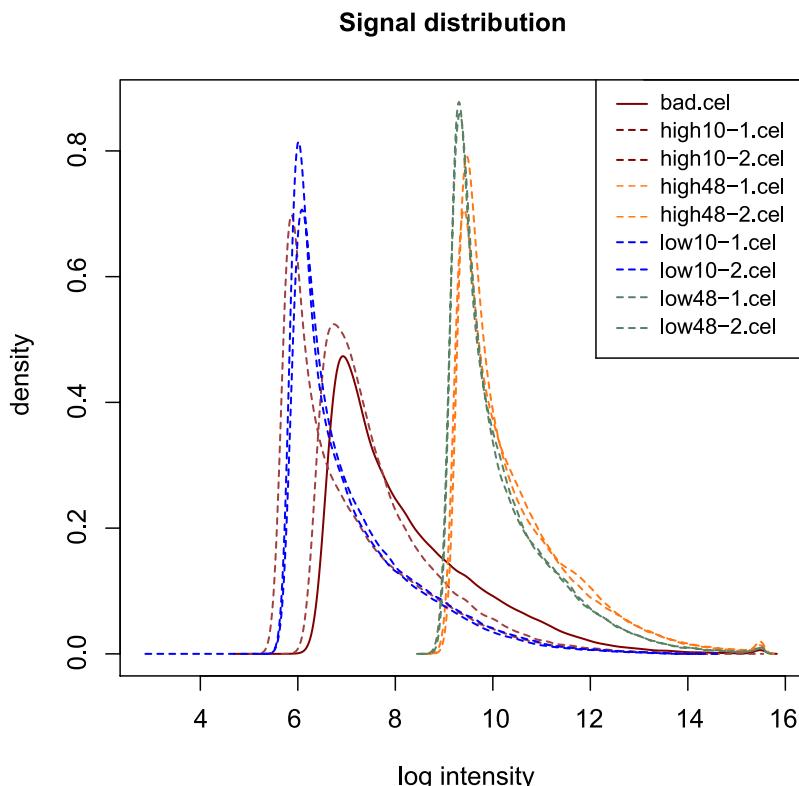


Figura 8. Histograma de los arrays correspondientes a los datos de estrógeno. Puede observarse la diferencia entre los arrays correspondientes al grupo *low* frente a los del grupo *high*. El array defectuoso no se diferencia especialmente de alguno de los que no lo son.

Diagramas de caja o “boxplots”

Como los histogramas, los diagramas de caja –basados en los distintos cuantiles de los valores– dan una idea de la distribución de las intensidades. La figura 9 muestra los diagramas de caja correspondientes a los 9 arrays del conjunto de datos estrogen.

Signal distribution

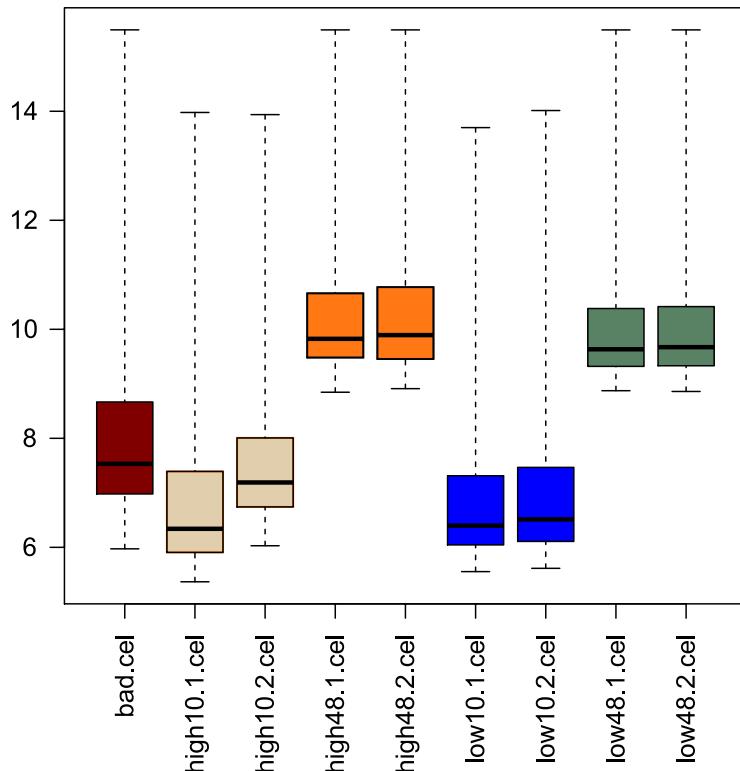


Figura 9. Diagramas de cajas de los arrays correspondientes a los datos de estrogen.

Gráficos de componentes principales

El análisis de componentes principales puede servir para detectar si las muestras se agrupan de forma “natural”, es decir, con otras muestras provenientes del mismo grupo o si no hay correspondencia clara entre grupos experimentales y proximidad en este gráfico. Cuando esto sucede no significa necesariamente que haya un problema, pero puede ser indicativo de efectos técnicos – como el conocido efecto *batch*– que podría ser necesario corregir.

```
> plotPCA <- function ( X, labels=NULL, colors=NULL, dataDesc="", scale=FALSE)
+ {
+ pcX<-prcomp(t(X), scale=scale) # o prcomp(t(X))
+ loads<- round(pcX$sdev^2/sum(pcX$sdev^2)*100,1)
+ xlab<-c(paste("PC1",loads[1],"%"))
+ ylab<-c(paste("PC2",loads[2],"%"))
+ if (is.null(colors)) colors=1
+ plot(pcX$x[,1:2],xlab=xlab,ylab=ylab, col=colors,
+       xlim=c(min(pcX$x[,1])-10, max(pcX$x[,1])+10),
+       ylim=c(min(pcX$x[,2])-10, max(pcX$x[,2])+10),
+       )
+ text(pcX$x[,1],pcX$x[,2], labels, pos=3, cex=0.8)
+ title(paste("Plot of first 2 PCs for expressions in", dataDesc, sep=" "), cex=0.8)
+ }
```

La figura 10 muestra dos diagramas de componentes principales realizados a partir de los datos normalizados del conjunto de datos *estrogen*. El gráfico de la parte superior, que incluye el array defectuoso, ilustra que la principal fuente de variabilidad es la diferencia de este array con el resto. Cuando se repite el análisis omitiendo esta muestra, puede verse cómo la principal fuente de variación (eje X) se asocia con el tiempo de exposición (48 h a la derecha, 10 h a la izquierda), mientras que la segunda fuente de variación se asocia con la exposición a los estrógenos (alto arriba, bajo abajo).

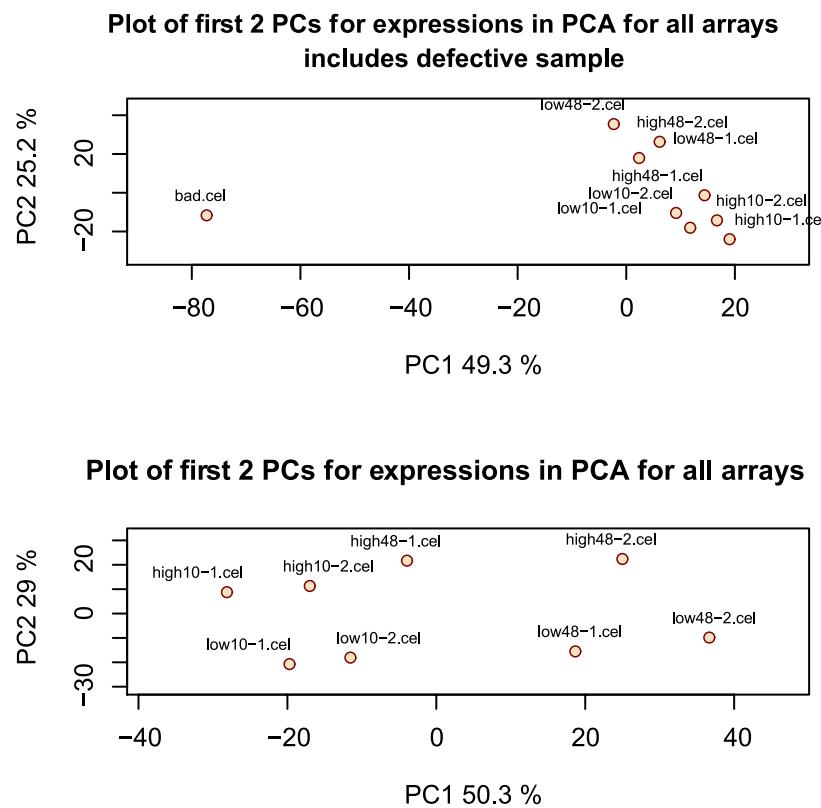


Figura 10. Gráfico de las dos primeras componentes principales para los datos de *estrogen*. La figura de la parte superior de la imagen se ha obtenido realizando el PCA con todos los arrays, incluyendo el array defectuoso. Como puede verse, la fuente principal de variabilidad es la diferencia entre este array y los demás. La figura de la parte inferior se ha obtenido realizando el PCA tan solo con los arrays correctos y explica de forma visible las dos fuentes principales de variación.

Imagen del chip

Otra forma de ver si las muestras se agrupan según los grupos experimentales, o mediante otros criterios, es usando un clúster jerárquico que realiza una agrupación básica de las muestras por grado de similaridad según la distancia que se utilice.

Como en el caso de las componentes principales, si las muestras se agrupan según las condiciones experimentales, es una buena señal, pero si no es así, puede deberse a la presencia de otra fuente de variación o bien al hecho de que se trata de un gráfico basado en todos los datos, y las condiciones experimentales pueden haber afectado un pequeño número de genes.

La figura 11 muestra cómo se agrupan los datos del conjunto *estrogen* sobre la base de un clúster jerárquico. Como en el caso de las componentes principales, tras eliminar el array defectuoso las muestras se separan, primero por el tiempo de exposición y luego por niveles de estrógenos suministrados.

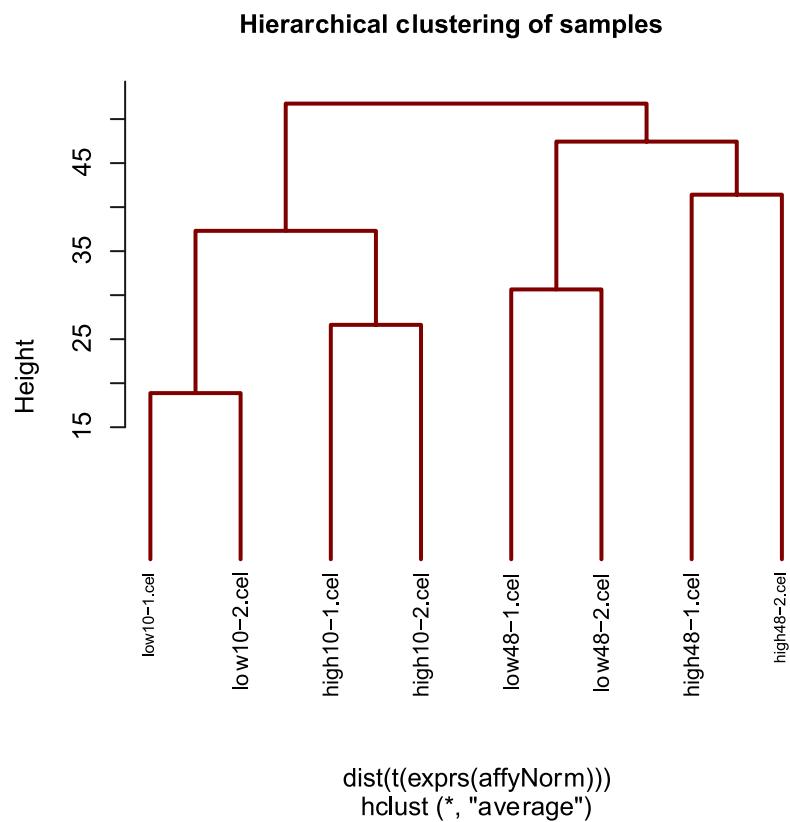


Figura 11. Agrupación de las muestras para los datos de *estrogen* en base a un clúster jerárquico.

3.2.2. Gráficos de diagnóstico para microarrays de dos colores

El diagnóstico de arrays de dos canales se basa principalmente en la imagen y en diferentes tipos de gráficos.

Diagramas de dispersión y “MAplots”

La normalización, discutida en este mismo apartado, es un punto clave en el proceso de análisis de microarrays y se ha dedicado un gran esfuerzo a desarrollar y probar diferentes métodos ([37], [49]). Una razón para ello es que hay diferentes artefactos técnicos que deben ser corregidos para poder ser utilizados, y no cualquier método puede funcionar con todos ellos.

En general, los métodos de normalización se basan en el siguiente principio: la mayor parte de los genes en un array se pueden expresar o no expresar ante cualquier condición, pero se espera que solo una pequeña cantidad de genes muestre cambios de expresión entre condiciones.

Esto da una idea de cómo debería ser un gráfico de intensidades. Por ejemplo, si no hubiese artefactos técnicos, en arrays de dos canales una gráfica de dispersión de intensidad del rojo frente al verde debería dejar la mayor parte de los puntos alrededor de una diagonal. Cualquier desviación de esta situación debería ser atribuible a razones técnicas, no biológicas, y por tanto, debería ser eliminada. Esto ha conducido a un método de normalización muy popular consistente en estimar la transformación a aplicar como una función de las intensidades, utilizando el método *lowess* en la representación transformada de la gráfica de dispersión conocida como el gráfico MAplot.

La figura 12a muestra un gráfico de dispersión del canal rojo frente al verde en un array de dos colores. El hecho de que los datos no estén centrados alrededor de la diagonal sugiere la necesidad de normalización.

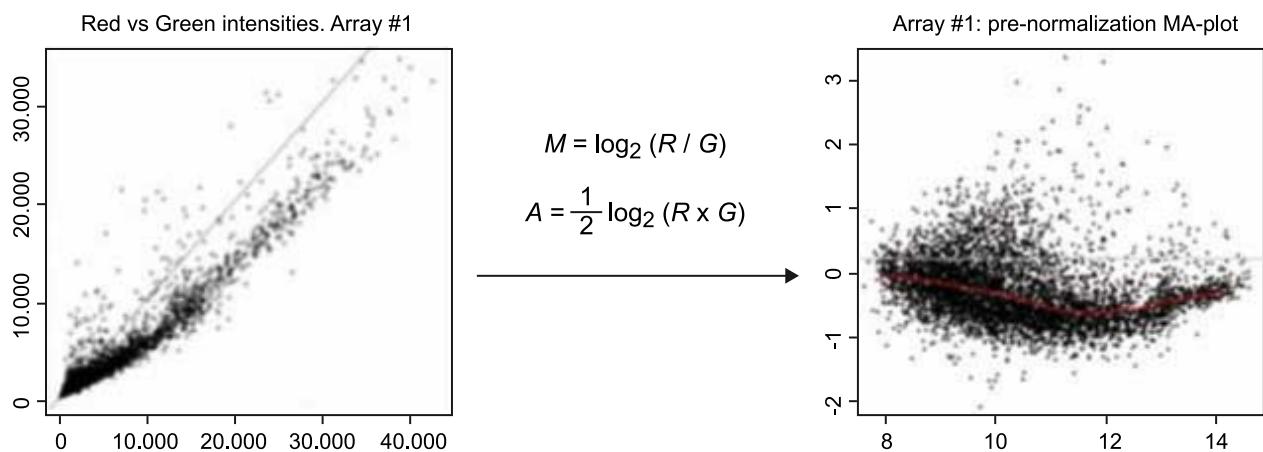


Figura 12. a. Gráfico de un canal frente al otro. b. MAplot (intensidad frente a *log-ratio*)

Una representación muy popular que ayuda a visualizar mejor esta asimetría es lo que se conoce como MAplot (figura 12b). Geométricamente, representa una rotación del gráfico de dispersión en la que el significado de los nuevos ejes es:

- $A = \frac{1}{2} (\log_2(R * G))$: El logaritmo de la intensidad media de los dos canales.
- $M = \log_2 \frac{R}{G}$: El logaritmo de la expresión relativa entre ambos canales (normalmente conocido como *log-ratio*).

Imagen del array

La imagen del chip (véase la gráfica izquierda de la figura 13) ofrece una visión rápida de la calidad del array, proporcionando información acerca del balance del color, la uniformidad en la hibridación y en los *spots*, de si el *background* es mayor de lo normal y de la existencia de artefactos como el polvo o pequeñas marcas (rasguños).

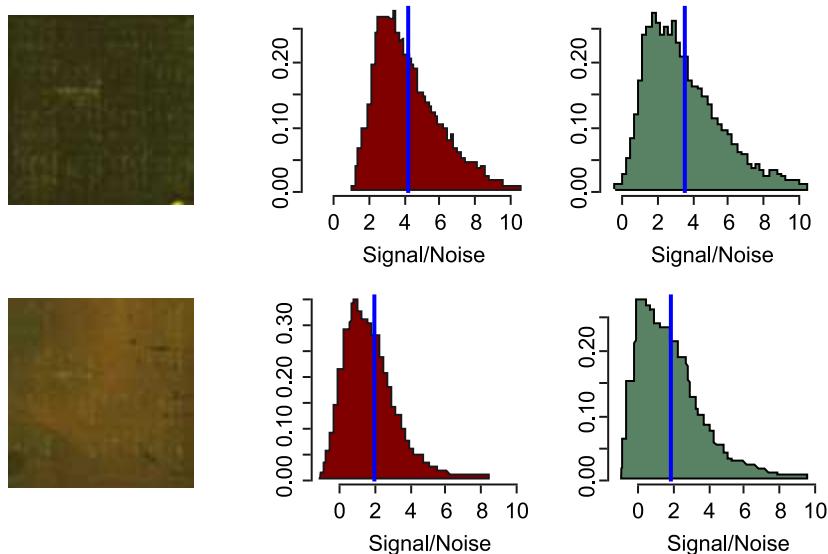


Figura 13. Imágenes de buena y de mala calidad
Cómo se puede ver en el gráfico superior, una imagen de buena calidad debería tener un *background* bajo y una alta relación señal/ruido. Imágenes de mala calidad, gráfico inferior, muestran un *background* alto y baja relación señal/ruido.

Histogramas de señales y de la relación señal-ruido

Estos gráficos (véase la gráfica derecha de la figura 13) son útiles para detectar posibles anomalías o un *background* excesivamente alto.

Boxplots

Un gráfico muy utilizado es el diagrama de cajas o boxplot múltiple con una caja por cada chip. Del alineamiento (o falta de él) y la semejanza (o disparidad) entre las cajas, se deduce si hace falta, o no, normalizar entre arrays.

En el caso de arrays de dos colores pueden utilizarse diagramas de cajas “dentro de arrays” (entre distintos sectores del mismo chip) y “entre arrays”.

3.2.3. Gráficos de diagnóstico para microarrays de un color

Imagen del chip

Los arrays de Affymetrix contienen millones de sondas por lo que no pueden examinarse a simple vista. A pesar de ello hay diversas formas de obtener una imagen que, en caso de presentar irregularidades pueden indicar algún tipo de problemas como burbujas, arañazos, etc. La figura 14 muestra algunas de las imágenes.

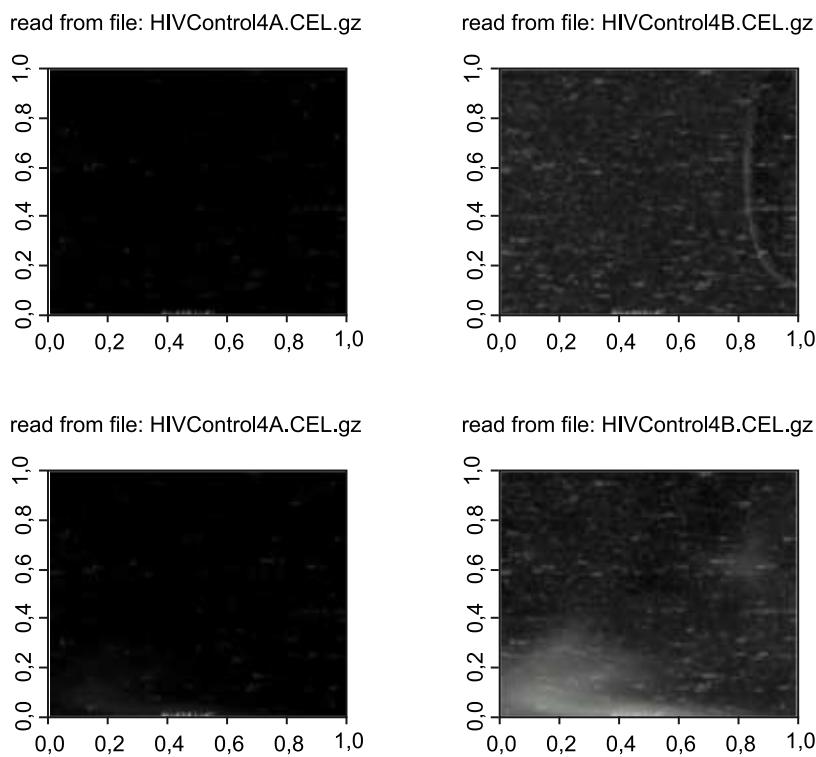


Figura 14. Imágenes de cuatro microarrays de Affymetrix.
La imagen del array de Affymetrix solo es útil para evidenciar grandes problemas como burbujas, arañazos, etc. En este ejemplo los dos arrays de la izquierda se considerarán aceptables y los de la derecha defectuosos.

Gráfico “M-A”

En los chips de dos colores el MApplot se utiliza para comparar los dos canales en cada array (rojo y verde). En cambio, en los chips de Affymetrics, en que solo hay un canal en cada array, la única forma de definir M (el *log ratio*) es a partir de la comparación entre pares de de valores, ya sea los arrays dos a dos o bien cada array respecto un valor de referencia que puede ser la mediana, punto a punto, de todos los arrays (véase por ejemplo la figura 15).

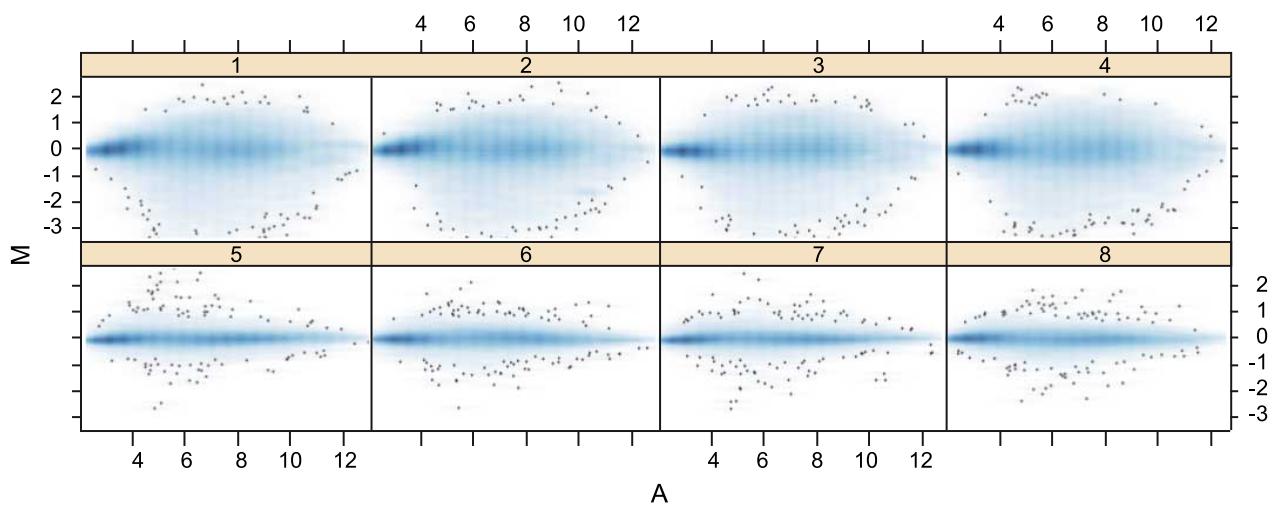


Figura 15. MApplot de un canal
En los chips de Affymetrix la única forma de definir M (el *log ratio*) es comparar entre cada array o bien con otros arrays o bien con un array de referencia creado, por ejemplo tomando gen a gen la mediana de todas las expresiones.

$M = \log_2(I_1) - \log_2(I_2)$: log ratio $A = \frac{1}{2}(\log_2(I_1) + \log_2(I_2))$: log de intensidades
 Donde I_1 es la intensidad del array de estudio, e I_2 es la intensidad media de arrays. Por lo general, se espera que la distribución en el gráfico se concentre a lo largo del eje $M = 0$.

Gráficos de degradación

La calidad de la hibridación del ARN a lo largo de los conjuntos de sondas puede variar si en un grupo de arrays el nivel de degradación es desigual y suele considerarse que su calidad es baja.

La figura 16 muestra los gráficos de degradación para el conjunto de datos estrogen pudiendo observarse cómo, en este caso, el array `bad.cel` no presenta un patrón de degradación distinto.

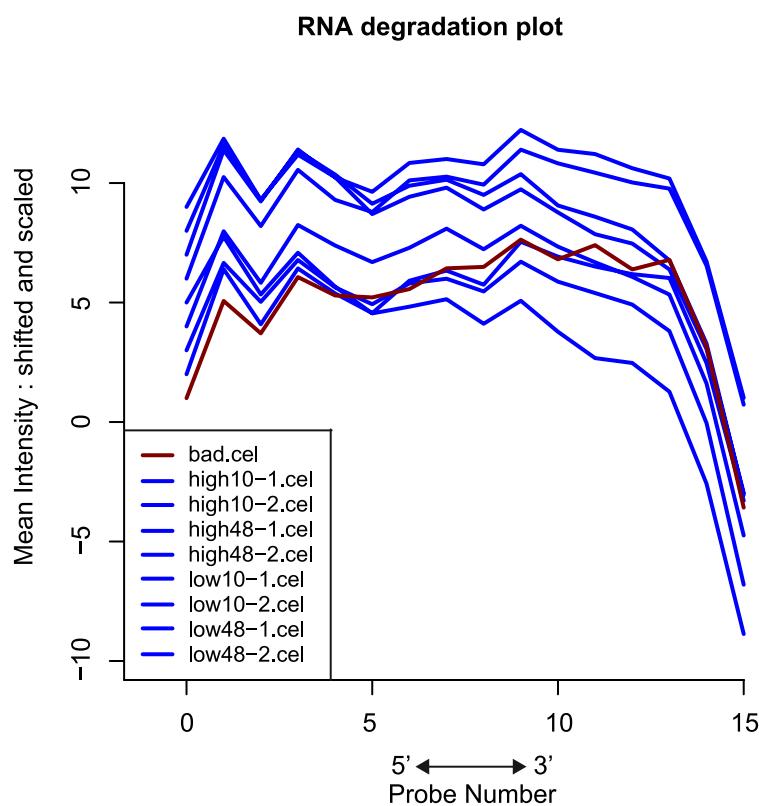


Figura 16. Los gráficos de degradación pueden tomar formas muy distintas. Las líneas pueden ser ascendentes, planas o irregulares, pero lo que indica un posible problema es una línea que cruza las demás, como en este ejemplo, el array "bad.cel".

Modelos de bajo nivel

Los modelos de bajo nivel (*probe-level-models* o PLM) ajustan a los valores de intensidad –a nivel de sondas, no de valores totalizados de gen– un modelo explicativo. Los valores estimados por este modelo se comparan con los valores reales y se obtienen los errores o residuos del ajuste. El análisis de dichos residuos procede de forma similar a lo que se realiza al analizar un modelo de regresión: si los errores no presentan ningún patrón especial, supondremos que el modelo se ajusta relativamente bien. Si en cambio observamos desvia-

ciones de esta presunta aleatoriedad, querrá decir que el modelo no explica bien las observaciones, lo cual se atribuirá a la existencia de algún problema con los datos.

Con los valores ajustados del modelo se calculan dos medidas:

- La expresión relativa en escala logarítmica *relative log expression (RLE)* es una medida estandarizada de la expresión. No es de gran utilidad pero debería presentar una distribución similar en todos los arrays.
- El error no estandarizado y normalizado o *NUSE* es el más informativo, ya que representa la distribución de los residuos a la que hacíamos referencia más arriba. Si un array es problemático la caja correspondiente en el box-plot aparece desplazada hacia arriba o hacia abajo de las demás.

La figura 17 muestra los gráficos RLE y NUSE para el conjunto de datos estrogen. En ambos gráficos puede verse cómo el array defectuoso “bad.cel” queda claramente diferenciado del resto.

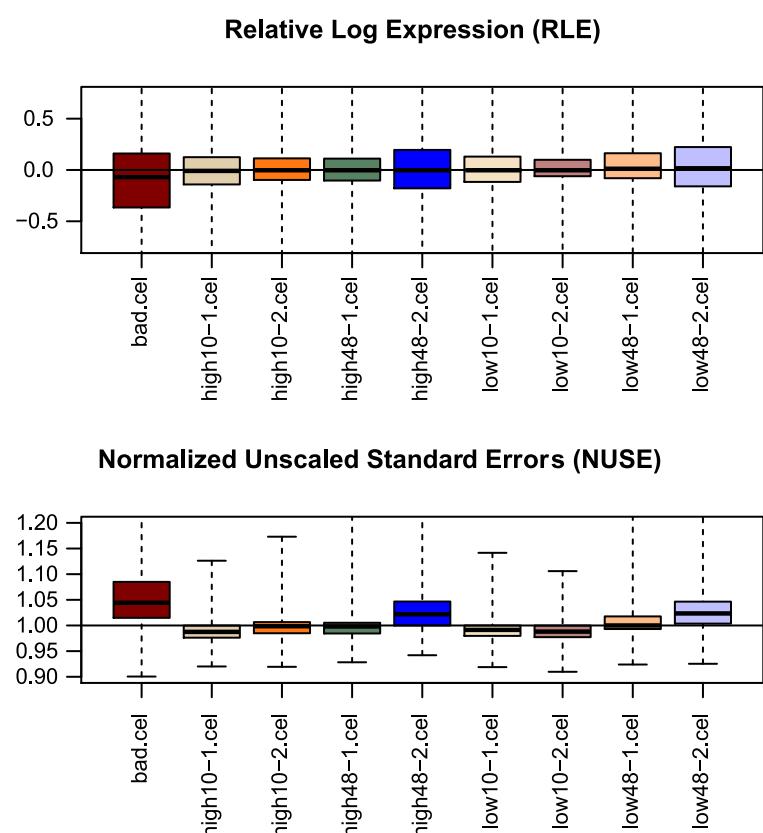


Figura 17. Los gráficos basados en modelos de bajo nivel, RLE (arriba) y NUSE (abajo) permiten detectar arrays problemáticos, como aquellos que difieren del patrón general.

3.3. Normalización de arrays de dos colores

La palabra *normalización* describe las técnicas utilizadas para transformar adecuadamente los datos antes de que sean analizados. El objetivo es corregir diferencias sistemáticas entre muestras, en la misma o entre imágenes, lo que no representa una verdadera variación entre las muestras biológicas.

Estas diferencias sistemáticas pueden deberse, entre otras, a:

- Cambios en la tinción que producen sesgos en la intensidad del *spot*.
- La ubicación en el array que puede afectar al proceso de lectura.
- Un problema en la placa de origen.
- La existencia de diferencias en la calidad de la impresión: pueden presentarse variaciones en los pins y el tiempo de impresión.
- Cambio en los parámetros de la digitalización (escaneo).

A veces puede ser difícil detectar estos problemas, aunque existen algunas formas de saber si es necesario realizar una normalización. Aquí destacamos dos posibilidades:

- 1) Realizar una autohibridación. Si hibridamos una muestra con ella misma, las intensidades deberían ser las mismas en ambos canales. Cualquier desviación de esta igualdad significa que hay un sesgo sistemático.
- 2) Detectar artefactos espaciales en la imagen o en la tinción de los gráficos de diagnóstico

3.3.1. Normalización global

Este método está basado en un ajuste global, es decir, en modificar todos los valores en una cantidad c , estimada de acuerdo a algún criterio. Por ejemplo, si sustraemos una cantidad fija de cada valor de expresión relativa, expresado como un *log-ratio*, tendremos:

$$\log_2 R/G \rightarrow \log_2 R/G - c = \log_2 R/(Gk) \quad (1)$$

Las opciones para k o $c = \log_2 k$ son las siguientes:

- 1) c = mediana o media de *log ratios* para un conjunto concreto de genes ya sean todos los genes, los genes de control o genes control o genes *housekeeping* (cuya expresión se supone que se mantiene constante entre las distintas condiciones en estudio).
- 2) k = razón de intensidades totales en los dos canales.

El objetivo de la normalización global es conseguir que la media –o la media-n, según como se elija la constante normalizadora– de todos los genes del array, una vez ajustados sus valores, sea igual a cero, que es lo que en principio debería valer si la suposición básica de que la mayoría de genes se expresan igual en ambas condiciones fuera cierta.

$$k = \sum R_i / \sum G_i$$

3.3.2. Normalización dependiente de la intensidad

En este caso se realiza una modificación específica para cada valor. Esta modificación se obtiene como una función de la intensidad total del gen ($c = c(A)$).

$$\log_2 R / G \rightarrow \log_2 R / G - c(A) = \log_2 R / (Gk(A)) \quad (2)$$

Una posible estimación de esta función puede hacerse utilizando la función *lowess* (*locally weighted scatterplot smoothing*).

3.4. Resumen (sumarización) y normalización de microarrays de Affymetrix

En los arrays de Affymetrix, como en todos los tipos de microarrays, tras escanear la imagen se obtienen una serie de valores de intensidad de cada elemento del chip. En el caso de estos arrays sabemos que cada valor no corresponde a la expresión de un gen:

- Hay múltiples valores (sondas o probes) por cada gen, que originan un *probeset*.
- Cada grupo de sondas consiste en múltiples pares de sondas, donde cada una puede tener dos elementos: Un *perfect match* que coincide exactamente con el fragmento del gen al que corresponde la sonda. Un *mismatch* que coincide con el anterior salvo por el valor central que se ha sustituido por el nucleótido complementario. Estos *mismatches* se introdujeron en los primeros arrays de Affymetrix para tener una medida de hibridación no específica, pero en las versiones más recientes se han abandonado.

El proceso que convierte las señales individuales en valores de expresión normalizados para cada gen consta de tres etapas:

- 1) Corrección del ruido de fondo o *background*.
- 2) Normalización para hacer los valores comparables.

3) Resumen o summarización de los valores de cada grupo de sondas en un único valor de expresión absoluto normalizado para cada gen.

A menudo los tres pasos se denominan genérica y equivocadamente normalización.

A diferencia de los chips de ADNc, aquí las medidas de expresión son absolutas (no se compara una condición contra otra) dado que cada chip se hibrida con un única muestra.

Hay muchos métodos para estimar la expresión (más de 30 publicados). Cada método contempla de forma explícita o implícita las tres formas de preprocesado: corrección del fondo, normalización y resumen.

Los principales métodos que consideraremos son:

- Microarray Suite (MAS). Método oficial de Affymetrix. Versiones 4.0 y 5.0.
- RMA (Bioconductor). Actualmente es el método estándar.

3.4.1. Métodos originales de Affymetrix

M.A.S. 4.0

Es el primer método introducido por Affymetrix. La corrección del fondo se realiza restando el *perfect match* del *mismatch*:

$$E_j = PM_j - MM_j \quad (3)$$

La normalización se realiza de forma global haciendo transformaciones de modo que la media de todo el chip sea la misma y la summarización se basa en calcular el promedio de las diferencias absolutas ignorando los pares que se desvían más de 3σ (es decir, tres desviaciones estándar) del valor medio μ :

$$Dif.Media = \frac{1}{|A|} \sum_{j \in A} (PM_j - MM_j) \quad (4)$$

Los problemas que presenta este método son:

- 1/3 de los MM son mayores que los PM.
- Pueden aparecer valores MM negativos.
- El uso de los MM añade ruido.

M.A.S. 5.0

Los problemas que presentaba el método M.A.S. 4.0 llevaron a sustituirlo por otra variante, el M.A.S 5.0, llamado así por venir implementado en el software de Affymetrix llamado “MicroArray Suite 5.0”.

Este método utiliza un estadístico robusto, *el biweight de Tukey*, para corregir y ponderar el fondo y calcular (estimar) la señal. El *biweight* de Tukey T_{bi} pondera los valores por su distancia a la mediana, es decir, mide la tendencia central pero realiza un ajuste de *outliers*.

La lógica de este método reside en pensar que el valor de MM no siempre tiene sentido, (p. ej si MM > PM). Dado que esto sucede en ocasiones, se realiza el cambio siguiente:

1) Se introduce el *background* específico de un conjunto de pruebas i de tamaño n basado en los pares de pruebas j :

$$SB_i = T_{bi}(\log(PM_{i,j}) - \log(MM_{i,j})) : j = 1, \dots, n \quad (5)$$

2) SB se utiliza para decidir cómo se ajusta el *background*.

- a) Si es grande los datos suelen ser fiables.
- b) Si es pequeño mejor basarse tan solo en PM.

Este método no solo corrige el *background* sino que también permite normalizar y sumarizar. Para ello se introduce el *mismatch* idealizado (IM), que permite corregir la intensidad de las pruebas individuales. Este método también ha sido muy criticado:

- Se considera que no tiene mucho sentido promediar las pruebas entre arrays, pues estos pueden tener características de hibridación intrínsecamente distintas.
- El método no mejora “aprendiendo” del funcionamiento entre arrays de las pruebas individuales.

3.4.2. El método RMA

Para compensar algunas deficiencias de los primeros métodos de resumen y normalización de arrays de Affymetrix, Irizarry y otros introdujeron en el 2003 ([23]) un método basado en la modelización de las intensidades de las sondas que, en vez de basarse en las distintas sondas de un gen dentro de un mismo array, se basa en los distintos valores de la misma sonda entre todos los arrays disponibles.

Esquemáticamente, los pasos que realiza el método RMA (*robust multi-array average*) son:

- 1) Ajusta el ruido de fondo (*background*) basándose solo en los valores PM y utilizando un modelo estadístico complejo en el que combina la modelización de la señal mediante una distribución exponencial con la del ruido mediante una distribución normal.
- 2) Toma logaritmos base 2 de cada intensidad ajustada por el *background*.
- 3) Realiza una normalización por cuantiles de los valores del paso 2 consistente en sustituir cada valor individual por el que tendría la misma posición en la distribución empírica estimada sobre todas las muestras, es decir, los promedios de las distribuciones de los valores ordenados de cada array (figura 18).
- 4) Estima las intensidades de cada gen separadamente para cada conjunto de sondas. Para ello realiza una técnica similar a una regresión robusta denominada *median polish* sobre una matriz de datos que tiene los arrays en filas y los grupos de sondas en columnas.

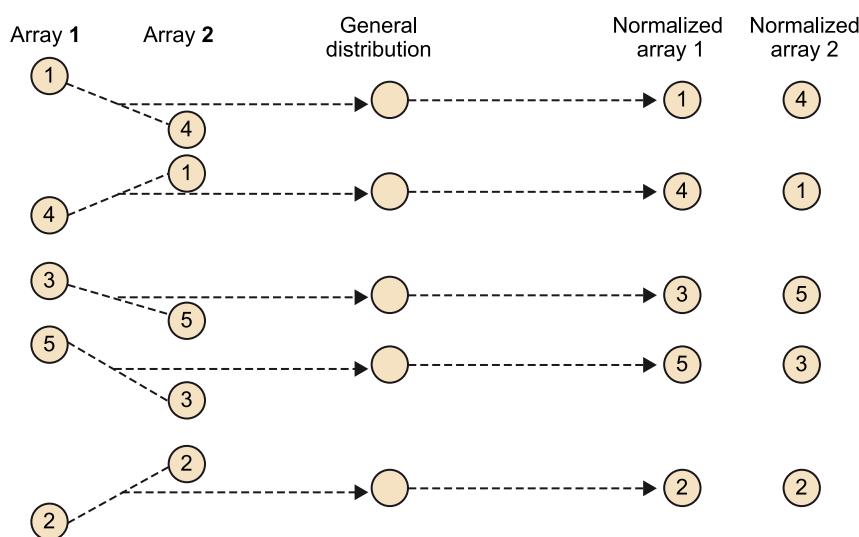


Figura 18. Esquema de la normalización por cuantiles. El método RMA incluye una normalización por cuantiles como la representada esquemáticamente en esta figura.

Como resultado final de todos los pasos anteriores se obtiene la matriz con los datos summarizados y normalizados. A pesar de no estar exento de críticas como la que afirma que este procedimiento compacta los valores reduciendo su variabilidad natural, este método se ha convertido en el estándar *de facto* actualmente por muchos usuarios de Bioconductor.

3.5. Filtraje no específico

El filtraje no específico es recomendable para eliminar el ruido de fondo y limitar los ajustes posteriores a los necesarios. Los principales procesos de filtraje son:

- Eliminación de los spots marcados como erróneos mediante flags y que son debidos a problemas en la hibridación o en el escaneo.
- Eliminación de spots con señales muy bajas debido a problemas en el *spotting* o a que no ha habido hibridación en ese spot (filtraje por señal).
- Eliminación de genes que no presenten una variación significativa en su señal, entre distintas condiciones experimentales (filtraje por variabilidad). Ante la duda, se debe ser conservador y reducir la operación de filtraje al mínimo.

El objetivo del filtraje es eliminar aquellos spots cuyas imágenes o señales sean erróneas por diferentes motivos, disminuyendo el ruido de fondo. Aunque existe controversia a su uso, prefiriendo el no filtraje a eliminar de forma no intencionada spots informativos.

4. Selección de genes diferencialmente expresados

4.1. Introducción

El motivo más habitual por el que se suelen utilizar microarrays es la búsqueda de genes cuya expresión cambia entre dos o más condiciones experimentales, por ejemplo, a consecuencia de un tratamiento, una enfermedad u otras causas (distintos tiempos, distintas líneas celulares, ...).

El problema consiste en identificar estos genes y suele denominarse selección de genes diferencialmente expresados (DEG) o bien comparación de clases.

El problema de seleccionar genes diferencialmente expresados se traduce de manera casi inmediata al problema estadístico de comparar variables y, en años recientes, se han desarrollado un gran número de métodos estadísticos para resolverlo. La mayoría son extensiones de los métodos estadísticos clásicos –pruebas- *t* o análisis de la varianza– adaptados en uno u otro sentido para tener en cuenta las peculiaridades de los microarrays.

Aunque el problema de la selección de genes diferencialmente expresados puede relacionarse directamente con la realización de pruebas estadísticas, en el caso de los microarrays, el hecho de que haya dos tecnologías que miden la expresión de dos formas distintas hace que se deba diferenciar la metodología a emplear en cada caso. Los arrays de dos colores combinan dos muestras en un chip y generan una medida de expresión relativa. Esto hace que, para comparar dos muestras de un mismo individuo, sean la opción naturalmente más apropiada.

En el caso de querer comparar muestras independientes de diferentes individuos, los arrays de un color son la mejor opción. Evidentemente, lo más común será disponer de una sola técnica y tener que adaptar los análisis estadísticos a la misma.

Vamos a plantear un posible esquema de trabajo, para situar la mejor opción en cada caso:

- Situación 1: experimento con 5 individuos diabéticos y 5 no diabéticos, independientes entre sí (muestras independientes).
 - Caso 1: Arrays de cDNA (2 colores): utilizaríamos 5 arrays diabético/referencia y 5 arrays no diabético/referencia

- Caso 2: Arrays de Affymetrix (1 color): utilizaríamos 5 arrays de diabético y 5 arrays de no diabético
- Situación 2: Experimento con 6 individuos de los que se ha tomado una muestra de tejido sano y otra de tejido tumoral (muestras apareadas o dependientes).
 - Caso 3: Arrays de cDNA (2 colores): utilizaríamos 6 arrays, uno por individuo, y en cada uno se realizaría la comparación tejido tumoral/tejido sano.
 - Caso 4: Arrays de Affymetrix (1 color): utilizaríamos 12 arrays, 2 por individuo, 6 con muestras de tejido tumoral y 6 con muestras de tejido sano.

4.1.1. Medidas naturales para comparar dos muestras

Recordemos que una vez se han hecho los experimentos con microarrays y NA, valor detectado por el escáner, algunas operaciones que se realicen tendrán en cuenta esta característica, según si la comparación a realizar se lleva a cabo con datos independientes (2 muestras, casos 1 y 2) o con datos dependientes (muestras apareadas, casos 3, 4). Algunas medidas naturales o razonables para la comparación de expresiones son las siguientes:

1) Para comparaciones directas, con expresiones relativas entre muestras apareadas o bien diferencias apareadas de expresiones absolutas:

a) *log ratio* promedio: $\bar{R} = \frac{1}{n} \sum_{i=1}^n R_i$

Ratio

A la ratio o razón de expresiones se suele denominar también *fold change (FC)* porque en arrays de dos colores representaba cuántas veces más está expresado el gen en una (R) que en otra condición (G). Al *log ratio* también se le llama *logFC* y por extensión, a la diferencia de medias en escala logarítmica también se la denomina *logFC*, dado que se realiza de forma implícita la aproximación siguiente:

$$\bar{X}_1 - \bar{X}_2 = \log Y_1 - \log Y_2 \approx \log(\bar{Y}_1) - \log(\bar{Y}_2) = \log\left(\frac{Y_1}{Y_2}\right)$$

b) *t-test* de una $t = \frac{\bar{R}}{SE}$, donde *SE* estima el error estándar del *log ratio* promedio.

c) *t-test* robusto: que se obtiene sustituyendo en la fórmula del test *t* el estimador del error estándar por una versión robusta del mismo como la *desviación absoluta media* o MAD.

2) Para comparaciones indirectas entre muestras independientes de expresiones relativas o absolutas:

a) Diferencia media $\bar{R}_1 - \bar{R}_2 = \frac{1}{n_1} \sum_{i=1}^{n_1} R_i - \frac{1}{n_2} \sum_{j=1}^{n_2} R_j$

b) t -test (clásico) de dos muestras $t = \frac{\bar{R}_1 - \bar{R}_2}{SE_{12}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$, donde SE_{12} es el error estándar ponderado de la diferencia de medias.

c) t -test robusto de dos muestras: que se obtiene sustituyendo en la fórmula del test t el estimador del error estándar de la diferencia por una versión robusta del mismo.

Un primer ejemplo

Consideremos la tabla 6 que representa una matriz de expresión simplificada que contiene las expresiones relativas (por ejemplo, entre tejido tumoral y sano del mismo individuo) de 5 genes en 6 muestras.

Tabla 6. Ejemplo ficticio y simplificado de una matriz de expresiones relativas.

Gen	R1	R2	R3	R4	R5	R6
A	2.50	2.70	2.50	2.80	3.20	2.00
B	0.01	0.05	-0.05	0.01	0.00	0.00
C	2.50	2.70	2.50	1.80	20.00	1.00
D	0.50	0.00	0.20	0.10	-0.30	0.30
E	0.10	0.11	0.10	0.10	0.11	0.09

Podemos calcular las medidas descritas para el caso de una muestra y decidir si un gen está expresado o no lo está. Se discutirá más adelante cómo precisarlo pero de momento nos quedaremos con la idea de que si la medida escogida es cero o cercana a cero, el gen no está diferencialmente expresado, y si es mayor o menor que cero, sí que lo está. Nos referimos a “cero” porque estamos hablando de logaritmos de razones: si la expresión es la misma en ambas condiciones, el cociente es uno y su logaritmo es cero.

Vale la pena insistir en el concepto de expresión diferencial: no nos preocupa cuál es la expresión del gen en una u otra muestra sino si son distintas.

La tabla 7 sugiere que podría considerarse que el gen A está diferencialmente expresado (promedio y t -test altos), mientras que el gen B o el D no lo están (promedio y test- t próximos a cero). Los genes C y D pueden llevar a conclusiones contradictorias según nos basemos en el promedio o el test t .

Tabla 7. Estadísticos de resumen y de test para los datos del ejemplo de la tabla 6.

Gen	Promedio	Err. std	T-test
A	2.617	0.397	14.735

Gen	Promedio	Err. std	T-test
B	0.003	0.032	0.233
C	5.083	7.335	1.550
D	0.133	0.273	1.091
E	0.102	0.008	30.200

Si se observan los valores del test *t* del gen C se concluye que el gen no aparenta estar diferencialmente expresado. Si en cambio se observa su promedio, parece que sí que lo está.

En el gen E pasa exactamente lo opuesto. La explicación de estas aparentes contradicciones se halla en el error estándar. En el gen C es muy elevado, debido a que el valor (20) es probablemente un valor atípico o *outlier*. En el gen D el error estándar es muy bajo, por lo que, al encontrarse en el denominador del *t*-test, aumenta artificialmente su valor.

4.1.2. Selección de genes diferencialmente expresados

Vamos a plantear la forma como se aborda este problema en el estudio de datos de microarrays. Dadas las características propias de este tipo de datos, se consideran otras formas de estimar el error estándar que no sean tan sensibles a valores extremos o muy bajos.

Consideremos un gen cualquiera que llamaremos *g*. Llamamos:

- R_g log-ratio medio observado.
- SE_g error estándar de R_g estimado a partir de los datos en el gen *g*.
- SE error estándar de R_g estimado a partir de los datos con la información de todos los genes.

Podemos considerar dos variantes para el test-*t*:

1) test-*t* global: se calcula en base a un único estimador de SE para todos los genes:

$$t = R_g / SE,$$

2) test-*t* específico: Utiliza un estimador distinto del error estándar para cada gen:

$$t = R_g / SE_g.$$

Cada aproximación tiene sus pros y sus contras como muestra la tabla 8.

Tabla 8. Ventajas e inconvenientes de los distintos enfoques posibles para la realización de pruebas de significación para seleccionar genes diferencialmente expresados.

Test	Pros	Contras
Test- <i>t</i> Global	Estimador estable de σ	Asume homocedasticidad
Test- <i>t</i> específico	Robusto a heterocedasticidad	Estimador de σ no estable

En la práctica muchos métodos de selección de genes diferencialmente expresados han acabado buscando un compromiso entre ambas aproximaciones para lo que proponen o derivan fórmulas, que de alguna forma, ponderan o combinan dos estimaciones del error estándar: una basada en todos los genes y otra específica de cada gen. La tabla 9 ilustra cómo algunos de los métodos más utilizados en la bibliografía incorporan esta idea. Aunque no entraremos en los detalles de las fórmulas, observad que su estructura es similar: en el numerador se encuentra el *fold-change* y en el denominador una expresión en la que se combina la estimación del error estándar de cada gen SE_g con algo más, ya sea una constante ("c") en el método SAM o una estimación de dicho error obtenida a partir de todos los genes analizados (SE en Cyber-T o SE_0 en el T-moderado).

Tabla 9. SAM, Cyber-T y el T-moderado son tres métodos muy populares para seleccionar genes diferencialmente expresados. Todos ellos realizan algún tipo de moderación de la varianza.

Método (referencia)	Fórmula
SAM (Tibshirani y otros, 2001)	$S = \frac{R_g}{c + SE_g}$
Cyber-T (Baldi y otros, 2001)	$t = \frac{R_g}{\sqrt{\frac{v_0 SE^2 + (n-1)SE_g^2}{v_0 + n - 2}}}$
T-moderado (Smyth, 2004)	$t = \frac{R_g}{\sqrt{\frac{d_0 \cdot SE_0^2 + d \cdot SE_g^2}{d_0 + d}}}$

Al hecho de incluir un coeficiente que tenga en cuenta la variabilidad de todos los genes en el array para estimar el error estándar de cada gen se le denomina moderación de la varianza (*variance shrinkage*) y es una de las aproximaciones en que existe cierto consenso ([2]) acerca de que sirven para mejorar la selección de genes diferencialmente expresados.

Ejemplo de utilización del test-t

Como ejemplo utilizaremos el conjunto de datos `celltypes` y supondremos que disponemos ya de los datos normalizados y filtrados almacenados en un objeto `expressionSet`.

Deseamos comparar la expresión génica en ratones estimulados con LPS frente a los que no han sido tratados. Para ello utilizamos la función `rowttests` del paquete `genefilter`, que realiza un test *t* sobre cada una de los genes a partir de una matriz de expresión y un vector de tipo factor que defina los grupos que se desea comparar (en este caso el campo *treat*).

Ejemplo

El ejemplo que se presenta aquí, así como los siguientes, se basan en un código real, pensado para que pueda ser reproducido. Para que esto sea posible se utilizan instrucciones y objetos de Bioconductor que simplifican el cálculo, aunque hacen que el código resulte más difícil de entender, especialmente si no se conocen las principales clases y métodos definidos en Bioconductor.

En consecuencia, nuestro consejo es que, en una primera lectura, se utilicen algunos códigos como caja negra para volver luego a ellos una vez se conozcan y comprendan las instrucciones que aparecen en el código.

```
> stopifnot(require(Biobase))
> load ("celltypes-normalized.rma.Rda")
> my.eset <- eset_rma_filtered
> grupo_1 <- as.factor(pData(my.eset)$treat)
> stopifnot(require(genefilter))
> teststat <- rowttests(my.eset, "treat")
> print(teststat[1:5,])
```

	statistic	dm	p.value
1415694_at	3.503426	0.9813765	5.693821e-03
1415698_at	-4.253446	-0.9200460	1.680334e-03
1415743_at	-15.642231	-1.0156246	2.335398e-08
1415760_s_at	-5.686252	-0.8663532	2.020937e-04
1415772_at	11.394671	1.1306309	4.745989e-07

Cuanto mayor sea el valor absoluto del estadístico *t* mayor es la probabilidad de que el gen esté diferencialmente expresado.

Genes diferencialmente expresados estadísticamente significativos

Como hemos visto en el subapartado anterior, dos medidas naturales para la selección de genes son el promedio de *log-ratios* –o la diferencia de promedios en el caso de muestras independientes– o el valor del estadístico de test (*t*-test) de una o dos muestras según si se trata de muestras apareadas o independientes respectivamente. Los primeros estudios de microarrays eran muy costosos y se

hacían con pocas o incluso ninguna réplica por condición experimental. En estas situaciones la única forma fiable de detectar una diferencia de expresión era a través del *log-ratio* o sus diferencias.

Rápidamente se puso en evidencia que para poder obtener los genes que estaban realmente diferencialmente expresados era preciso disponer de un soporte estadístico que permitiera tener en cuenta la variación aleatoria existente entre muestras.

En la práctica esto se reduce a afirmar que, si además de la diferencia de expresión entre las condiciones experimentales, se lleva a cabo un test estadístico, dispondremos de una medida objetiva, el *p*-valor que nos servirá para decidir qué genes se declaran diferencialmente expresados, a saber, aquellos en los que el *p*-valor del test sea inferior a un cierto umbral como 0.05 o 0.01.

Tal como se ha indicado anteriormente, un test estadístico procede decidiendo rechazar la hipótesis nula si el *p*-valor es más pequeño que el nivel de significación del test.

Siguiendo con el ejemplo anterior podemos ordenar los resultados de los tests en base a los *p*-valores:

```
> ranked <- teststat[order(teststat$p.value),]
> print(ranked[1:5,])

      statistic        dm     p.value
1449383_at   -48.61014 -2.044928 3.276616e-13
1430127_a_at    46.90904  1.508843 4.672074e-13
1451421_a_at   -40.72173 -1.445182 1.909283e-12
1450826_a_at    37.62239  5.147992 4.193941e-12
1416122_at     34.66314  1.482676 9.460684e-12
```

Ahora podríamos seleccionar, por ejemplo, los genes cuyo *p*-valor fuera inferior a 0.01.

```
> selectedTeststat <- ranked[ranked$p.value < 0.01, ]
```

Esto deja un total de 1.594 con un *p*-valor inferior a 0.01.

Gráficos de volcan o *volcano plots*

Si se opta por calcular los valores de significación (*p*-valores) de los genes, resulta interesante relacionar el tamaño del efecto (*effect size*, reflejado en la diferencia de medias o *fold change*) con la significación estadística. El *volcano*

plot es una representación gráfica que permite ordenar los genes a lo largo de dos dimensiones, la biológica, representada por el *fold change* y la estadística representada por el logaritmo negativo del *p*-valor.

En la escala horizontal se representa el cambio entre los dos grupos (en escala logarítmica, de manera que la regulación positiva o negativa se representa de forma simétrica). En la escala vertical se representa el *p*-valor del test en una escala logarítmica negativa, de modo que los *p*-valores más pequeños aparecen mayores.

Así pues, puede considerarse que el primer eje indica impacto biológico del cambio (a más efecto biológico mayor *fold-change*) y el segundo muestra la evidencia estadística, o la fiabilidad del cambio.

La figura 19 muestra un *volcano plot* para ejemplo de los *celltypes*.

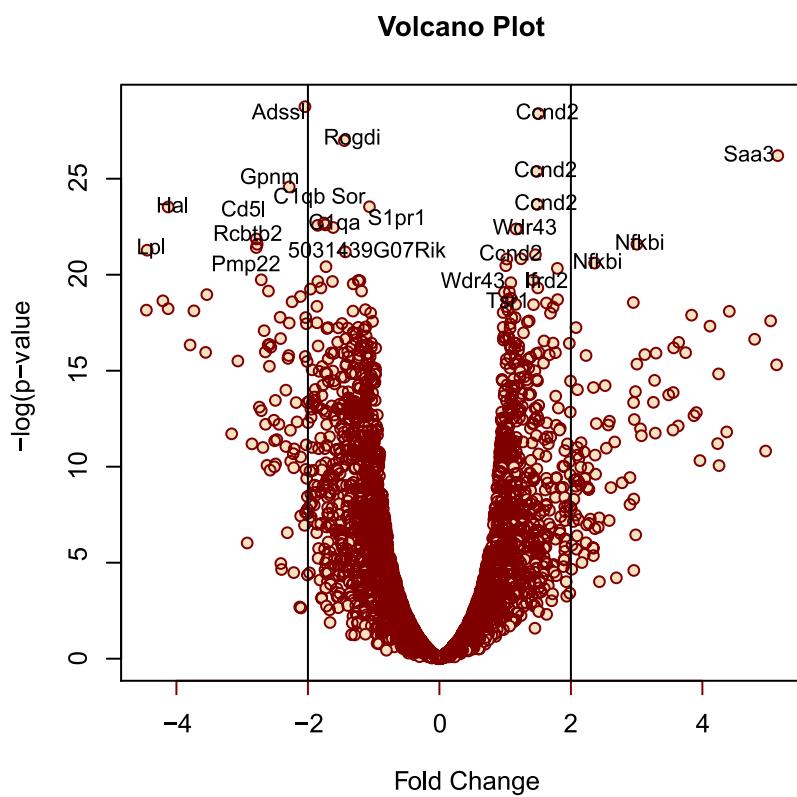


Figura 19. *Volcano plot*.
Ejemplo de *Volcano plot* que muestra los genes candidatos a considerarse como diferencialmente expresados en la comparación “LPS” frente a “Medium”. Cuanto más “hacia arriba” y “hacia afuera” se encuentra un gen, más fuerte es la evidencia a favor de que esté diferencialmente expresado.

4.1.3. Potencia y tamaño muestral

Tal como se ha indicado más arriba, para realizar un buen test suele controlarse la probabilidad de error de tipo I (de falsos positivos) y buscar, de entre los tests candidatos, aquellos que tengan una menor probabilidad de error de tipo II, o equivalentemente, una mayor potencia. A partir de este planteamiento,

existe, en el contexto estadístico estándar, multitud de formas de determinar cuál debe ser el tamaño muestral necesario para obtener una potencia dada fijados el tamaño de efecto (*fold-change*) y la probabilidad de error de tipo I.

En el caso de los microarrays, se han desarrollado diversas fórmulas para realizar cálculos de este tipo, pero la compleja estructura de los datos microarrays hace que sean relativamente discutibles.

Simon ([41]) sugiere la fórmula siguiente, que es una generalización de las fórmulas clásicas para problemas de dos muestras.

El tamaño total requerido para detectar genes diferencialmente expresados en al menos una diferencia δ con una probabilidad de error de tipo I (FP), α y una probabilidad de error de tipo II (FN) $1 - \beta$ se calcula:

$$n = \frac{4(z_{\alpha/2} + z_{\beta})^2}{(\delta/\sigma)^2},$$

donde z_{α} y z_{β} son los percentiles $100\alpha/2$ y 100β de la distribución normal $N(0,1)$ y σ es la desviación estándar de un gen dentro de una clase (de un grupo). Obviamente, σ es siempre desconocida, por lo que, sin una muestra piloto con que estimarla, el cálculo es más imaginativo que realista.

Además de esto, el número de arrays usualmente recomendado queda lejos de la cantidad asequible para la mayor parte de los experimentos ([32], [45]). Lo que muchos usuarios hacen a la práctica es buscar un equilibrio entre los costes y la reproducibilidad, y, de hecho, tienden a usar una cantidad fija de arrays, tal como 3 o 5 sin consideraciones adicionales.

Por ejemplo, si ponemos $\alpha = 0.001$, $1 - \beta = 0.95$, $\delta = 1$ y estimamos σ entre todas las muestras, el número de réplicas biológicas que necesitaremos será de 35.8.

4.1.4. El problema de la multiplicidad de tests

El análisis de microarrays se realiza gen a gen e involucra múltiples tests (en inglés, *multiple testing*), miles, probablemente. Esto significa que, a medida que crece el número de genes, la probabilidad de declarar erróneamente al menos un gen diferencialmente expresado va en aumento, y si no se realiza algún tipo de ajuste, el número de falsos positivos será tanto más alto cuantos más genes estemos analizando.

Hay muchas formas de intentar controlar estas probabilidades de error y puede verse una excelente revisión en Dudoit ([13]).

De forma simplificada consideramos las dos aproximaciones más utilizadas.

Una posibilidad es mirar de controlar la probabilidad de obtener al menos un falso positivo o *family-wise-error-rate (FWER)*. El más popular de estos métodos de control es la corrección de Bonferroni, consistente en multiplicar el *p*-valor por el número de tests realizados. La misma Dudoit y muchos otros autores han desarrollado variantes de los métodos clásicos de ajuste FWER usando por ejemplo tests de permutaciones.

El criterio FWER es quizás demasiado restrictivo, dado que el control de los falsos positivos implica un considerable incremento de falsos negativos. En la práctica, sin embargo, muchos biólogos parecen estar dispuestos a aceptar que se produzcan algunos errores, siempre y cuando esto permita realizar descubrimientos. Por ejemplo, un investigador debe considerar aceptable cierta pequeña proporción de errores (digamos del 10 al 20%) entre sus descubrimientos. En este caso, el investigador está expresando interés en controlar el porcentaje de falsos descubrimientos (FDR), es decir, lo que es la proporción de falsos positivos sobre el total de genes inicialmente identificados como expresados diferencialmente. A diferencia del nivel de significación que queda determinado antes de examinar los datos, la FDR es una medida de confianza *a posteriori*, ya que emplea información disponible en los datos para estimar las proporciones de falsos positivos que han tenido lugar. Si se obtiene una lista de los genes expresados diferencialmente en los que el FDR se controla hasta, digamos, el 20%, cabe esperar que el 20% de estos genes representen falsos positivos. Lo cual supone un enfoque menos restrictivo que controlar el FWER.

La decisión de controlar el FDR o el FWER depende de los objetivos del experimento. Si, por ejemplo, el objetivo es la captura de genes, es razonable permitir cierta cantidad de falsos positivos y es preferible seleccionar FDR. Si por el contrario se trabaja con una lista de un tamaño menor al deseado para verificar la expresión de ciertos genes específicos, entonces el FWER es el criterio apropiado.

El ejemplo siguiente muestra cómo se realiza el ajuste de *p*-valores usando el paquete `multtest` de Bioconductor para ajustar por los métodos de Bonferroni (FWER), Benjamini & Hochberg (FDR) o Benjamini & Yekutieli (BY, FDR).

```
> stopifnot(require(multtest))
> procs <- c("Bonferroni", "BH", "BY")
> adjPvalues <- mt.rawp2adjp(teststat$p.value, procs)
> names(adjPvalues)

[1] "adjp" "index" "h0.ABH" "h0.TSBH"

> ranked.adjusted<-cbind(ranked, adjPvalues$adjp)
> head(ranked.adjusted)
      statistic        dm      p.value       rawp   Bonferroni
1449383_at -48.61014 -2.044928 3.276616e-13 3.276616e-13 1.478082e-09
1430127_a_at  46.90904  1.508843 4.672074e-13 4.672074e-13 2.107573e-09
```

```

1451421_a_at -40.72173 -1.445182 1.909283e-12 1.909283e-12 8.612774e-09
1450826_a_at 37.62239 5.147992 4.193941e-12 4.193941e-12 1.891887e-08
1416122_at 34.66314 1.482676 9.460684e-12 9.460684e-12 4.267714e-08
1448303_at -31.92365 -2.283765 2.140612e-11 2.140612e-11 9.656299e-08
BH BY
1449383_at 1.053786e-09 9.475226e-09
1430127_a_at 1.053786e-09 9.475226e-09
1451421_a_at 2.870925e-09 2.581421e-08
1450826_a_at 4.729717e-09 4.252773e-08
1416122_at 8.535429e-09 7.674717e-08
1448303_at 1.609383e-08 1.447093e-07

```

Si seleccionamos los genes en base a su *p*-valor ajustado, por ejemplo por el método de Benjamini y Yekutieli, se obtienen los siguientes genes:

```

> selectedAdjusted<-ranked.adjusted[ranked.adjusted$BY<0.001,]
> nrow(selectedAdjusted)

[1] 449

```

4.2. Modelos lineales para la selección de genes: limma

En el subapartado anterior se ha discutido el uso del test *t* y sus extensiones para la selección de genes diferencialmente expresados en situaciones relativamente sencillas, es decir, cuando solo hay dos grupos.

En muchos estudios el número de grupos a considerar es más de dos y las fuentes de variabilidad pueden ser más de una, por ejemplo, una puede ser el tratamiento pero otra puede ser la edad de los individuos o cualquier otra condición fijada por el investigador o derivada de la heterogeneidad de las muestras.

En estas situaciones una aproximación razonable en problemas con una variable respuesta es el análisis de la varianza. En este subapartado se presenta una aproximación equivalente que, de forma general, se denomina el modelo lineal. Este método —que engloba el análisis de la varianza y la regresión— es uno de los más usados en estadística y ha sido popularizado en el campo de microarrays gracias a los trabajos de Gordon Smyth, quien ha creado el paquete *limma*, que se ha convertido en la herramienta más utilizada para el análisis de microarrays.

4.2.1. El modelo lineal general

El modelo lineal (ver por ejemplo Faraway, [16]) es un marco general para la modelización y el análisis de datos de estadística.

El método consiste en asumir una relación lineal entre los valores observados de una variable *respuesta* y las condiciones experimentales. A partir de aquí se obtienen estimadores para los parámetros del modelo y de sus errores estándar, y (con algunas condiciones extra) es posible hacer inferencia acerca del experimento.

La aplicación de modelos lineales puede ser vista como un proceso secuencial, con los siguientes pasos:

- 1) Especificar el diseño del experimento: qué muestras se asignan a qué condiciones.
- 2) (Re-)Escribir un modelo lineal para este diseño en forma de $Y = X\beta + \epsilon$, donde X se denomina la matriz de diseño.
- 3) Una vez que el modelo se ha especificado, se puede aplicar la teoría general para estimar los parámetros y los contrastes (comparaciones entre los valores de los parámetros).
- 4) Si se cumplen ciertas condiciones de validez para el modelo, es posible realizar inferencia sobre los parámetros del mismo, es decir, se pueden contrastar hipótesis sobre dichos parámetros.

El esquema anterior se puede aplicar a casi cualquier tipo de situación experimental. En el subapartado siguiente se presentan un par de ellas.

4.2.2. Ejemplos de situaciones modelizables linealmente

Ejemplo 1: Experimento *Swirl-Zebrafish*

Swirl es una mutación puntual que provoca defectos en la organización del embrión en desarrollo a lo largo de su eje dorsal-ventral. Como resultado, algunos tipos celulares se reducen y otros se expanden. Un objetivo de este trabajo fue identificar los genes con expresión alterada en el mutante *Swirl* en comparación con el tipo salvaje.

- 1) El diseño experimental para este estudio fue el siguiente:

Tabla 10

Slide	Cy3	Cy5
1	W	M
2	M	W
3	W	M
4	M	W

- Cada microarray contenía 8.848 sondas de cDNA (genes o secuencias EST).
- Cuatro réplicas por array (*slide*): 2 juegos de pares de intercambio de color.
- El cDNA del mutante *swirl* (*S*) se marca con Cy5 o Cy3, y el cDNA del *wild type* se marca con el otro.

2) El modelo lineal derivado del diseño anterior fue:

a) El parámetro de interés es: $\alpha = \mathbf{E} \left(\log \frac{S}{W} \right)$.

b) Las muestras 1 y 3 están marcadas con *S* (verde = *green*) y *W* (rojo = *red*), y las muestras 2 y 4 son las mismas con los colorantes intercambiados (*dye-swapped*).

c) El modelo, $\mathbf{y} = \mathbf{X}\alpha + \boldsymbol{\epsilon}$, es:

$$\begin{aligned} y_1 &= \alpha + \varepsilon_1 \\ y_2 &= -\alpha + \varepsilon_2 \\ y_3 &= \alpha + \varepsilon_3 \\ y_4 &= -\alpha + \varepsilon_4 \end{aligned} \Rightarrow \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}}_{\text{Matriz de diseño, } \mathbf{X}} \alpha + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \end{pmatrix} \quad (6)$$

4.2.3. Ejemplo 2: Comparación de tres grupos

Los plásmidos IncHI codifican genes de resistencia múltiple a los antibióticos en *S enterica*.

El plásmido R27 de la cepa salvaje es termosensible al transferirse.

Algunos fenotipos mutantes relacionados con *hha* y *hns* cromosómicos participan en diferentes procesos metabólicos de interés en la conjugación termorregulada.

El objetivo del experimento es encontrar qué genes se expresan diferencialmente en tres tipos de mutantes diferentes, M_1 , M_2 y M_3 .

Posibles estrategias de diseño

Este experimento debe ser planteado de forma diferente según el tipo de arrays (uno o dos colores) y qué comparaciones son las de mayor interés.

- Array de dos colores
 - Diseño de referencia: hibridar cada mutante (M_i) frente a salvaje (“Wild type”) (W).

- Diseño loop: hibridar cada mutante el uno con el otro en un doble loop que incluye intercambio de colorantes o (*dye-swapping*).
- Array de un color: hibridar mutantes y salvajes separadamente.

Representación del diseño de referencia

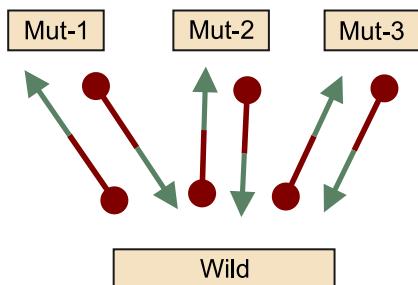


Figura 20. Diseño de referencia para comparar los tres mutantes. Cada mutante se ha hibridado con una muestra de referencia (salvaje) y las comparaciones entre ellos se realizarán de forma indirecta.

- Permite la comparación directa de mutantes frente a salvajes, aunque los mutantes tienen que compararse entre ellos de forma indirecta.
- El número de parámetros a estimar es 3, relación intuitiva entre el número de parámetros y mutantes.
- Las comparaciones de mutante frente a mutante son menos eficientes.

Representación del diseño de *loop* (bucle)

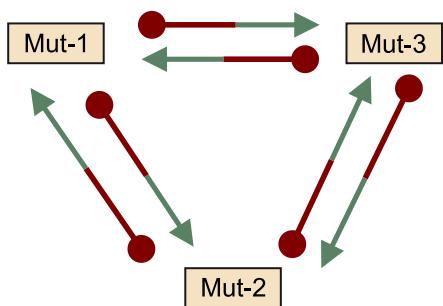


Figura 21. Diseño en bucle para comparar los tres mutantes. Cada mutante se ha hibridado con otro mutante con lo que las comparaciones entre ellos se realizarán de forma directa

- Permite la comparación directa de mutante frente a mutante.
- El número de parámetros a estimar es 2. Menos intuitivo.
- Las comparaciones mutante frente a mutante son más eficientes.

Representación del diseño mediante arrays de un color

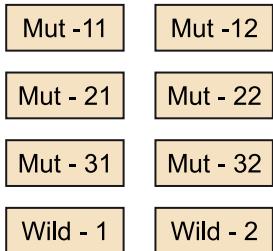


Figura 22. Si se utilizan arrays de un color, se pueden realizar todas las comparaciones con la misma eficiencia.

- Permite la comparación directa de:
 - mutante frente a *wild*
 - mutante frente a mutante
- El número de parámetros a estimar es 4.
- Todas las comparaciones se pueden hacer de forma igualmente eficiente.

Modelo lineal para el diseño de referencia

Modelo, $\mathbf{y} = \mathbf{X}\boldsymbol{\alpha} + \boldsymbol{\epsilon}$, y contrastes $\mathbf{C}'\boldsymbol{\beta}$

1) Parámetros del modelo:

$$\alpha_1 = \mathbb{E} \left(\log \frac{M_1}{W} \right), \alpha_2 = \mathbb{E} \left(\log \frac{M_2}{W} \right), \alpha_3 = \mathbb{E} \left(\log \frac{M_3}{W} \right).$$

2) Contrastos: Comparaciones interesantes.

$$\begin{aligned} \beta_1 &= \alpha_1 - \alpha_2 \\ \beta_2 &= \alpha_1 - \alpha_3 \\ \beta_3 &= \alpha_2 - \alpha_3 \end{aligned}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}}_{\text{Matriz de diseño, } \mathbf{X}} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{pmatrix} \quad (7)$$

$$\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix}}_{\text{Matriz de contraste, } \mathbf{C}} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix}. \quad (8)$$

Modelo lineal para el diseño de *loop*

Modelo, $\mathbf{y} = \mathbf{X}\alpha + \epsilon$, y contrastes $\mathbf{C}'\beta$

1) Parámetros del modelo:

$$\alpha_1 = \mathbf{E} \left(\log \frac{M_1}{M_2} \right), \alpha_2 = \mathbf{E} \left(\log \frac{M_2}{M_3} \right).$$

$$\alpha_3 \text{ no es necesaria: } \log \left(\frac{M_1}{M_3} \right) = \log \left(\frac{M_1}{M_2} \right) - \log \left(\frac{M_2}{M_3} \right).$$

2) Contrastos: Algunas de las comparaciones deseadas son precisamente los parámetros.

$$\begin{aligned}\beta_1 &= \alpha_1, \\ \beta_2 &= \alpha_2, \\ \beta_3 &= \alpha_1 + \alpha_2.\end{aligned}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -1 \\ -1 & 0 \\ 0 & -1 \\ -1 & 1 \end{pmatrix}}_{\text{Matriz de diseño, } \mathbf{X}} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{pmatrix} \quad (9)$$

$$\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}}_{\text{Matriz de contraste, } \mathbf{C}} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}. \quad (10)$$

Modelo lineal para el diseño del array de un color

Modelo, $\mathbf{y} = \mathbf{X}\alpha + \epsilon$, y contrastes $\mathbf{C}^1'\beta, \mathbf{C}^2'\beta$

1) Parámetros del modelo:

$$\alpha_1 = \mathbf{E} (\log M_1), \alpha_2 = \mathbf{E} (\log M_2), \alpha_3 = \mathbf{E} (\log M_3), \alpha_4 = \mathbf{E} (\log W).$$

2) Contrastos: Dos posibles conjuntos de comparaciones interesantes.

a) Comparación entre tipos de mutantes ($\mathbf{C}^1'\beta$)

$$\begin{aligned}\beta_1^1 &= \alpha_1 - \alpha_2, \\ \beta_2^1 &= \alpha_3 - \alpha_2, \\ \beta_3^1 &= \alpha_2 - \alpha_3.\end{aligned}$$

b) Comparación entre cada mutantes y el tipo salvaje ($C^2\beta$)

$$\begin{aligned}\beta_1^2 &= \alpha_4 - \alpha_1, \\ \beta_2^2 &= \alpha_3 - \alpha_1, \\ \beta_3^2 &= \alpha_2 - \alpha_1.\end{aligned}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{Matriz de diseño, } X} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_7 \\ \varepsilon_8 \end{pmatrix} \quad (11)$$

$$\begin{pmatrix} \beta_1^1 \\ \beta_2^1 \\ \beta_3^1 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 \end{pmatrix}}_{\text{Matriz de contraste, } C^1} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix}. \quad (12)$$

$$\begin{pmatrix} \beta_1^2 \\ \beta_2^2 \\ \beta_3^2 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{pmatrix}}_{\text{Matriz de contraste, } C^2} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix}. \quad (13)$$

Ejemplo 3: Estudio de la influencia de las citoquinas en ratones viejos

El objetivo de este experimento es estudiar el efecto de la estimulación mediante lipopolisacáridos (LPS) sobre la regulación mediante citoquinas y ver cómo esta relación se ve afectada por la edad.

Se trata de un modelo de un factor (tratamiento asignable por el individuo) y un bloque (edad no assignable, prefijada en cada ratón). En la práctica, el modelo equivale al del análisis de la varianza de dos factores.

1) El diseño experimental para este estudio fue el siguiente:

Tabla 11: El diseño experimental en el estudio "celltypes" se define a través de la información contenida en la tabla.

Array	Tratamiento	Edad
1	LPS	Viejo
2	LPS	Joven
3	Medio	Viejo
4	Medio	Joven

Array	Tratamiento	Edad
5	LPS	Viejo
6	LPS	Joven
7	Medio	Viejo
8	Medio	Joven
9	LPS	Viejo
10	LPS	Joven
11	Medio	Viejo
12	Medio	Joven

- Se utilizaron microarrays de un color (Affymetrix).
- Cada condición se replicó tres veces.
- Las preguntas específicas a responder:
 - ¿Cuál es el efecto del tratamiento en ratones viejos?
 - ¿Cuál es el efecto del tratamiento en ratones jóvenes?
 - ¿En qué genes el efecto es diferente?

2) En este caso se pueden considerar distintos modelos lineales derivados del hecho de que este experimento admite diferente parametrizaciones:

a) Factores separados con 2 niveles cada uno para tratamiento (LPS/Med), edad (joven/viejo) y su interacción:

$$Y_{ijk} = \underbrace{\alpha_i}_{\text{Trat}} + \underbrace{\beta_j}_{\text{Edad}} + \underbrace{\gamma_{ij}}_{\text{Interacción}} + \varepsilon_{ijk}, \quad i = 1, 2, j = 1, 2, k = 1, 2, 3$$

Esta primera parametrización parece natural, pero es más complicado basarse en ella para responder a las preguntas propuestas.

b) Un factor combinado con 4 niveles:

(LPS.Aged, Med.Aged, LPS.Young, Med.Young)

$$Y_{ij} = \alpha_i + \varepsilon_{ij}, \quad i = 1, \dots, 4, j = 1, 2, 3.$$

Esta parametrización parece más rígida, pero se adapta mejor para responder a las preguntas planteadas.

Aquí se adopta la segunda parametrización.

c) Parámetros del modelo:

$$\begin{aligned}\alpha_1 &= \mathbf{E}(\log LPS.Aged), \alpha_2 = \mathbf{E}(\log Med.Aged), \\ \alpha_3 &= \mathbf{E}(\log LPS.Young), \alpha_4 = \mathbf{E}(\log Med.Young).\end{aligned}$$

d) *Contrastes*: Preguntas que interesa responder:

$$\begin{aligned}\beta_1^1 &= \alpha_3 - \alpha_1, \text{ Efecto del tratamiento en ratones viejos} \\ \beta_2^1 &= \alpha_4 - \alpha_2, \text{ Efecto del tratamiento en ratones jóvenes} \\ \beta_3^1 &= (\alpha_3 - \alpha_1) - (\alpha_2 - \alpha_4), \text{ Interacción: diferencia entre efectos}\end{aligned}$$

e) Modelo lineal:

Modelo, $\mathbf{y} = \mathbf{X}\boldsymbol{\alpha} + \boldsymbol{\epsilon}$, y contrastes $\mathbf{C}'\boldsymbol{\beta}$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{Matriz de Diseño, } \mathbf{X}} + \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \\ \epsilon_7 \\ \epsilon_8 \end{pmatrix} \quad (14)$$

$$\begin{pmatrix} \beta_1^1 \\ \beta_2^1 \\ \beta_3^1 \end{pmatrix} = \underbrace{\begin{pmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ -1 & 1 & 1 & -1 \end{pmatrix}}_{\text{Matriz de contraste, } \mathbf{C}} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} \quad (15)$$

4.2.4. Estimación e inferencia con el modelo lineal

Una vez se ha expresado el experimento como un modelo lineal:

$$\mathbf{E}(\mathbf{y}_g) = \mathbf{X}\boldsymbol{\alpha}_g, \quad \text{var}(\mathbf{y}_g) = W_g \sigma_g, \quad (16)$$

es posible usar la teoría estándar del modelo lineal (ver [16]) para obtener:

- Estimación de los parámetros: $\hat{\alpha}_g (\approx \alpha)$.
- Desviación estándar de las estimaciones: $\hat{s}_g = s_g (\approx \sigma)$.
- Error estándar de las estimaciones: $\widehat{\text{var}}\hat{\alpha}_g = V_g s_g^2$.

Estas estimaciones son la base para realizar inferencia sobre α es decir, para contrastar la hipótesis $H_0: \alpha = 0$, basado en el hecho de que:

$$t_{gj} = \frac{\alpha_{gj}}{s_g \sqrt{v_{gj}}} \sim \text{Distribución de Student.} \quad (17)$$

De forma análoga pueden derivarse fórmulas para $\alpha_1 - \alpha_2$, es decir, para decidir acerca de las comparaciones.

Los procedimientos de estimación y de inferencia no dependen de qué parametrización se ha adoptado, a pesar de que distintas parametrizaciones puedan dar lugar a distintos valores numéricos.

Fortaleza y debilidades del modelo lineal

El enfoque del modelo lineal es flexible y potente:

- Se puede adaptar a situaciones diferentes y complejas.
- Siempre produce buenas estimaciones (Un resultado muy conocido, el teorema de Gauss-Markov, establece que estas estimaciones son las mejores posibles entre las que proporcionan los estimadores lineales insesgados).
- Si las suposiciones son ciertas proporciona una base para la inferencia.

Por otro lado hay que tener en cuenta que, si las suposiciones no se cumplen, entonces las conclusiones deben tomarse con precaución.

Y lo que es peor, aun siendo válidas las suposiciones del modelo, los resultados pueden verse afectados por el tamaño de la muestra de manera que, en muestras pequeñas donde puede haber variaciones más grandes, es fácil que se obtengan valores t no significativos o, por el contrario, excesivamente significativos (si la variación es muy reducida).

La metodología desarrollada por Smyth ([42]) basada en los resultados de Löns-ted & Speed ([44]) está dirigida a abordar cómo hacer frente a estas debilidades.

4.2.5. Modelos lineales para microarrays

Smyth ([42]) considera el problema de la identificación de genes que se expresan diferencialmente en las condiciones especificadas en el diseño de experimentos de microarrays. Como hemos dicho repetidamente, la variabilidad de los valores de expresión difiere entre genes, pero la naturaleza paralela de la inferencia en microarrays sugiere la posibilidad de usar la información de todos los genes a la vez para mejorar la estimación de los parámetros, lo que puede llevar a una inferencia más fiable.

Básicamente, lo que propone Smyth ([42]) es una solución en tres pasos:

1) Se plantea el problema como un modelo lineal con una componente bayesiana, ya que se supone que los mismos parámetros a estimar son variables (no constantes) con distribuciones *prior* que se estimarán a partir de la información de todos los genes.

2) A continuación se obtienen las estimaciones de los parámetros del modelo. La aproximación utilizada garantiza que estos estimadores tienen un comportamiento robusto incluso para un pequeño número de arrays.

3) Finalmente se calcula un *odds-ratio* que viene a ser la razón entre probabilidad de que un gen esté diferencialmente expresado frente a la de que no lo esté y se asocia este valor, denominado estadístico B , con un estadístico t moderado y su p -valor.

$$B = \log \frac{P[\text{Afectado} | M_{ij}]}{P[\text{No Afectado} | M_{ij}]},$$

gen = i ($i = 1 \dots N$), réplica = j ($j = 1, \dots, n$), M_{ij} representa la expresión del gen i en la réplica j .

El hecho de trabajar con logaritmos permite poner el punto de corte en el cero: A mayor valor positivo más probable es que el gen esté diferencialmente expresado. A mayor valor negativo, más probable es que no lo esté.

4.2.6. Implementación y ejemplos

El paquete de Bioconductor `limma` al que hemos hecho referencia en los párrafos anteriores implementa el método de Smyth [37] para la regularización de la varianza, lo que lo ha convertido en muy popular entre los usuarios de microarrays.

El código siguiente muestra cómo se crea la matriz del diseño y los contrastes para realizar el análisis mediante modelos lineales:

```
require(Biobase)
require(limma)
load ("celltypes-normalized.rma.Rda")
my.eset <- eset_rma_filtered
targets <- pData(my.eset)
age.treat<- paste(targets$age,targets$treat, sep=".") 
lev<-factor(age.treat, levels=unique(age.treat))
design <-model.matrix(~0+lev)
colnames(design)<-levels(lev)
rownames(design) <-rownames(targets)
print(design)

Aged.LPS Aged.MED Young.LPS Young.MED
```

```

Aged_LPS_80L.CEL    1      0      0      0
Aged_LPS_86L.CEL    1      0      0      0
Aged_LPS_88L.CEL    1      0      0      0
Aged_Medium_81m.CEL 0      1      0      0
Aged_Medium_82m.CEL 0      1      0      0
Aged_Medium_84m.CEL 0      1      0      0
Young_LPS_75L.CEL   0      0      1      0
Young_LPS_76L.CEL   0      0      1      0
Young_LPS_77L.CEL   0      0      1      0
Young_Medium_71m.CEL 0      0      0      1
Young_Medium_72m.CEL 0      0      0      1
Young_Medium_73m.CEL 0      0      0      1

attr(),"assign")
[1] 1 1 1 1
attr(),"contrasts")
attr(),"contrasts")$lev
[1] "contr.treatment"

> require(limma)
> cont.matrix <- makeContrasts (
+     LPS.in.AGED=(Aged.LPS-Aged.MED),
+     LPS.in.YOUNG=(Young.LPS-Young.MED),
+     AGE=(Aged.MED-Young.MED),
+     levels=design)
> cont.matrix

    Contrasts
    Levels      LPS.in.AGED LPS.in.YOUNG AGE
    Aged.LPS        1          0    0
    Aged.MED       -1          0    1
    Young.LPS       0          1    0
    Young.MED       0          -1   -1

```

Una vez definida la matriz de diseño y los contrastes, podemos pasar a estimar el modelo, estimar los contrastes y realizar las pruebas de significación que nos indiquen, para cada gen y cada comparación, si puede considerarse diferencialmente expresado.

```

require(limma)
fit<-lmFit(my.eset, design)
fit.main<-contrasts.fit(fit, cont.matrix)
fit.main<-eBayes(fit.main)
save(fit.main, file="celltypes-fit.main.Rda")

```

La penúltima instrucción ejecuta el proceso de regularización de la varianza utilizando modelos bayesianos empíricos para combinar la información de toda la matriz de datos y de cada gen individual y obtener estimaciones de error mejoradas.

El análisis proporciona los estadísticos de test habituales como *fold-change* estadísticos *t* moderados o *p*-valores ajustados, que se utilizan para ordenar los genes de más a menos diferencialmente expresados.

A fin de controlar el porcentaje de falsos positivos que puedan resultar del alto número de contrastes realizados simultáneamente, los *p*-valores se ajustan de forma que tengamos control sobre la tasa de falsos positivos utilizando el método de Benjamini y Hochberg ([4]).

La función `topTable` genera para cada contraste una lista de genes ordenados de más a menos diferencialmente expresados.

```
> topTab_LPS.in.AGED <- topTable (fit.main, number=nrow(fit.main), coef="LPS.in.AGED",
+ adjust="fdr")
> topTab_LPS.in.YOUNG <- topTable (fit.main, number=nrow(fit.main), coef="LPS.in.YOUNG",
+ adjust="fdr")
> topTab_AGE <- topTable (fit.main, number=nrow(fit.main) , coef="AGE", adjust="fdr")
```

Se pueden visualizar los resultados mediante un *volcano plot*, que en este caso representa en abscisas los cambios de expresión en escala logarítmica, y en ordenadas el estadístico B en vez del “menos logaritmo” del *p*-valor.

```
coefnum = 1
opt <- par(cex.lab = 0.7)
volcanoplot(fit.main, coef=coefnum, highlight=10, names=fit.main$ID,
            main=paste("Differentially expressed genes",
            colnames(cont.matrix)[coefnum], sep="\n"))
abline(v=c(-1,1))
par(opt)
```

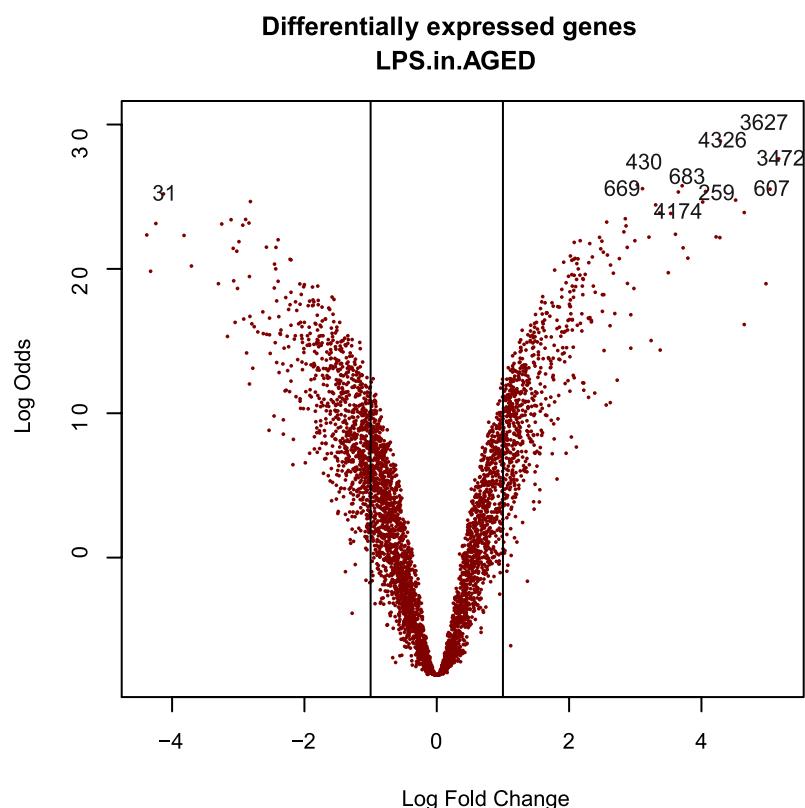


Figura 23. Volcano plot que visualiza los genes diferencialmente expresados (aquellos con *fold-change* superior a 1 o inferior a -1 y un valor de B (*log odds*) en la comparación seleccionada.)

5. Después de la selección: análisis de listas de genes

5.1. Introducción

El resultado de un análisis de microarrays como los descritos hasta el momento suele ser una lista de genes, es decir, un archivo o una tabla con los identificadores de los genes considerados, diferencialmente expresados en una o más comparaciones.

La pregunta obvia que los investigadores se plantean al disponer de estas listas es cuál es su significado biológico.

Para el bioinformático esta pregunta se replantea de la siguiente manera: ¿Qué se puede hacer para convertir —de forma más o menos automática— la lista obtenida en algún tipo de información biológicamente relevante? Esta cuestión ha llevado al desarrollo de métodos y herramientas que de forma general se agrupan bajo el título de métodos para el análisis de listas de genes o métodos de análisis de la significación biológica.

Siendo este un campo muy extenso, no lo vamos a desarrollar en profundidad sino que nos limitaremos a considerar brevemente algunas aproximaciones sencillas, que pueden contribuir a una mejor interpretación de los resultados de un experimento de microarrays. El artículo de Khatri y otros [29] realiza una revisión del estado del arte de los métodos para el análisis de la significación biológica.

Los aspectos que trataremos serán:

- Análisis comparativo de listas de genes.
- Anotación de los resultados.
- Visualización de la matriz de expresión para los genes seleccionados en una o más comparaciones.
- Análisis básico de la significación biológica: *gen enrichment analysis*.

5.2. Análisis comparativo de listas de genes

Muchos estudios analizan el comportamiento de los genes bajo varios tratamientos o condiciones experimentales.

En estos casos puede ser interesante ver cómo cambia un gen bajo distintos tratamientos o bien ver qué genes son afectados de forma parecida o distinta bajo los mismos. La forma habitual de hacer esta comparación es mediante algún programa que permita comparar los elementos de dos o más listas. En Bioconductor, por ejemplo, el paquete `limma` tiene algunas funciones que permiten seleccionar los genes cambiados simultáneamente en dos o más condiciones. El apartado 6.4.2 ilustra esta comparación con las tres listas de genes diferencialmente expresados, obtenidas en el análisis de los genes del ejemplo *estrogen*.

5.3. Anotación de los resultados

La identificación de los genes seleccionados puede resultar más sencilla para el especialista en un campo si se utilizan nombres estándar como el símbolo del gen o *gene symbol*. Ahora bien hemos de recordar que los microarrays suele ser un producto comercial en los que la empresa que los crea decide qué sondas de cada gen o exón coloca en cada posición del array. Esto significa que para poder indicar el nombre de cada gen seleccionado, es preciso disponer de algún tipo de tabla de anotaciones que relacione cada sonda con los genes a los que apunta. Esta tabla la deberían proporcionar las compañías que producen los arrays, pero es algo que queda a su criterio y, en el mejor de los casos, cada empresa puede seguir unos criterios distintos.

Afortunadamente, si se trabaja con Bioconductor existe una solución homogénea a este problema. Se trata de los paquetes de anotaciones creados y mantenidos por el equipo de Bioconductor a partir de la información publicada por cada compañía, pero con una forma común de acceso. Es decir, aunque las distintas empresas utilicen formatos distintos, desde el punto de vista del usuario de Bioconductor esto no importará: si existe el paquete se utiliza como todos los demás.

De hecho, Bioconductor incorpora varios tipos de paquetes de anotaciones:

- **Paquetes de anotaciones de microarrays:** Asocian a cada sonda la información disponible para ella en distintas bases de datos. Consisten en una serie de tablas SQL (de hecho cada paquete es una base de datos SQL), que asocian las sondas de cada gen con sus correspondencias en distintas bases de datos.
- **Paquetes de anotaciones de organismos:** Contienen asociaciones similares a las anteriores pero las claves principales de cada tabla no son las sondas de un microarray determinado sino los genes de un organismo dado. Es decir, en estos paquetes cada tabla asocia el identificador Entrez del gen con sus correspondientes identificadores en otras bases de datos.
- **Paquetes de anotaciones de bases de datos:** Estos paquetes difieren de los anteriores porque lo que hacen es almacenar en el mismo formato que

los anteriores –bases de datos SQL consultables en R– una copia de ciertas bases de datos como la Gene Ontology (GO) o la Kyoto Encyclopedia of Genes and Genomes (KEGG).

Para saber qué anotaciones están disponibles, debe cargarse el paquete y llamar la función del mismo nombre. Por ejemplo, para los datos del caso *celltypes* y saber qué anotaciones están disponibles, debe cargarse el paquete de anotaciones correspondiente (en este caso, mouse4302.db) e invocar la función que tiene el mismo nombre que el paquete, pero sin el sufijo ".db".

```
> if (!require(mouse4302.db)) {
+     biocLite("mouse4302.db")
+
> require(mouse4302.db)
> anotData <- capture.output(mouse4302())
> print(anotData[1:15])

[1] "Quality control information for mouse4302:"
[2] ""
[3] ""
[4] "This package has the following mappings:"
[5] ""
[6] "mouse4302ACNUM has 45101 mapped keys (of 45101 keys)"
[7] "mouse4302ALIAS2PROBE has 70701 mapped keys (of 128120 keys)"
[8] "mouse4302CHR has 38464 mapped keys (of 45101 keys)"
[9] "mouse4302CHRENGTHS has 35 mapped keys (of 35 keys)"
[10] "mouse4302CHRLOC has 34026 mapped keys (of 45101 keys)"
[11] "mouse4302CHRLOCEND has 34026 mapped keys (of 45101 keys)"
[12] "mouse4302ENSEMBL has 33575 mapped keys (of 45101 keys)"
[13] "mouse4302ENSEMBL2PROBE has 17074 mapped keys (of 20989 keys)"
[14] "mouse4302ENTREZID has 38535 mapped keys (of 45101 keys)"
[15] "mouse4302ENZYME has 4534 mapped keys (of 45101 keys)"

> cat ("... output continues until ", length(anotData), " lines.\n")

... output continues until 46 lines.
```

Si en vez del paquete del microarray usáramos el paquete de organismo, en este caso del ratón:

```
> if (!require(org.Mm.eg.db)) {
+     biocLite("org.Mm.eg.db")
+
> require(org.Mm.eg.db)
> anotData <- capture.output(org.Mm.eg())
> print(anotData[1:15])
```

```
[1] "Quality control information for org.Mm.eg:"
[2] ""
[3] ""
[4] "This package has the following mappings:"
[5] ""
[6] "org.Mm.egACCNUM has 36333 mapped keys (of 57955 keys)"
[7] "org.Mm.egACCNUM2EG has 472125 mapped keys (of 472125 keys)"
[8] "org.Mm.egALIAS2EG has 128120 mapped keys (of 128120 keys)"
[9] "org.Mm.egCHR has 56836 mapped keys (of 57955 keys)"
[10] "org.Mm.egCHRLLENGTHS has 35 mapped keys (of 35 keys)"
[11] "org.Mm.egCHRLLOC has 21224 mapped keys (of 57955 keys)"
[12] "org.Mm.egCHRLOCEND has 21224 mapped keys (of 57955 keys)"
[13] "org.Mm.egENSEMBL has 21575 mapped keys (of 57955 keys)"
[14] "org.Mm.egENSEMBL2EG has 20989 mapped keys (of 20989 keys)"
[15] "org.Mm.egENSEMBLPROT has 21553 mapped keys (of 57955 keys)"

> cat ("... output continues until ", length(anotData), " lines.\n")

... output continues until 55 lines.
```

Cada tabla de asociación puede consultarse de diversas formas:

- Con las funciones `get` o `mget`.
- Convirtiéndola en una tabla y extrayendo valores.
- En algunos casos utilizando funciones específicas como `getSYMBOL` o `getEG` (por Entrez Gene) cuando existan.

Las instrucciones siguientes muestran cómo recuperar los nombres de los genes (*gene symbol*) correspondientes a los 5 primeros genes seleccionados en una de las comparaciones realizadas en el análisis del ejemplo *celltypes* (efecto de la estimulación con LPS en ratones viejos) que denominamos "LPS_AGED".

Aunque en el ejemplo se utilizan los tres métodos mencionados, normalmente se aplicará tan solo uno de ellos.

```
> top5 <- topTab_LPS.in.AGED$ID[1:5]
> cat("Usando mget\n")

#Usando mget

> geneSymbol5.1 <- unlist(mget(top5, mouse4302SYMBOL))
> geneSymbol5.1

1450826_a_at 1457644_s_at 1449450_at 1419607_at 1419681_a_at
 "Saa3"      "Cxcl1"     "Ptges"      "Tnf"       "Prok2"
```

```
> cat("Usando toTable\n")

#Usando toTable

> genesTable<- toTable(mouse4302SYMBOL)
> rownames(genesTable) <- genesTable$probe_id
> genesTable[top5, 2]

[1] "Saa3" "Cxcl1" "Ptges" "Tnf" "Prok2"

> cat("Usando getSYMBOL\n")

#Usando getSYMBOL

> require(annotate)
> geneSymbol5.3 <- getSYMBOL(top5, "mouse4302.db")
> geneSymbol5.3

1450826_a_at 1457644_s_at 1449450_at 1419607_at 1419681_a_at
"Saa3"      "Cxcl1"     "Ptges"      "Tnf"      "Prok2"
```

5.4. Visualización de los perfiles de expresión

Tras seleccionar los genes diferencialmente expresados, podemos visualizar las expresiones de cada gen agrupándolas para destacar los genes que se encuentran *up* o *down* regulados simultáneamente, constituyendo perfiles de expresión.

Hay distintas formas de visualización pero aquí tan solo se presenta el uso de mapas de color o *Heatmaps*. La idea de los mapas de color es convertir los valores numéricos de una matriz de datos en tonos de color, de manera que sea posible obtener una visión de conjunto de cómo varían los distintos valores, por ejemplo, entre muestras o entre genes. Para ello, además de convertir los datos en colores según algún criterio (por ejemplo, los valores bajos en verde y los altos en rojo, con una transición gradual entre ambos) se suele realizar una agrupación jerárquica de las columnas (muestras), las filas (genes), o ambos a la vez, con lo cual, los valores parecidos tenderán a quedar agrupados, lo que ayudará a visualizar la variación conjunta.

En el apartado 6.4.4 se muestra un *heatmap* realizado con los genes diferencialmente expresados del ejemplo *estrogen*.

5.5. Análisis de significación biológica

En los subapartados anteriores se ha visto cómo encontrar los identificadores de los genes en distintas bases de datos, lo que permite, por ejemplo, conocer sus nombres comunes (*gene symbol*).

Otra aproximación razonable es estudiar las funciones de los genes buscando sus anotaciones en bases de datos de anotación funcional como la *Gene Ontology* (GO), o la *Kyoto Encyclopedia of Genes and Genomes* (KEGG).

El código siguiente muestra cómo obtener las anotaciones de la *gene ontology* para el primer gen seleccionado en una de las comparaciones realizadas en el ejemplo *celltypes* (efecto de la estimulación con LPS en ratones viejos) que denominamos "LPS_AGED".

```
> require(annotate)
> top1 <- topTab_LPS.in.AGED$ID[1:1]
> geneSymbol1 <- getSYMBOL(top1, "mouse4302.db")
> GOAnots1 <- mget(top1, mouse4302GO)
> for (i in 1:length(GOAnots1)){
+   for (j in 1:length(GOAnots1[[i]])){
+     GOAnot <- GOAnots1[[i]][[j]][[1]]
+     cat(top1[i],geneSymbol1[i],GOAnot,substr(Term(GOAnot),1,30), "\n")
+   }
+ }
```

1450826_a_at Saa3 GO:0006953 acute-phase response
 1450826_a_at Saa3 GO:0005576 extracellular region
 1450826_a_at Saa3 GO:0034364 high-density lipoprotein parti

Como se ve en el ejemplo, el número de anotaciones para un gen en la Gene Ontology es muy alto, aparte de que, aunque aquí no se muestra, no todas las anotaciones tienen la misma fiabilidad.

A parte del problema de lo extenso de las anotaciones, está el hecho de que antes de empezar a hacer inferencias sobre el significado de una anotación debería poderse establecer si dicha anotación está relacionada con el proceso que se está estudiando o aparece por azar entre la muchas anotaciones de los genes de la lista. Hay diferentes métodos y modelos para hacer esto (ver Draghici y otros, [30] o Mosquera y Sánchez-Pla, [36], [39]) pero aquí se presentará brevemente lo que se conoce por análisis de enriquecimiento.

5.5.1. Análisis de enriquecimiento

El objetivo del análisis de enriquecimiento es establecer si una determinada categoría, que representa, por ejemplo, un proceso biológico (GO) o una vía metabólica o *pathway* (KEGG), aparece con mayor o menor frecuencia en la

lista de genes seleccionados que en la población de genes, desde donde se han obtenido, es decir, el array, el genoma, o simplemente los genes que fueron seleccionados para la prueba. La idea básica es que si una categoría de la GO (es decir, una cierta función molecular, un proceso biológico dado o incluso cierta componente celular) aparece más a menudo en la lista que en general, es probable que tenga algo que ver con el proceso en base al cual se ha seleccionado la lista que se estudia.

Por ejemplo, consideremos un experimento que da una lista de genes y el 10% de los genes más diferencialmente expresados están asociados con el término *apoptosis* en la GO (GO:0006915). Esto puede parecer una proporción inusualmente grande de la lista de genes, dado que la apoptosis es un proceso biológico muy específico. Para determinar cuánto más grande de lo normal es esta proporción, debe compararse con la proporción de genes relacionados con apoptosis en la lista de genes de referencia, por ejemplo el conjunto de todos los genes del microarray, más a menudo, los genes que se han incluido en el análisis tras descartar los que pueden considerarse ruido de fondo.

El análisis estadístico realizado para comparar proporciones es un test hipergeométrico o el test exacto de Fisher que se utiliza para probar la hipótesis:

$$H_0: p_{sel}^A = p_{all}^A \quad vs \quad H_1: p_{sel}^A \neq p_{all}^A , \quad (18)$$

donde A representa el conjunto de genes cuya más/menos representación está siendo considerada, p_{sel}^A es la proporción de genes seleccionados que están incluidos en este conjunto de genes y p_{all}^A es la proporción de genes de la lista de referencia.

5.5.2. Análisis de enriquecimiento con Bioconductor

Hay muchas herramientas que pueden ayudar a realizar este análisis. Siguiendo nuestra costumbre usaremos las que contiene el paquete de R (Bioconductor), principalmente en el paquete G0stats.

El análisis realizado utilizando este paquete procede de la forma siguiente:

- Toma como entrada los identificadores de *Entrez* de la lista de genes seleccionada, así como el nombre del paquete de la anotación correspondiente al array que ha sido usado para el análisis.
- La salida del análisis es la lista de categorías que aparece más/menos representado en cada conjunto seleccionado.

El código siguiente ilustra cómo se procederá para realizar un análisis de enriquecimiento con las listas de genes seleccionados en la primera de las comparaciones realizadas en este apartado.

```
> require(GOstats)
> require(mouse4302.db)
> if(!(require(org.Mm.eg.db)))
+ biocLite("org.Mm.eg.db")
> require(org.Mm.eg.db)
> # Seleccionamos la "topTable"
> topTab <- topTab_LPS.in.AGED
> # Definimos el universo de genes: todos los que se han incluido en el análisis
> # El programa trabaja con identificadores "entrez" y no admite duplicados
>
> entrezUniverse = unique(getEG(as.character(topTab$ID), "mouse4302.db"))
> # Escogemos los grupos de sondas a incluir en el análisis
> # Este análisis trabaja bien con varios centenares de genes
> # por lo que es habitual basarse en p-valores sin ajustar para incluirlos
>
> whichGenes<-topTab["adj.P.Val"]<0.001
> geneIds <- unique(getEG(as.character(topTab$ID[whichGenes]), "mouse4302.db"))
> # Creamos los "hiperparámetros" en que se basa el análisis
> GOparams = new("GOHyperGParams",
+ geneIds=geneIds, universeGeneIds=entrezUniverse,
+ annotation="org.Mm.eg.db", ontology="BP",
+ pvalueCutoff=0.001, conditional=FALSE,
+ testDirection="over")
> KEGGparams = new("KEGGHyperGParams",
+ geneIds=geneIds, universeGeneIds=entrezUniverse,
+ annotation="org.Mm.eg.db",
+ pvalueCutoff=0.01, testDirection="over")
> # Ejecutamos los análisis
>
> GOhyper = hyperGTest(GOparams)
> KEGGhyper = hyperGTest(KEGGparams)
> # Creamos un informe html con los resultados
> comparison = "topTab_LPS.in.AGED"
> GOfilename =file.path(resultsDir,
+ paste("GOResults.",comparison,".html", sep=""))
> KEGGfilename =file.path(resultsDir,
+ paste("KEGGResults.",comparison,".html", sep=""))
> htmlReport(GOhyper, file = GOfilename, summary.args=list("htmlLinks"=TRUE))
> htmlReport(KEGGhyper, file = KEGGfilename, summary.args=list("htmlLinks"=TRUE))
```

6. Caso resuelto: Selección de genes diferencialmente expresados

6.1. Introducción

El análisis de datos de microarrays suele proceder, en general secuencialmente, a través de una serie de etapas tal y como se puede ver en la figura 1.

En cada una de las etapas pueden llevarse a cabo diversos procesos, con diversas variantes de métodos parecidos. A lo largo de la primera década del 2000 se ha desarrollado el proyecto Bioconductor, que ha impulsado la creación de cientos de paquetes de R que implementan casi todas las capacidades posibles de análisis de microarrays. Aunque nadie puede conocer todos los paquetes, con tan solo conocer algunos de ellos es posible llevar a cabo potentes análisis con relativa facilidad.

El objetivo de este apartado es ilustrar de forma breve pero completa cómo se hace un análisis de microarrays usando las facilidades que ofrecen Bioconductor y R.

Como se ha dicho, uno de los problemas en este tipo de estudios es que en cada etapa se puede proceder de varias formas, lo que da lugar a un asfixiante número de posibilidades especialmente para el neófito. Con el fin de evitar este problema, de momento se describe una sola de las opciones posibles para cada paso, lo que constituye un proceso “al estilo Bioconductor”.

6.1.1. El ejemplo

El ejemplo analizado consiste en unos datos disponibles en forma de paquete (*estrogen*) en Bioconductor. Se trata de un estudio en el que se analiza la influencia del tratamiento con estrógeno y del tiempo transcurrido desde el tratamiento en la expresión de genes asociados con cáncer de mama y que ha sido descrito en el apartado 1.3.3.

El diseño experimental para este estudio consiste en un diseño factorial de 2 factores: *estrogen* que puede tomar los valores 'presente' o 'ausente' según si se ha suministrado o no tratamiento con estrógenos, y tiempo de exposición que toma los valores '10h' (10 horas) o '48h' (48 horas) según haya durado la exposición al estrógeno. La documentación del paquete *estrogen* ofrece más detalles acerca del diseño y las variables utilizadas. Si el paquete no se encuentra instalado, puede hacerse con la instrucción `biocLite`, que se descarga de la web de Bioconductor.

```
> if (!require("estrogen", character.only=T)){  
+   source("http://bioconductor.org/biocLite.R")  
+   biocLite("estrogen")  
+ }
```

En lo que sigue supondremos que se ha llevado a cabo una instalación estándar de Bioconductor, es decir, se han ejecutado las instrucciones:

```
> source("http://bioconductor.org/biocLite.R")  
> biocLite("estrogen")
```

6.1.2. Directorios y opciones de trabajo

Para facilitar supondremos que trabajamos en un directorio escogido por nosotros y cuya localización se asigna a la variable `workingDir`. Los datos se copiarán en un subdirectorio del anterior denominado `data`, que se almacenará en la variable `dataDir` y los resultados se almacenarán en un directorio "results", cuyo nombre completo se almacenará en la variable `resultsDir`.

```
> workingDir <- getwd()  
> system("mkdir data")  
> system("mkdir results")  
> dataDir <- file.path(workingDir, "data")  
> resultsDir <- file.path(workingDir, "results")  
> setwd(workingDir)  
  
> options(width=80)  
> options(digits=5)
```

6.2. Obtención y lectura de los datos

Para un análisis de datos de microarrays de Affymetrix se necesitan los archivos de imágenes escaneadas (.CEL) y un archivo en el que se asigne una condición experimental a cada archivo.

6.2.1. Los datos

Una ventaja de este ejemplo es que los datos se encuentran disponibles tras instalar el paquete `estrogen`, por lo que se pueden copiar en un directorio de trabajo o simplemente trabajar con ellos en el directorio original. El directorio en que se encuentran los archivos .CEL es:

```
> require(estrogen)  
> estrogenDir <- system.file("extdata", package = "estrogen")  
> print(estrogenDir)
```

```
[1] "/home/alex/R/x86_64-pc-linux-gnu-library/2.15/estrogen/extdata"
```

Para realizar el análisis, es preciso copiar todos los archivos del directorio “ext-data” a nuestro directorio de trabajo “data”.

Por ejemplo, para copiarlos desde R, en el sistema operativo linux, se escribirá:

```
> system(paste ("cp ", estrogenDir,"/* ./data", sep=""))
```

Los archivos copiados son:

- Los ocho archivos .CEL para el análisis.
- Un archivo defectuoso para que se vea cómo debe salir un “array malo” en el control de calidad.
- Un archivo de covariables o *targets*, que se usará al leer los archivos .CEL para incorporar la información de las covariables en el objeto que se creará.

6.2.2. Lectura de los datos

El proceso de leer los datos puede parecer un poco extraño a primera vista, pero la idea es simple:

- En primer lugar creamos algunas estructuras de datos que contienen la información sobre las variables y –opcionalmente– sobre las anotaciones y el experimento.
- A continuación nos basamos en estas estructuras para leer los datos y crear los objetos principales que se utilizarán para el análisis.

```
> require(BioBase)
> require(affy)
> sampleInfo <- read.AnnotatedDataFrame(file.path(estrogenDir,"targLimma.txt"),
+ header = TRUE, row.names = 1, sep="\t")
> fileNames <- pData(sampleInfo)$FileName
> rawData <- read.affybatch(filenames=file.path(estrogenDir,fileNames),
+ phenoData=sampleInfo)
```

Una forma alternativa de leer los datos consiste en acceder al directorio donde se encuentran y escribir `ReadAffy`. En este ejemplo se incluye un archivo defectuoso con fines didácticos, llamado *badcel*. Para ilustrar las diferencias entre archivos correctos e incorrectos, se puede leer en otro objeto.

```
> require(affy)
> setwd(estrogenDir)
```

```
> rawData.wrong <- ReadAffy()  
> setwd(workingDir)
```

6.3. Exploración, control de calidad y normalización

Tras leer los datos pasamos al preprocesado. Aunque puede interpretarse de distintas formas, esta fase suele consistir en:

- 1) Realizar algunos gráficos con los datos para hacerse una idea de cómo ha resultado el experimento.
- 2) Realizar un control de calidad más formal.
- 3) Normalizar y, en el caso de Affymetrix, resumir (sumarizar) las expresiones.

6.3.1. Exploración y visualización

La exploración previa puede hacerse paso a paso o en bloque si se utiliza algún paquete como `arrayQualityMetrics`. En la ayuda de este paquete se encuentra una descripción de los análisis básicos que permite realizar.

El primer gráfico que suele hacerse es algún tipo de gráfico de histograma, que, de hecho, suele denominarse gráfico de densidad, que muestre la distribución de las señales en los datos originales, es decir, sin normalizar. Estos gráficos también permiten hacerse una idea de si las distribuciones de los distintos arrays son similares en forma y posición.

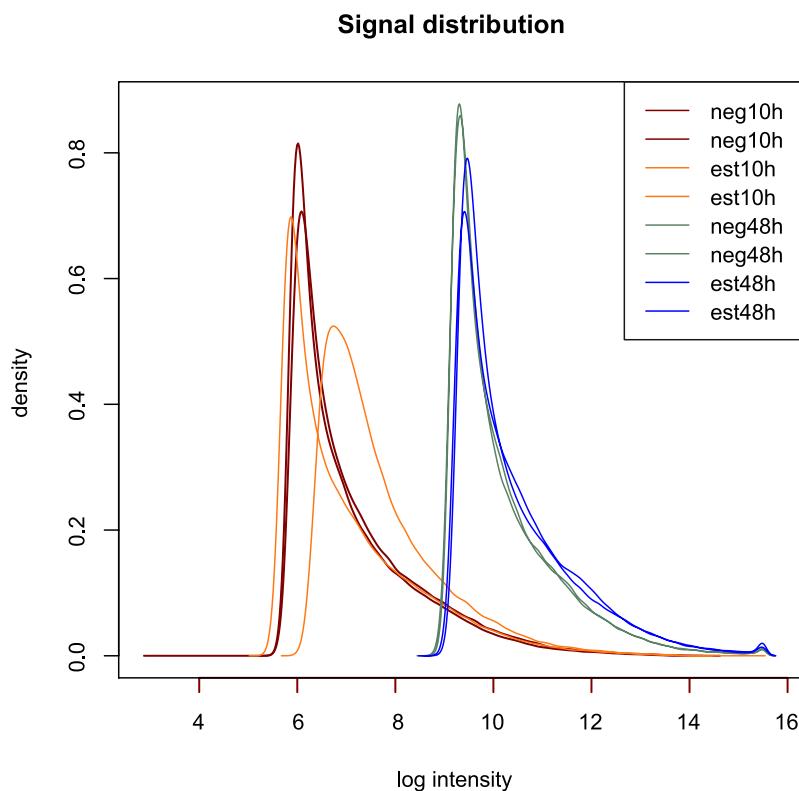


Figura 24. Gráficos de densidad para los 8 microarrays del estudio *estrogen*. Aunque su forma no sugiere que haya algún array problemático, los gráficos correspondientes a 10 y 48 horas son claramente distintos los que sugiere la necesidad de normalizar los datos.

El diagrama de cajas proporciona, como el histograma, una idea de la distribución de los datos.

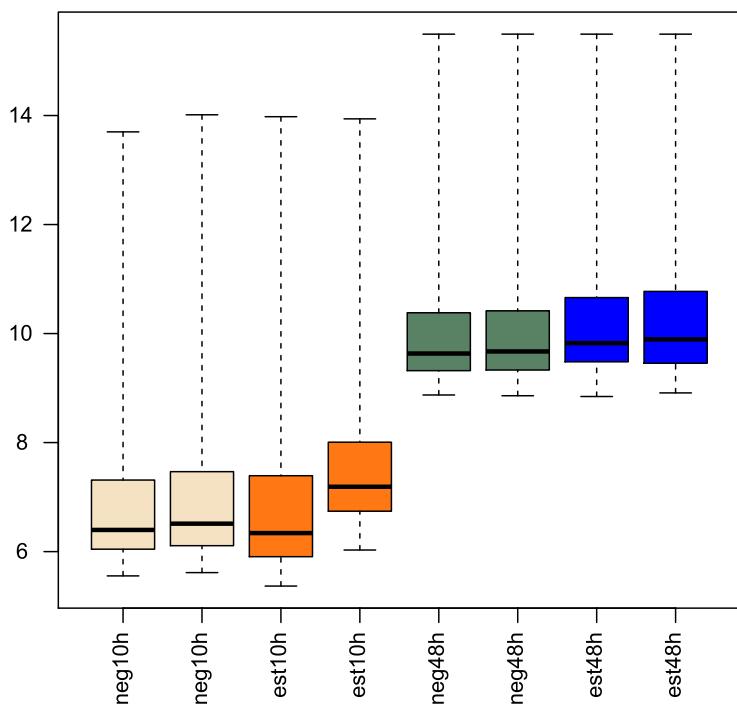


Figura 25. Boxplots correspondientes a las 8 muestras del ejemplo *estrogen*. Como pasaba con los gráficos de densidad se puede ver dos grupos bien definidos de muestras.

Finalmente un clúster jerárquico seguido de un dendrograma nos puede ayudar a hacernos una idea de si las muestras se agrupan por condiciones experimentales.

Si lo hacen es bueno, pero si no, no es necesariamente indicador de problemas, puesto que es un gráfico basado en todos los datos.

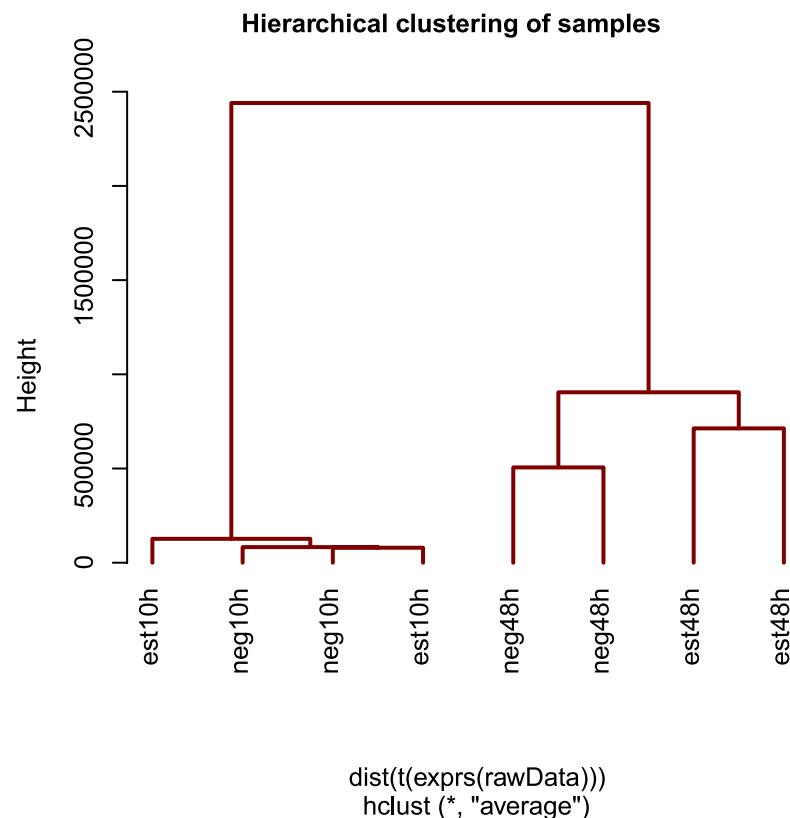


Figura 26. Dendrograma resultante de realizar un clúster jerárquico de las muestras, basado en la información de todos los genes. Como en los otros gráficos, se muestra cómo el principal factor que determina diferencias entre muestras es el tiempo de exposición.

6.3.2. Control de calidad

Las exploraciones anteriores nos proporcionan una idea de cómo son los datos.

Se pueden realizar controles de calidad más estrictos como:

- Los controles de calidad estándar de Affymetrix, descritos en el paquete `simpleaffy`.
- Controles basados en modelos a nivel de sondas, descritos en el paquete `affyPLM`.

El paquete `affyQCReport` encapsula los análisis que pueden realizarse con el paquete `simpleaffy`, de forma que con una instrucción se pueden realizar todos los análisis y enviar la salida a un archivo.

```
> stopifnot(require(affyQCReport))
```

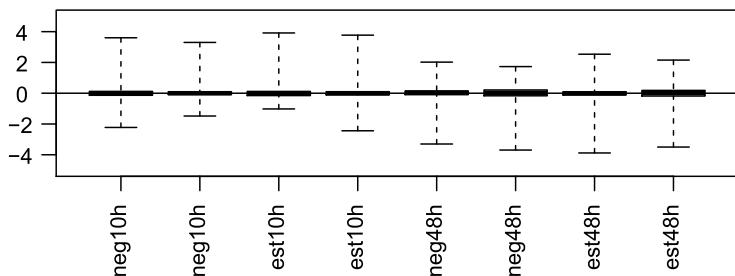
```
> QCReport(rawData, file=file.path(resultsDir, "QCReport.pdf"))
```

El paquete `affyPLM` realiza un control de calidad basado en lo que se conoce como modelos a nivel de sonda o *probe-level models*, conocidos por sus siglas (PLM).

```
> stopifnot(require(affyPLM))
> Pset<- fitPLM(rawData)
```

Como resultado del ajuste PLM se pueden obtener dos gráficos, uno de expresiones relativas y otro con errores estandarizados (figura 27). Si los datos son de calidad, ambos gráficos deben ser centrados y relativamente simétricos. Cambios en esta situación sugieren problemas en los arrays que no los verifiquen.

Relative Log Expression



Normalized Unscaled Standard Errors

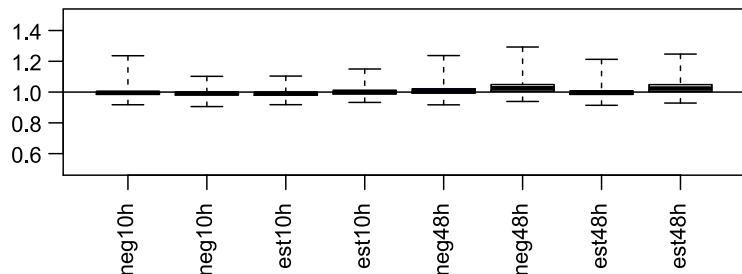


Figura 27. Gráficos RLE y NUOE resultantes de aplicar un análisis de bajo nivel con cada uno de los arrays. La regularidad de las cajas sugiere que no hay ningún problema en los datos.

Para concluir con este subapartado, merece la pena tener en cuenta que todos estos gráficos son exploratorios. Raramente se descarta un array si solo un grafico lo sugiere.

Merece la pena repetir la exploración y el control de calidad con el objeto `rawData.wrong`. En este caso hay un array defectuoso y se ve muy claramente que quiere decir array problemático en los gráficos.

6.3.3. Normalización y filtraje

Una vez realizado el control de calidad se procede a normalizar los datos y summarizarlos.

Hecho esto puede realizarse un filtraje no específico con el fin de eliminar genes que constituyen básicamente ruido, bien porque sus señales son muy bajas o bien porque apenas varían entre condiciones, por lo que no aportan nada a la selección de genes diferencialmente expresados.

Normalización

La normalización puede hacerse por distintos métodos (MAS5, VSN, RMA, GCRMA, etc.). En este ejemplo se utilizará el método RMA pero no se realizará filtraje alguno. Esto puede implicar quizás que para seleccionar genes diferencialmente expresados basándose en el ajuste de *p*-valores debamos utilizar criterios menos restrictivos que si hubiéramos filtrado, pero tiene la ventaja de eliminar un paso que, en el mejor de los casos, resulta controvertido.

El procesado mediante RMA implica un proceso en tres etapas:

- 1) Corrección de fondo (el RMA hace precisamente esto).
- 2) Normalización para hacer los valores de los arrays comparables.
- 3) Resumen (sumarización) de las diversas sondas asociadas a cada grupo de sondas para dar un único valor.

```
> stopifnot(require(affy))
> normalize <- T
> if(normalize) {
+   eset_rma <- rma(rawData)
+   save(eset_rma, file=file.path(dataDir,"estrogen-normalized.Rda"))
+ } else{
+   load (file=file.path(dataDir,"estrogen-normalized.Rda"))
+ }

Background correcting
Normalizing
Calculating Expression
```

La normalización hace que los valores de los arrays sean comparables entre ellos, aunque los distintos métodos sitúan los valores en escalas distintas, por lo que lo que no resulta directamente comparable son los valores normalizados por distintos métodos.

Un boxplot de los valores normalizados sugiere que los valores ya están en una escala en donde se pueden comparar.

```
> boxplot(eset_rma, main="Boxplot for RMA-normalized expression values ",  
+           names=sampleNames, cex.axis=0.7, col=info$grupo+1, las=2)
```

Boxplot for RMA–normalized expression values

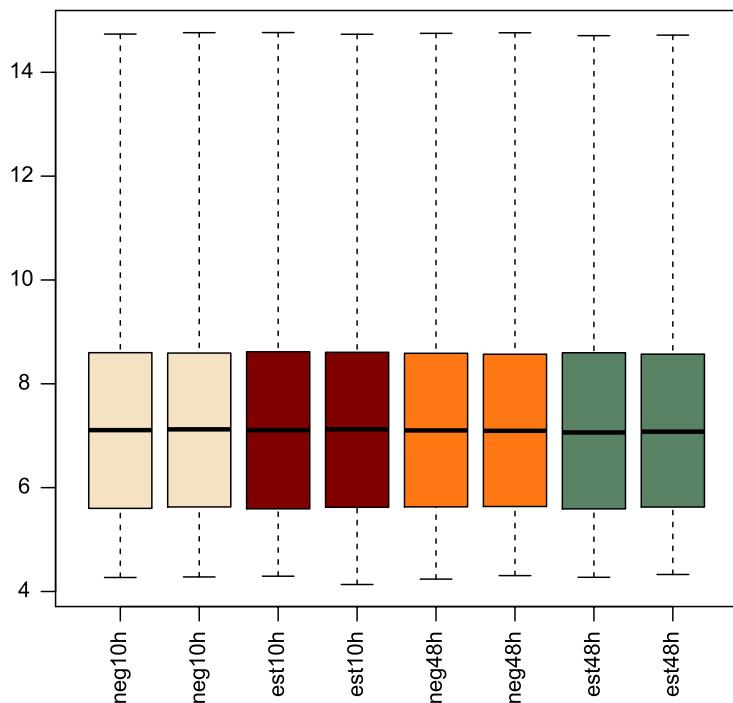


Figura 28. El diagrama de caja de los valores normalizados y sumarizados muestra cómo la distribución de todos los arrays coincide exactamente. Esto, sin embargo, no es un indicio de buena calidad, sino que es una consecuencia de aplicar el método de normalización por cuantiles que fuerza precisamente esta homogeneidad.

El código siguiente, que no se ejecuta, muestra cómo se podrían aplicar distintos métodos para luego compararlos entre ellos.

```
> eset_mas5 <- mas5(rawData) # Uses expresso (MAS 5.0 method) much slower than RMA!  
> stopifnot(require(gcrma))  
> eset_gcrma <- gcrma(rawData) # The 'library(gcrma)' needs to be loaded first.  
> stopifnot(require(plier))  
> eset_plier <- justPlier(rawData, normalize=T) # The 'library(plier)' needs to be loaded first.  
> compara <- data.frame(RMA=exprs(eset_rma)[,1], MAS5 =exprs(eset_mas5)[,1],  
+                         GCRMA=exprs(eset_gcrma)[,1], PLIER =exprs(eset_plier)[,1])  
> pairs(compara)
```

Filtraje

El filtraje no específico permite eliminar los genes que varían poco entre condiciones o que deseamos quitar por otras razones, como por ejemplo que no disponemos de anotación para ellos. La función `nsFilter` permite eliminar los genes que, o bien varían poco, o bien no se dispone de anotación para ellos.

Si al filtrar deseamos usar las anotaciones, o la falta de ellas, como criterio de filtraje, debemos disponer del correspondiente paquete de anotaciones.

```
> require(genefilter)
> filtered <- nsFilter(eset_rma, require.entrez=TRUE,
+                         remove.dupEntrez=TRUE, var.func=IQR,
+                         var.cutoff=0.5, var.filter=TRUE,
+                         filterByQuantile=TRUE, feature.exclude="^AFFX")
```

La función `nsFilter` devuelve los valores filtrados en un objeto `expressionSet` y un informe de los resultados del filtraje.

```
> names(filtered)

[1] "eset" "filter.log"

> class(filtered$eset)

[1] "ExpressionSet"
attr(,"package")
[1] "Biobase"

> print(filtered$filter.log)

$numDupsRemoved
[1] 2950

$numLowVar
[1] 4402

$numRemoved.ENTREZID
[1] 853

$feature.exclude
[1] 19

> eset_filtered <- filtered$eset
```

Podemos grabar el objeto `eset_rma` y los datos filtrados para su posterior uso.

```
> save(eset_rma, eset_filtered, file=file.path(resultsDir, "estrogen-normalized.Rda"))
```

Después del filtraje han quedado 4409 genes disponibles para analizar.

6.4. Selección de genes diferencialmente expresados

Como en las etapas anteriores la selección de genes diferencialmente expresados (GDE) puede basarse en distintas aproximaciones, desde la *t* de Student al programa `limma` pasando por multitud de variantes.

En este ejemplo se aplicará la aproximación presentada por Smyth [37] basada en la utilización del modelo lineal general, combinada con un método para obtener una estimación mejorada de la varianza.

6.4.1. Análisis basado en modelos lineales

El apartado 4.2. de este manual o el manual del programa `limma` contienen explicaciones detalladas sobre construir un modelo lineal para este problema y cómo utilizarlo para seleccionar genes diferencialmente expresados.

Matriz de diseño

El primer paso para el análisis es crear la matriz de diseño. La situación discutida en este ejemplo se puede modelizar de dos formas:

- Como un modelo de dos factores: *estrógeno* (Pres./Aus.) y tiempo de exposición (10h/48h) con interacción.
- Como un modelo de un factor con cuatro niveles:
 - *est10h* (presencia de estrógenos y tiempo de exposición 10 h),
 - *neg48h* (presencia de estrógenos y tiempo de exposición 48 h),
 - *est10h* (presencia de estrógenos y tiempo de exposición 10 h),
 - *neg48h* (ausencia de estrógenos y tiempo de exposición 48 h).

Tal como se describe en el manual de `limma`, la segunda parametrización resulta a menudo más cómoda, a pesar de parecer menos intuitiva, porque permite formular con más facilidad que la de dos factores las preguntas que típicamente interesan a los investigadores.

El modelo lineal para este estudio será:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{Design Matrix, } \mathbf{X}} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_7 \\ \varepsilon_8 \end{pmatrix}$$

Los parámetros del modelo representan las cuatro combinaciones tiempo/estrógeno.

$$\begin{aligned}\alpha_1 &= \mathbf{E}(\log Abs.10h), \\ \alpha_2 &= \mathbf{E}(\log Pres.10h), \\ \alpha_3 &= \mathbf{E}(\log Abs.48h), \\ \alpha_4 &= \mathbf{E}(\log Pres.48h).\end{aligned}$$

La matriz de diseño puede definirse manualmente o a partir de un factor creado específicamente para ello. Manualmente, sería:

```
> design.1<-matrix(
+   c(1,1,0,0,0,0,0,0,
+     0,0,1,1,0,0,0,0,
+     0,0,0,0,1,1,0,0,
+     0,0,0,0,0,0,1,1),
+   nrow=8,
+   byrow=F)
> colnames(design.1)<-c("neg10h", "est10h", "neg48h", "est48h")
> rownames(design.1) <- c("low10A", "low10B", "hi10A" , "hi10B", "low48A", "low48B", "hi48A" ,
+ "hi48B")
> print(design.1)
      neg10h est10h neg48h est48h
low10A    1      0      0      0
low10B    1      0      0      0
hi10A     0      1      0      0
hi10B     0      1      0      0
low48A    0      0      1      0
low48B    0      0      1      0
hi48A     0      0      0      1
hi48B     0      0      0      1
```

Contrastes

Dado un modelo lineal definido a través de una matriz de diseño, pueden formularse las preguntas de interés como contrastes, es decir, comparaciones entre los parámetros del modelo.

Cada parametrización distinta requerirá de unos contrastes diferentes para las mismas preguntas, por lo que habitualmente se utilizará la parametrización que permita formular de forma más clara las comparaciones de interés.

En este caso interesa estudiar

- Efecto del estrógeno al inicio del tratamiento
- Efecto del estrógeno al cabo de un tiempo del tratamiento

- Efecto del tiempo en ausencia de estrógeno

Esto se puede formular fácilmente con la parametrización adoptada.

$$\begin{aligned}\beta_1^1 &= \alpha_2 - \alpha_1, && \text{Efecto del estrógeno pasadas 10 horas} \\ \beta_2^1 &= \alpha_4 - \alpha_3, && \text{Efecto del estrógeno pasadas 48 horas} \\ \beta_3^1 &= \alpha_3 - \alpha_1, && \text{Efecto del tiempo en ausencia de Estrógeno}\end{aligned}$$

```
> cont.matrix <- makeContrasts (
+ Estro10=(est10h-neg10h),
+ Estro48=(est48h-neg48h),
+ Tiempo=(neg48h-neg10h),
+ levels=design)
> cont.matrix

Contrasts
Levels   Estro10  Estro48  Tiempo
neg10h     -1       0      -1
est10h      1       0       0
neg48h      0      -1       1
est48h      0       1       0
```

Estimación del modelo y selección de genes

Una vez definida la matriz de diseño y los contrastes, podemos pasar a estimar el modelo, estimar los contrastes y realizar las pruebas de significación que nos indiquen, para cada gen y cada comparación, si puede considerarse diferencialmente expresado.

El método implementado en `limma` amplía el análisis tradicional utilizando modelos de Bayes empíricos para combinar la información de toda la matriz de datos y de cada gen individual y obtener estimaciones de error mejoradas.

El análisis proporciona los estadísticos de test habituales como *Fold-change t*-moderados o *p*-valores ajustados, que se utilizan para ordenar los genes de más a menos diferencialmente expresados.

A fin de controlar el porcentaje de falsos positivos que puedan resultar del alto número de contrastes realizados simultáneamente, los *p*-valores se ajustan de forma que tengamos control sobre la tasa de falsos positivos utilizando el método de Benjamini y Hochberg ([4]).

La función `topTable` genera para cada contraste una lista de genes ordenados de más a menos diferencialmente expresados.

```
> topTabEstro10 <- topTable (fit.main, number=nrow(fit.main), coef="Estro10", adjust="fdr")
```

```
> topTabEstro48 <- topTable (fit.main, number=nrow(fit.main), coef="Estro48", adjust="fdr")
> topTabTiempo <- topTable (fit.main, number=nrow(fit.main) , coef="Tiempo", adjust="fdr")
```

Una forma de visualizar los resultados es mediante un **volcano plot**, que representa en abscisas los cambios de expresión en escala logarítmica, y en ordenadas el “menos logaritmo” del *p*-valor o alternativamente el estadístico *B*.

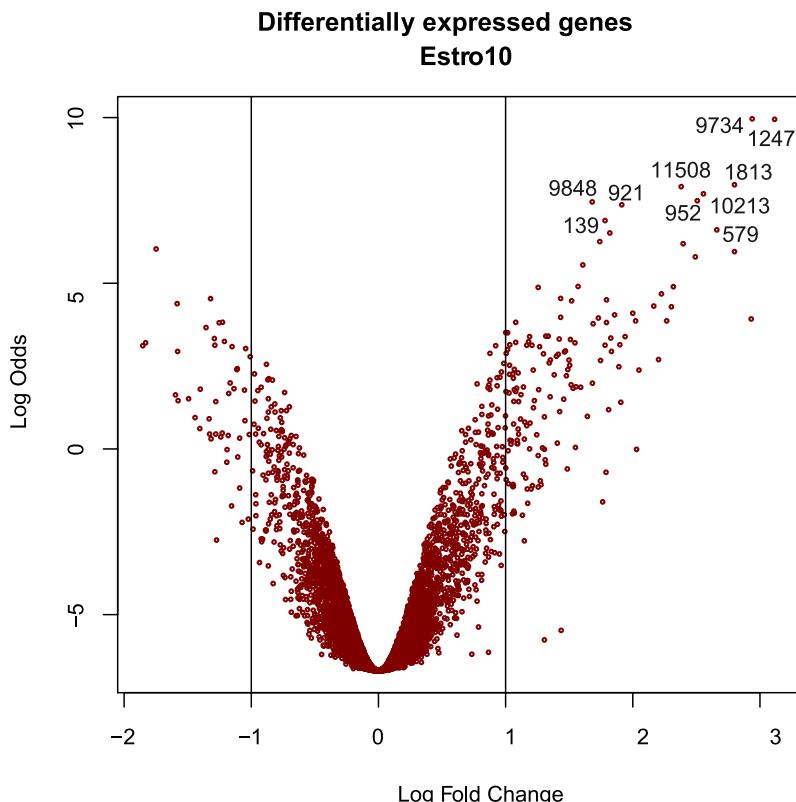


Figura 29. *Volcano plot* con los genes de la comparación sobre el efecto del estrógeno después de una exposición breve (10 h). Los genes cuyo *log odds* es superior a 0 y cuyo *log fold change* es, en valor absoluto, superior a 1, son candidatos a estar diferencialmente expresados.

6.4.2. Comparaciones múltiples

Cuando se realizan varias comparaciones a la vez puede resultar importante ver qué genes cambian simultáneamente en más de una comparación. Si el número de comparaciones es alto, también puede ser necesario realizar un ajuste de *p*-valores entre las comparaciones, distinto del realizado entre genes.

La función `decidetests` permite realizar ambas cosas. En este ejemplo no se ajustarán los *p*-valores entre comparaciones. Tan solo se seleccionarán los genes que cambian en una o más condiciones.

El resultado del análisis es una tabla, que llamaremos `res` y que para cada gen y cada comparación contiene un 1 (si el gen está sobreexpresado o *up* en esta condición), un 0 (si no hay cambio significativo) o un -1 (si está *down* regulado).

Para resumir dicho análisis podemos contar qué filas tienen como mínimo una celda distinta de cero:

```
> sum.res.rows<-apply(abs(res),1,sum)
> res.selected<-res[sum.res.rows!=0,]
> print(summary(res))
> vennDiagram (res.selected[,1:3], main="Genes in common #1", cex=0.9)
  Estro10 Estro48 Tiempo
-1      23      99     27
0      12512    12375  12591
1       90      151      7
```

Un diagrama de Venn permite visualizar la tabla anterior sin diferenciar entre genes *up* o *down* regulados.

Genes in common #1

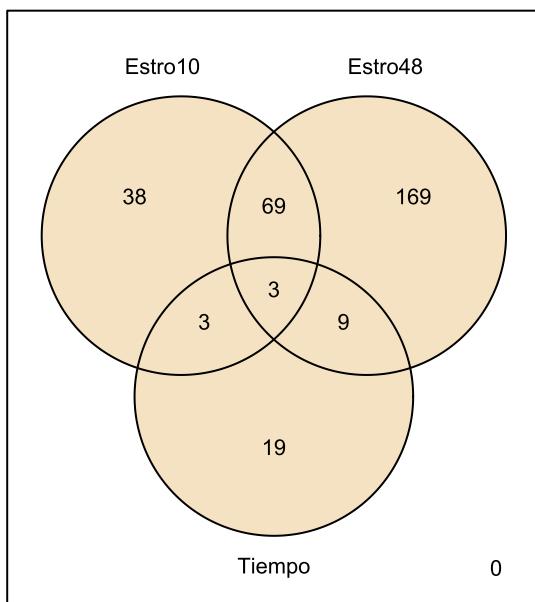


Figura 30. El diagrama de Venn muestra el número de genes seleccionados en cada comparación y cuantos hay en común entre las comparaciones dos a do o con las tres.

6.4.3. Anotación de resultados

La identificación de los genes seleccionados puede resultar más sencilla para el especialista en un campo si se utilizan nombres estándar como el símbolo del gen o *genesymbol*. Con esta finalidad, cada paquete de anotaciones tiene tablas de correspondencia entre los distintos tipos de identificadores, principalmente entre los del array y los de otras bases de datos.

Para saber qué anotaciones están disponibles, debe cargarse el paquete y llamar la función del mismo nombre.

```
> require(hgu95av2.db)
```

```
> hgu95av2()
```

Bioconductor dispone de algunos paquetes que permiten aprovechar esta funcionalidad anterior para obtener las anotaciones de cada gen y generar una tabla HTML con enlaces a algunas bases de datos.

De forma sencilla es posible obtener tablas con las anotaciones correspondientes a los genes seleccionados. Si se desea ser más ambicioso, es posible generar tablas en las que se combinen hiperenlaces a las anotaciones con los resultados de la selección de genes.

Tablas de anotaciones sencillas

El paquete `annaffy` permite de forma muy simple generar una tabla de anotaciones con hiperenlaces a las bases de datos para cada anotación seleccionada.

La instrucción siguiente crea una tabla con las anotaciones disponibles para los genes seleccionados en la sección de comparaciones múltiples.

```
> require(annaffy)
> genesSelected <- rownames(res.selected)
> at <- aaffTableAnn(genesSelected, "hgu95av2.db")
> saveHTML (at, file.path(resultsDir, "annotations.html"),
+           "Annotations for selected genes")
```

6.4.4. Visualización de los perfiles de expresión

Tras seleccionar los genes diferencialmente expresados, podemos visualizar las expresiones de cada gen agrupándolas para destacar los genes que se encuentran *up* o *down* regulados simultáneamente constituyendo perfiles de expresión.

Hay distintas formas de visualización, pero aquí tan solo se presenta el uso de mapas de color o `Heatmaps`.

En primer lugar seleccionamos los genes a visualizar: se toman todos aquellos que han resultado diferencialmente expresados en alguna de las tres comparaciones.

```
> probeNames<-rownames(res)
> probeNames.selected<-probeNames[sum.res.rows!=0]
> exprs2cluster <-exprs(eset_rma) [probeNames.selected, ]
```

Para representar el `Heatmap` solo necesitamos la matriz de datos resultante.

```
> color.map <- function(horas) { if (horas< 20) "yellow" else "red" }
> grupColors <- unlist(lapply(pData(eset_rma)$time.h, color.map))
```

```
> heatmap(exprs2cluster, col=rainbow(100), ColSideColors=grupColors, cexCol=0.9)
```

Si se desea realizar mapas de color más sofisticados, puede utilizarse el paquete `gplots` que implementa una versión mejorada en la función `heatmap.2`

```
> color.map <- function(horas) { if (horas< 20) "yellow" else "red" }
> grupColors <- unlist(lapply(pData(eset_rma)$time.h, color.map))
> require("gplots")
> heatmap.2(exprs2cluster,
+             col=bluered(75), scale="row",
+             ColSideColors=grupColors, key=TRUE, symkey=FALSE,
+             density.info="none", trace="none", cexCol=1)
```

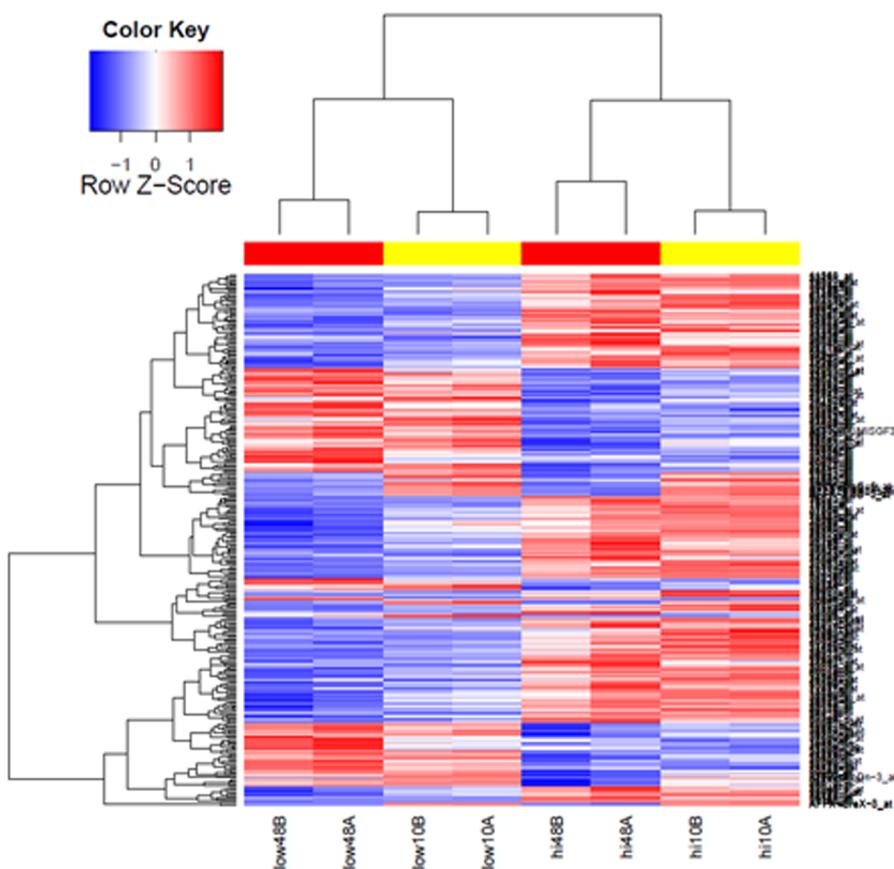


Figura 31. Heatmap para los genes seleccionados como diferencialmente expresados en alguna de las tres comparaciones

7. Métodos avanzados: Busca de patrones y predicción

7.1. Descubrimiento de grupos en datos de microarrays

El análisis de clústeres, conocido también como *class discovery*, fue en los primeros tiempos de los microarrays el método más popular de entre los empleados para tratar de identificar y agrupar genes expresados de forma similar y correlacionar los resultados con los procesos biológicos. En el apartado 3.7.2 ya se hizo una mención al análisis de conglomerados. Veremos aquí cómo se puede aplicar esta metodología para poder agrupar los genes.

La idea subyacente en esta aproximación era (es) que cuando dos genes estén corregulados y por lo tanto, funcionalmente relacionados, es probable que se expresen de forma simultánea; es decir, cuando la expresión de uno de ellos aumente, también aumentará la del otro, con lo que será posible agruparlos.

La figura 32 es un ejemplo de cómo 6 genes distintos presentan perfiles de expresión similares a lo largo de múltiples condiciones experimentales en un experimento de respuesta al estrés en levadura.

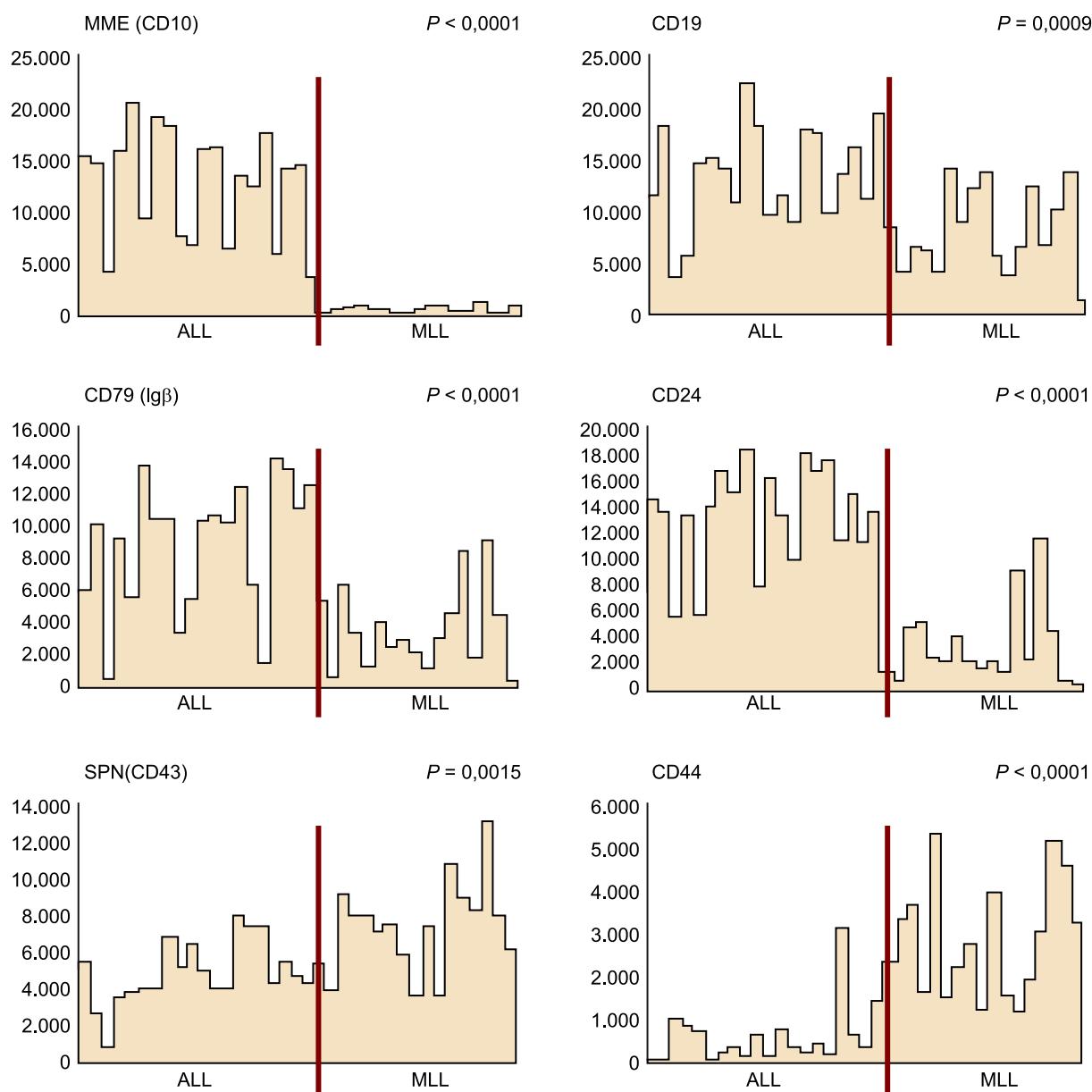


Figura 32. Comparación de los perfiles de expresión de 6 genes (MME, CD19, CD79, CD24, SPN, CD44) entre muestras de dos variantes de leucemia (AML y ALL).

A partir de la aplicación de un método de agrupación, suele ser posible la representación de los grupos en un plano con lo que se consigue una representación sencilla de una matriz de datos de alta dimensión que permite visualizar los datos de forma intuitiva, de forma parecida a lo que permite el análisis de componentes principales.

El análisis de conglomerados se ha aplicado tanto a la clasificación de genes (o variables) como de arrays (o muestras). Básicamente, la clasificación de los genes persigue:

- La identificación de grupos de genes corregulados.
- La identificación de patrones de expresión espacial o temporal.
- La reducción de la redundancia en modelos predictivos.

Si por el contrario se agrupan las muestras, el resultado nos permitirá:

- Identificar nuevas clases biológicas (p. ej. nuevas clases de tumor).
- Detectar artefactos experimentales.

Usualmente se realiza la agrupación simultánea de los arrays y los genes, es decir, de las columnas y las filas de la matriz de expresión tal como se muestra en la figura 32.

7.1.1. Principios y métodos para el descubrimiento de clases

El análisis de conglomerados es una técnica estadística clásica, o mejor dicho, un amplio abanico de métodos y técnicas cuyo rasgo común es la búsqueda de patrones en los datos que permitan juntar los más similares entre sí, diferenciándolos de los más heterogéneos.

Prácticamente todos los métodos de agrupación se han utilizado de una forma u otra para el análisis de datos de microarrays pero sin duda los métodos más populares son el clúster jerárquico entre los métodos aglomerativos y el método de las k -medias entre los divisivos. La diferencia entre ambas aproximaciones se muestra en la figura 33.

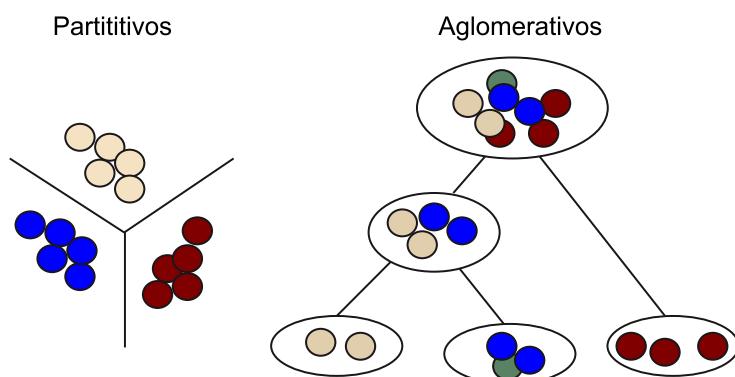


Figura 33. Métodos aglomerativos y divisivos
Los métodos aglomerativos parten de una situación en la que cada grupo inicial está formado por un solo elemento y en el proceso los va agrupando hasta llegar a un único grupo formado por todos los elementos. En los métodos divisivos se parte de un grupo con todos los elementos, que se va dividiendo en subgrupos.

El análisis de conglomerados siempre considera dos aspectos complementarios:

- La medida de distancia que se utilizará para cuantificar la similaridad entre filas o columnas.
- El algoritmo de agrupación con el que se determinarán qué datos son similares o cuáles no lo son.

A continuación se repasan brevemente cada una de estas componentes centrándose en sus aplicaciones a los microarrays.

Medidas de distancia

Dados dos valores x , y de un espacio n dimensional, una distancia es una función de dichos puntos que permite cuantificar su similaridad. Existen multitud de distancias y algunas de las más conocidas son:

1) La distancia euclídea

$$d_{Euc}(x - y) = \sqrt{\sum_{i=1}^n |x_i - y_i|^2}$$

2) La distancia valor absoluto o distancia ciudad

$$d_{abs}(x - y) = \sum_{i=1}^n |x_i - y_i|$$

3) La distancia correlación

$$d_{cor}(x - y) = 1 - cor(x, y)$$

donde $cor(x, y)$ representa el coeficiente correlación lineal entre x e y .

La elección de la distancia es un tema importante que influye en los resultados de la agrupación. Por ejemplo, la distancia euclídea detecta bien diferencias absolutas, mientras que la correlación es mejor para detectar cambios de tendencia (si unos “suben” u otros “bajan” esta distancia será pequeña independientemente de la diferencia entre unos y otros).

Algoritmos de agrupación

Podemos distinguir distintos métodos:

1) **Métodos jerárquicos.** El clúster jerárquico es sin duda la herramienta más conocida entre los usuarios de los microarrays. Fue popularizado por los primeros estudios realizados por Eisen ([15]), quien, además, proporcionaba un programa libre para agrupar y visualizar genes, lo que le dio una gran difusión. Este método se ha utilizado principalmente para la agrupación de las muestras más que de los genes, ya que al ser su número mucho menor, facilita la interpretación del dendrograma obtenido.

Aunque los métodos jerárquicos son superados en calidad y precisión por muchas otras técnicas de agrupación, existe una técnica de visualización que hace que su uso continúe. Se trata de los mapas de color o *heatmaps* (ver Gentleman y otros [18], capítulo 10 y apartado 5.4 de este documento), que consiste en

un rectángulo dividido en filas y columnas que constituyen bloques que están coloreados, donde el color de cada uno de estos bloques representa un nivel de expresión de un gen en un array (véase la figura 31 del apartado 6.4.4.).

2) El método *k-means*. Este método (ver Kaufman & Rosseeuw [26]) es asimismo muy popular a pesar de tener la desventaja de que es un algoritmo que comienza con una muestra de k genes elegidos al azar de la matriz original de datos. Cada uno de ellos se utiliza como el centroide inicial de los k clústeres que se van a formar, lo que hace que el resultado final sea muy sensible a estas elecciones. Cada clúster inicial está representado por un elemento llamado centroide. En este caso el investigador debe probar con distintas cantidades de clústeres iniciales (k) y a continuación seleccionar la k que mejor se ajusta a los datos.

Los dos métodos de *clustering* comentados presentan un problema que resulta difícil de resolver, y es que el orden de los genes para un clúster dado y el orden en que se muestran los clústeres no aportan ningún tipo de información biológica útil.

Esto implica que algunos clústeres que quedan representados cerca unos de otros pueden ser menos parecidos entre sí que con otros clústeres que aparecen en posiciones más lejanas.

3) Otros métodos de agrupación. Entre la multitud de métodos existentes podemos destacar *partition around medoids (PAM)* similar al *K-means* pero más robusto y sin inicialización aleatoria o los mapas autoorganizativos (*self-organizing maps (SOM)* [10] que se han aplicado con bastante éxito a datos de microarrays.

Los mapas autoorganizados (*self-organizing maps*) se obtienen a partir de la aplicación de los métodos basados en redes neuronales. El algoritmo permite, de forma iterativa, que los patrones más parecidos se vayan juntando entre sí y alejándose de aquellos otros que son diferentes. Este tipo de algoritmos son más fiables y robustos, puesto que se basan en redes neuronales que por definición son capaces de trabajar con grandes cantidades de datos con ruido. Sin embargo, no carece de ciertos inconvenientes. SOM es una herramienta particularmente útil en el tratamiento de datos procedentes de series temporales. De cualquier modo, cada uno de ellos tiene sus propios inconvenientes, y para muchos usuarios la agrupación jerárquica sigue siendo la opción favorita.

7.1.2. Consistencia de la agrupación

Número de clústeres

Un problema complejo en el análisis de clústeres es el saber cuántos clústeres existen realmente, dado que debemos recordar que la pertenencia de un individuo a un clúster no está fijada de antemano, sino que la establece el algorit-

mo. Se trata de un problema importante pero complejo, y muchos autores han propuesto diversas aproximaciones para resolverlo (véase, por ejemplo Milligan y Cooper [34]).

Uno de los enfoques más populares es el del gráfico de silueta (Silhouetteplot introducido por Rousseeuw ([38]) o el de la silueta promedio (Average Silhouette) donde Kaufman y Rousseeuw ([26]) amplían lo previamente publicado.

Algunos de estos métodos se han aplicado con éxito a datos de microarray. En otros casos se han desarrollado extensiones específicas para adaptarse mejor a sus particularidades:

- Yeung y otros ([50]) y Mc Lachlan propusieron distintos tipos de métodos basados en modelos.
- Hastie y otros ([21]) introdujeron el estadístico “GAP” como base para la selección del número de clústeres.
- Dudoit y Fridlyand ([11]) propusieron un algoritmo llamado Clest, que usa el remuestreo para estimar la cantidad de clústeres en base a la precisión de la predicción. El método se puede usar con cualquier algoritmo de agrupamiento y parece estar mejor adaptado para agrupar muestras que para agrupar genes.

Validación de las agrupaciones

Cuando se realiza un *clustering* de muestras, el dendrograma puede dar una idea acerca de la similaridad y relación entre las muestras, pero no nos indica la robustez del método frente a la variabilidad asociada al proceso de muestreo.

Con el fin de extraer conclusiones válidas de la estructura de la agrupación subyacente en los datos, es necesario investigar cómo afecta la variabilidad a los resultados del análisis del clúster.

Por otro lado, la evaluación de la validez de los clústeres obtenidos es especialmente importante cuando se agrupan datos de microarrays. El hecho de que las proteínas se organicen en vías y los genes estén corregulados sugiere que el perfil de expresión de un gran número de genes poseen una estructura de agrupación natural. Por lo tanto, existen una serie de clústeres “reales” que deberían ser descubiertos.

La dificultad en la validación de clústeres es que no hay una clasificación inicial contra la que se puedan comparar los resultados del *clustering*. Una forma de abordar este problema es examinar las relaciones entre los resultados

del *clustering* y algunas variables externas que no hayan sido utilizadas previamente, siempre y cuando ello sea posible. Además se pueden haber generado algunos clústeres que no tengan relación con estas variables.

Un enfoque común para evaluar la validez del clúster es utilizar algún tipo técnica de remuestreo, tal como el método de bootstrap desarrollado por Kerr y Churchill ([28]).

La *Figure of Merit* (FOM) introducida por Yeung y otros ([50]) es otro enfoque puramente experimental ampliamente utilizado en otros contextos, que ha sido validado específicamente para datos de microarrays.

Comentarios finales

Hasta el momento se ha considerado la utilización de los métodos de la agrupación para encontrar grupos de genes corregulados o muestras relacionadas que pertenecen a grupos no identificados previamente.

En concreto podemos encontrar una agrupación de muestras que sugiere que, donde inicialmente no parecía haber subgrupos, han aparecido grupos asociados de alguna forma al proceso en estudio (es decir, obtenemos agrupaciones de muestras/individuos que no conocíamos previamente). Este tipo de agrupación se realiza principalmente con el conjunto de genes en el cual se ha observado un cambio, por ejemplo, los genes diferencialmente expresados.

Sin embargo, existe otra aplicación importante del *clustering* de muestras, que se realiza en todos los genes, y no solo en aquellos que han sido seleccionados. El objetivo sería agrupar los datos iniciales (normalizados) para descubrir patrones, probablemente debido a algún efecto sistemático (bloque). Puede haber múltiples fuentes de esta variación sistemática: lotes de producción, técnicas, de origen biológico (líneas celulares), etc. Una visualización adecuada de los clústeres obtenidos puede ayudar a descubrir la existencia de estos efectos.

Después de detectar tales efectos no esperados, es posible incluirlos en el modelo utilizado para la detección de genes expresados diferencialmente, por lo que se pueden estimar su efecto de forma separada de los factores importantes de manera que no afecten a los resultados finales del análisis.

7.2. Diagnósticos moleculares y métodos de clasificación

El objetivo de la predicción de una clase o grupo (en análisis de microarrays como en la mayoría de los problemas de clasificación) es desarrollar una función multivariante para predecir con exactitud la pertenencia de clase (fenotípico) de un nuevo individuo. Es decir, si a cada objeto (paciente, muestra, chip) i se le asigna:

- una etiqueta de clase (o respuesta) $Y \in \{1, 2, \dots, K\}$ y
- un vector de características de las variables de predicción (medición de la expresión de g genes), $\mathbf{X} = (X_1, \dots, X_g)$,

el objetivo consiste en obtener la predicción de $Y(\mathbf{X})$ para un nuevo individuo sin clasificar.

Desde el punto de vista biomédico, es importante distinguir entre la predicción de clase (asignación de una nueva muestra a categorías existentes) y la de pronóstico (predicción de la evolución de la enfermedad de un paciente).

Un ejemplo de predicción es el estudio de clasificación molecular de pacientes afectados de distintas variantes de leucemia realizado por Golub y otros ([20]) (figura 34a), mientras que un ejemplo de pronóstico puede ser la construcción de un indicador para determinar qué tumores pueden desarrollar la metástasis después de un cierto período de tiempo estudiado por Van't Veer y otros ([47]) (figura 34b). Díaz-Uriarte [9] contiene una revisión de diversos aspectos relacionados con la construcción y validación de predictores a partir de datos genómicos.

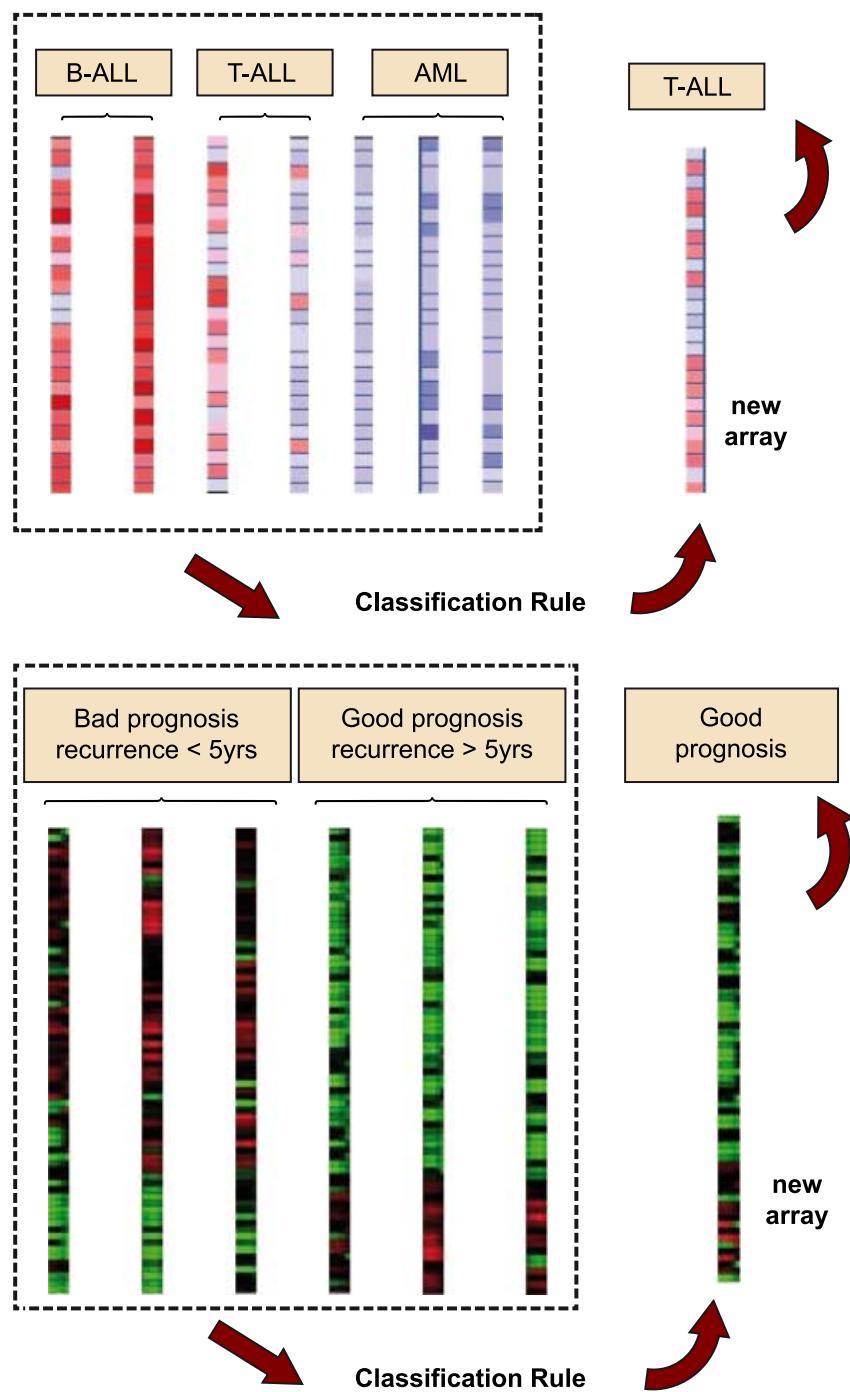


Figura 34
Dos ejemplos para ilustrar la clasificación de problemas en el análisis de los datos de microarrays. a. Ejemplo de predicción de clase: Asignación del tipo de patología a un nuevo paciente. Tomado de Golub y otros ([20]). b. Predicción de desarrollo de metástasis

Este subapartado está organizado de la siguiente manera: en primer lugar se presentan brevemente algunos de los principales métodos de clasificación, con énfasis en su aplicación al análisis de microarrays. A continuación se discute el problema de la selección de variables para la predicción y se dan algunas ideas sobre cómo medir el rendimiento de un clasificador. Finalmente, se concluye con una breve discusión sobre algunas cuestiones específicas de la predicción de clase con datos de expresión y reproduciendo algunos “consejos prácticos” para los usuarios.

Métodos de predicción de clase

El número de métodos de clasificación disponible es muy alto, probablemente debido al hecho de que es un término muy general que puede abarcar desde una “simple regresión” logística a un complejo sistema de máquinas de vectores soporte multicategóricas. Veámoslos:

- **Análisis discriminante.** Uno de los métodos más populares entre los estadísticos es sin duda el análisis discriminante ([8]), que permite clasificar los resultados binarios o múltiples usando una función lineal o cuadrática –llamada función discriminante– de las variables continuas, que, bajo supuestos de normalidad, se puede obtener mediante maximización de la razón de la verosimilitud entre grupos frente a la verosimilitud dentro de los grupos. En análisis de microarrays se han utilizado con éxito dos variantes del análisis discriminante. Uno de ellos es el *análisis discriminante lineal diagonal* (DLDA), que proporciona la discriminación óptima cuando la matriz de varianzas-covarianzas se supone diagonal. Otra es el algoritmo de la ponderación de los votos *weight voting algorithm*, introducidos por Golub y otros ([20]), que se ha convertido en relativamente popular y viene a ser una variante de DLDA ([13]).
- **Métodos de vecino más próximo.** Los métodos del vecino más cercano o de *nearest neighbor* (KNN) también se han utilizado profusamente probablemente debido a su simplicidad. Estos métodos consisten en predecir el grupo de un caso de prueba mediante la mayoría de votos entre los (k) vecinos más cercanos a dicho caso. El método es sencillo, intuitivo y no realiza ninguna suposición sobre los datos.
- **Las máquinas de vectores soporte.** Estos métodos, también denominados *support vector machines* (SVM), provenientes del campo del aprendizaje automático o *machine learning* han tenido también bastante éxito en el campo de los microarrays a pesar de que, en contraste con los métodos del vecino más próximo, no son en absoluto intuitivos. La idea de los SVM es que si no puede separar bien dos grupos en un espacio lo intenta proyectando los valores a un espacio de mayor dimensión hasta obtener el mejor hiperplano de separación entre las clases, definido como aquel que garantiza que hay una distancia máxima entre el hiperplano y el punto más cercano de cualquiera de las clases. Incluso cuando no hay hiperplano de separación, SVMs puede producir clasificadores decentes tratando de maximizar el margen y permite algunos errores de clasificación sujeta a la restricción de que el error total (distancia del hiperplano en el “lado equivocado”) es menor que una constante. La flexibilidad y versatilidad de la SVM ha hecho una opción muy popular entre los profesionales, pero su característica de “caja negra”, así como el hecho de que son más difíciles de entender que los enfoques más simples, probablemente ha restringido su extensión.

- **Otros métodos.** Existen muchos métodos más, desde métodos estadísticos tradicionales, como la regresión logística a métodos modernos más sofisticados, como los *random forests*, para no hablar de todos los métodos desarrollados *ad hoc* para el análisis de datos de expresión como es el análisis de predicción de microarrays (PAM) o el de *gene shaving*.

En el problema de la predicción de la clase en microarrays, como es el caso en otros campos, se ha hecho un uso extensivo de los métodos de agregación, esto es la combinación de varios predictores para obtener clasificadores mejorados. La agregación se sugirió por primera vez por Breiman ([5]), quien encontró que el aumento en la precisión se podría obtener por agregación de factores predictivos construidos a partir versiones perturbadas del conjunto de aprendizaje. Bagging ([5]) y Boosting ([17]) son dos métodos de agregación basados en el remuestreo que se han aplicado con éxito relativo en microarrays.

Comparación entre los métodos

Dado el número y la diversidad de los métodos disponibles una de las primeras preocupaciones de un usuario potencial de los métodos de predicción de la clase es cuál debe utilizar para cada problema.

Para ayudar a responder a esta pregunta, Dudoit ([12]) hizo una comparación de varios métodos de clasificación populares. Su principal conclusión fue que los clasificadores simples, tales como análisis discriminante lineal diagonal (DLDA) y el vecino más cercano (NN) funcionan notablemente bien en comparación con los más sofisticados, como los árboles de la clasificación agregada.

Won Lee y otros ([48]) ampliaron el análisis anterior incluyendo más métodos (hasta 21) y más conjuntos de datos (7). En algunos aspectos alcanzaron conclusiones similares a las de ([12]), aunque acabaron argumentando que los métodos más complejos, como las máquinas soporte vector, obtenían un mejor rendimiento.

A pesar de todo lo dicho, después de más de 10 años de aplicaciones en este campo, todavía no existe un consenso claro sobre cuál es el método más adecuado para cada caso. Si que hay, sin embargo, un cierto consenso sobre qué se debe evitar. Allison ([2]) o Dupuy y Simon ([14]) enumeran algunas buenas prácticas que conviene seguir cuando se utilizan los métodos de predicción. Recientemente se ha llevado a cabo un gran estudio, el “MAQC-II” (“Micro Arrays Quality Control-II”, [7]), del que se han derivado también algunas conclusiones sobre qué es lo que influye, positiva o negativamente, en la calidad y fiabilidad de un predictor hecho con datos de microarrays.

7.2.1. La selección de variables para la clasificación

Con el fin de construir un predictor uno debe decidir qué variables va a utilizar. Esto no es un problema trivial, ya que esta selección va a guiar todo el proceso e influirá considerablemente en los resultados. Si el número de variables es pequeño, no es difícil elegir entre ellas o simplemente utilizarlos todos. Sin embargo, cuando hay miles de variables para escoger la selección, se convierte en una difícil labor.

Algunos métodos, como SVM, KNN o DLDA, pueden utilizar tantas variables como se les suministre. Otros métodos, tales como la regresión logística, o regresión de Cox, no pueden, debido a lo que se conoce como la “maldición de la dimensionalidad” o problema “ $p >> n$ ”, que es el hecho de que el número de variables (= funciones = genes) (p) es mucho mayor que el número de muestras (n). En cualquier caso, el uso de algún procedimiento para la preselección de genes se considera que beneficia el rendimiento de la predicción. Existen muchos métodos de selección pero no se discutirán aquí. Una buena revisión se encuentra disponible en [31].

Evaluación del clasificador

Siempre es posible, dados unos datos y unas etiquetas de clase, construir un clasificador que asigne cada individuo a una clase. Sin embargo, en la práctica, lo que se necesita es alguna garantía de que el clasificador que se va obtener es lo suficientemente bueno, asumiendo que la obtención del predictor “óptimo” suele ser imposible salvo en casos puntuales. Para decidir la calidad de un predictor solemos basarnos en dos conceptos:

- Su discriminabilidad, es decir, lo bien que predice observaciones “nuevas” obtenidas de forma independiente de aquellas con las que se ha construido el predictor,
- Su fiabilidad, entendido como la capacidad de proporcionar previsiones (clasificaciones) similares cuando los datos presentan (pequeñas) variaciones no atribuibles a los grupos que se clasifican.

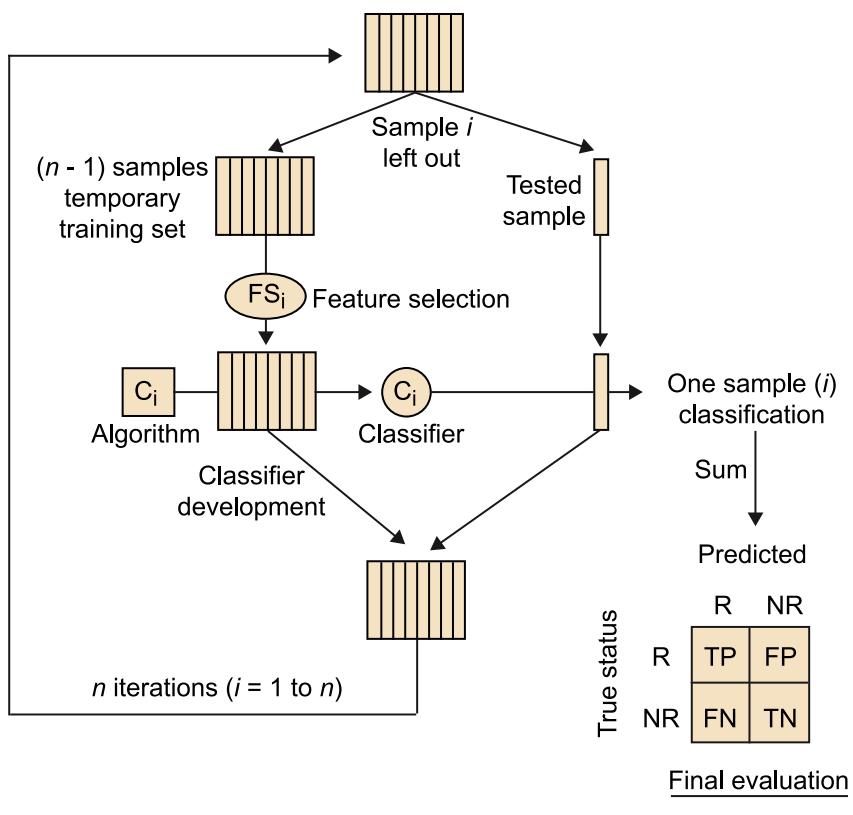
En la práctica, muchos usuarios se basan en algún tipo de tasa de error para evaluar la discriminabilidad del predictor, es decir, en el porcentaje de clasificación errónea frente al de predicciones correctas obtenidas por el clasificador.

Otro aspecto importante es que cualquier regla de clasificación tiene que evaluarse no por su capacidad de predecir las observaciones con que ha sido desarrollada sino por cómo predice nuevas observaciones independientes de las anteriores.

Ahora bien, en la práctica casi nunca se dispone de un grupo de datos independiente con el que evaluar el clasificador, por lo que se reutilizan los datos originales mediante algún procedimiento –que se denomina de validación cruzada– que permita simular de alguna forma la independencia entre las muestras. El método más conocido es el *leave-one-out* (LOO), consistente en que, si se tiene una muestra de n individuos, construir n predictores con $n - 1$ individuos, dejando uno fuera cada vez y clasificándolo con la regla construida sobre los $n - 1$ restantes.

La recomendación de la mayoría de los expertos como Simon y otros ([41]) es que no solo es preciso utilizar algún sistema de validación cruzada, sino que este debe integrarse en todo el proceso de construcción del predictor; es decir, no debe realizarse validación cruzada en una etapa y en otra no, sino que todas las fases de construcción y validación deben someterse a este proceso.

Las figuras 35 y 36, inspiradas en Dupuy ([14]), muestran de forma simplificada dos esquemas habituales de validación cruzada.



1 Fully developed classifier

Figura 35. Esquema de validación *leave-one-out* adaptado de [41]. Este método consiste en realizar tantas iteraciones como elementos contiene la muestra EN cada iteración se elimina a un individuo distinto de la muestra, se construye el predictor con los restantes y se clasifica al individuo eliminado. Este proceso se repite n veces y el porcentaje de malas clasificaciones se utiliza para estimar la fiabilidad del clasificador.

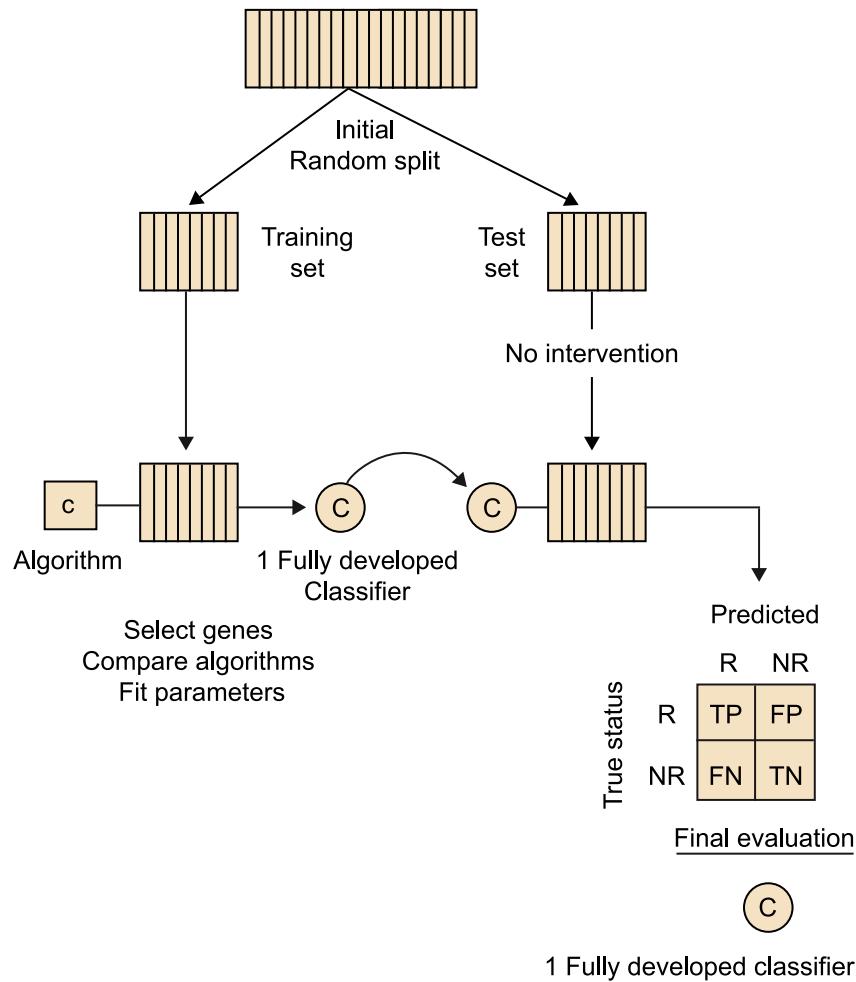


Figura 36. Esquema de validación *random split* adaptado de [41]. Este método consiste en dividir los datos originales en dos grupos escogidos aleatoriamente. Uno se utiliza para construir el predictor y el otro para estimar la probabilidad de error. Este proceso se repite muchas veces y el porcentaje promedio de malas clasificaciones estima la fiabilidad del clasificador.

8. Caso resuelto: Descubrimiento de clases

8.1. Clustering y visualización

En este caso se presentan distintos métodos de agrupación (*clustering*) disponibles en R y Bioconductor. Se utilizará el conjunto de datos *yeast* que puede descargarse desde la página del proyecto Yeast Cell Cycle Project (Proyecto de estudio del ciclo celular de la levadura): <http://genome-www.stanford.edu/cellcycle/data/rawdata/>.

Para descargarlo se tiene que acceder a la dirección mencionada y, desde ella, seleccionar el enlace *Tab delimited data* y descargar el archivo *combined.txt*.

8.1.1. Preprocesamiento de los datos

La agrupación de datos se suele aplicar después del preprocesamiento y filtraje de los mismos. El conjunto de datos de ejemplo ya ha sido normalizado y filtrado por lo que no nos ocuparemos de este aspecto.

El primer paso, como de costumbre, será establecer el directorio de trabajo que contiene los datos. El archivo de datos contiene información de varios experimentos (factor alfa, *cdc15* y cinéticas de decantación). Para simplificar este ejercicio nos ocuparemos únicamente de los datos denominados *cdc15* (*cell division control protein15*).

El primer paso consiste en extraer los valores del grupo *cdc15* del conjunto de datos pre-normalizados y eliminar los genes con valores faltantes:

```
> workingDir <- getwd()
> dataDir <- file.path(workingDir, "data")
> d <- read.table(file.path(dataDir, "combined.txt"), sep="\t", header=T)
> names(d)

[1] "X" "cln3.1" "cln3.2" "clb" "clb2.2" "clb2.1"
[7] "alpha" "alpha0" "alpha7" "alpha14" "alpha21" "alpha28"
[13] "alpha35" "alpha42" "alpha49" "alpha56" "alpha63" "alpha70"
[19] "alpha77" "alpha84" "alpha91" "alpha98" "alpha105" "alpha112"
[25] "alpha119" "cdc15" "cdc15_10" "cdc15_30" "cdc15_50" "cdc15_70"
[31] "cdc15_80" "cdc15_90" "cdc15_100" "cdc15_110" "cdc15_120" "cdc15_130"
[37] "cdc15_140" "cdc15_150" "cdc15_160" "cdc15_170" "cdc15_180" "cdc15_190"
[43] "cdc15_200" "cdc15_210" "cdc15_220" "cdc15_230" "cdc15_240" "cdc15_250"
[49] "cdc15_270" "cdc15_290" "cdc28" "cdc28_0" "cdc28_10" "cdc28_20"
[55] "cdc28_30" "cdc28_40" "cdc28_50" "cdc28_60" "cdc28_70" "cdc28_80"
```

```
[61] "cdc28_90" "cdc28_100" "cdc28_110" "cdc28_120" "cdc28_130" "cdc28_140"
[67] "cdc28_150" "cdc28_160" "elu" "elu0" "elu30" "elu60"
[73] "elu90" "elu120" "elu150" "elu180" "elu210" "elu240"
[79] "elu270" "elu300" "elu330" "elu360" "elu390"

> cdc15 <- which(substr(names(d),1,6)=="cdc15_")
> dat <- d[,cdc15]
> names(dat)

[1] "cdc15_10" "cdc15_30" "cdc15_50" "cdc15_70" "cdc15_80" "cdc15_90"
[7] "cdc15_100" "cdc15_110" "cdc15_120" "cdc15_130" "cdc15_140" "cdc15_150"
[13] "cdc15_160" "cdc15_170" "cdc15_180" "cdc15_190" "cdc15_200" "cdc15_210"
[19] "cdc15_220" "cdc15_230" "cdc15_240" "cdc15_250" "cdc15_270" "cdc15_290"

> rownames(dat) <- d$X
> dat <- na.omit(dat)
```

Los datos descargados de la web habían sido previamente normalizados. Para comprobarlo podemos realizar un boxplot y comprobar que los datos presentan una distribución similar y centrada en el cero como sería de esperar con valores de expresión relativa normalizados.

```
> boxplot(dat, las=2, cex.axis=0.7)
```

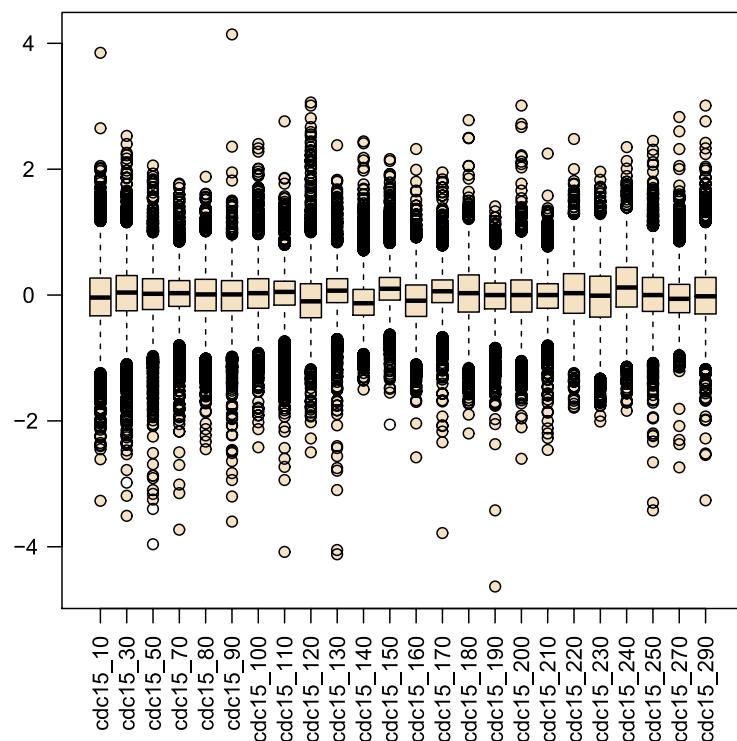


Figura 37. El boxplot de los datos seleccionados para el estudio de agrupación muestra que estos presentan una distribución similar y centrada en el cero, como es de esperar en datos de expresión relativa normalizados.

A continuación seleccionaremos solo aquellos genes que se encuentran entre el 1% de las desviaciones estándar más altas:

```
> percentage <- c(0.99)
> sds <- apply(dat, MARGIN=1, FUN="sd")
> sel <- (sds>quantile(sds,percentage))
> dat.sel <- dat[sel, ]
> dim(dat.sel)
[1] 44 24
```

8.1.2. Agrupación jerárquica de las muestras

La primera aproximación que suele hacerse para establecer la relación entre muestras es la aplicación de algun método de agrupamiento jerárquico. En este caso se realizará un clúster jerárquico basado en distancias euclídeas y enlaces promedio (*average linkage*).

```
> distmeth <- c("euclidean")
> Distan <- dist(t(dat.sel), method=distmeth)
> treemeth <- c("average")
> hc <- hclust(Distan, method=treemeth)
```

Para representar la estructura jerárquica de un agrupamiento se utiliza el dendrograma, un gráfico que tiene forma de árbol invertido, donde los nombres de los genes equivaldrían a las hojas.

```
> plot(hc, hang=-1)
```

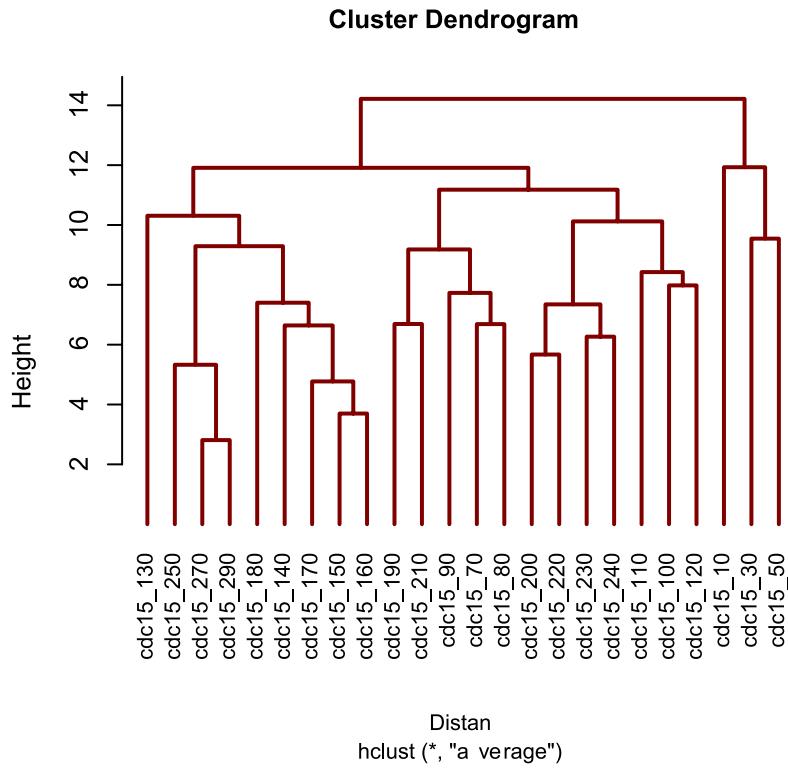


Figura 38. Dendrograma que refleja la similitud de las muestras agrupadas mediante clúster jerárquico.

Esta no es la única forma de realizar este tipo de agrupación y, en general, se recomienda probar con distintos métodos razonables y ver hasta qué punto cambian los resultados que se obtienen.

Visualización un dendrograma a partir de las correlación entre genes

A pesar de que en los ejemplos anteriores se han utilizado distancias euclídeas, los perfiles de expresión génica acostumbran a agruparse en función de los coeficientes de correlación.

Para ello es necesario calcular la correlación entre los genes y cambiar el formato de la matriz de correlación para que pueda contener las distancias. A continuación, el árbol puede dibujarse normalmente.

```
> cor.pe <- cor(as.matrix(dat.sel), method=c("pearson"))
> cor.pe.rows <- cor(t(as.matrix(dat.sel)), method=c("pearson"))
> cor.sp <- cor(as.matrix(dat.sel), method=c("spearman"))
> dist.pe <- as.dist(1-cor.pe)
> dist.pe.rows <- as.dist(1-cor.pe.rows)
> dist.sp <- as.dist(1-cor.sp)
> hc.cor <- hclust(dist.pe, method=treemeth)
> hc.cor.rows <- hclust(dist.pe.rows, method=treemeth)

> oldpar <- par(mfrow=c(2,1))
> plot(hc.cor, main="Agrupación de genes\n basada en la correlación de Pearson")
> plot(hc.cor.rows, main="Agrupación de muestras\n basada en la correlación de Pearson")
```

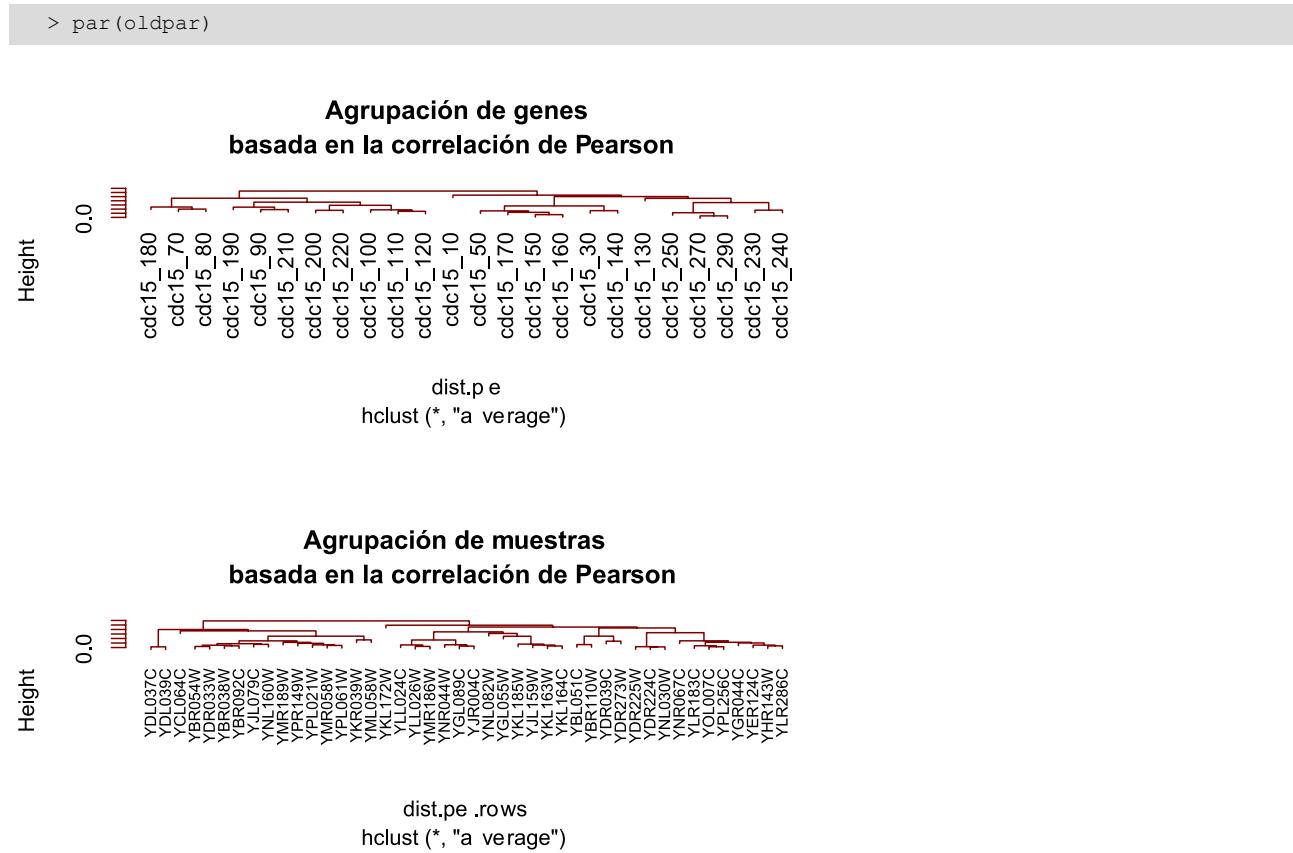


Figura 39. Dendrogramas mostrando la agrupación jerárquica de genes y de muestras basadas en la correlación de Pearson.

8.1.3. Agrupación de genes por el método de las k-means

Si en lugar de muestras deseamos agrupar por genes, es mejor usar métodos no jerárquicos partitivos como k-means.

Un inconveniente de la agrupación por k-means es que hace falta escoger el número de clusters (k) antes de realizar la agrupación.

Por ejemplo, para producir un agrupamiento k-means con 5 grupos, haremos:

```
> k <- c(5)
> km <- kmeans(dat.sel, k, iter.max=1000)
> #summary(km)
> #head(km)
```

El método de las k-means permite estudiar la consistencia de la agrupación a partir del cálculo de las sumas de cuadrados dentro de cada grupo (*within SS*). El promedio de estas sumas de cuadrados para todos los grupos creados (variable *withinss*) es una medida de la variabilidad dentro de los grupos, es decir, cómo son de parecidos los genes dentro de los grupos.

```
> mean(km$withinss)
```

Una manera de evaluar las agrupaciones generadas mediante *k-means* es ejecutar el mismo análisis varias veces –guardando cada vez el resultado como un nuevo objeto– y seleccionar aquella agrupación que ofrece un menor valor de *withinss*.

Dibujo de una agrupación k-means

Los resultados de los métodos divisivos como *k-means* no pueden visualizarse mediante un dendrograma, por lo que es preciso recurrir a otras representaciones gráficas.

Podemos, por ejemplo, pintar los miembros de cada grupo con un color y un símbolo distintos.

```
> cl <- km$cluster
> plot(dat.sel[,1], dat.sel[,2], col=cl)
> points(km$centers, col = 1:5, pch = 8)
```

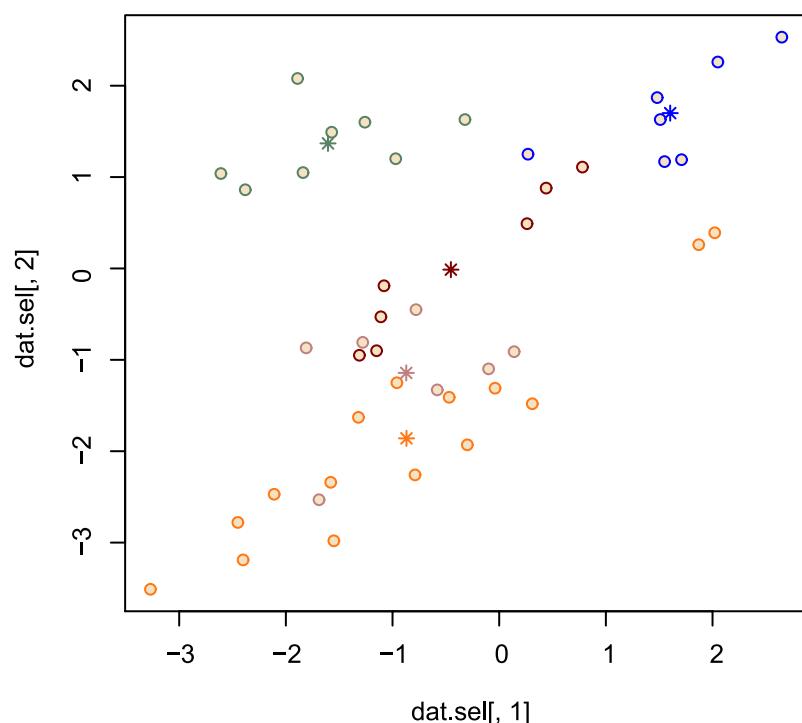


Figura 40

Para visualizar un solo grupo y sus valores de expresión, seleccionaremos los genes que lo constituyen y a continuación, dibujaremos los perfiles de expresión de estos genes utilizando diferentes colores.

```
> set3 <- data.frame(dat.sel, cl)
> cl1 <- dat.sel[which(cl==1),] # Here we have 3 genes
> plotcol <- colorRampPalette(c("Grey", "Black"))(5)
> plot(t(cl1[1,]), type="l", col=plotcol[1])
> lines(t(cl1[2,]), type="l", col=plotcol[3])
```

```
> lines(t(cl1[3,]), type="l", col=plotcol[5])
```

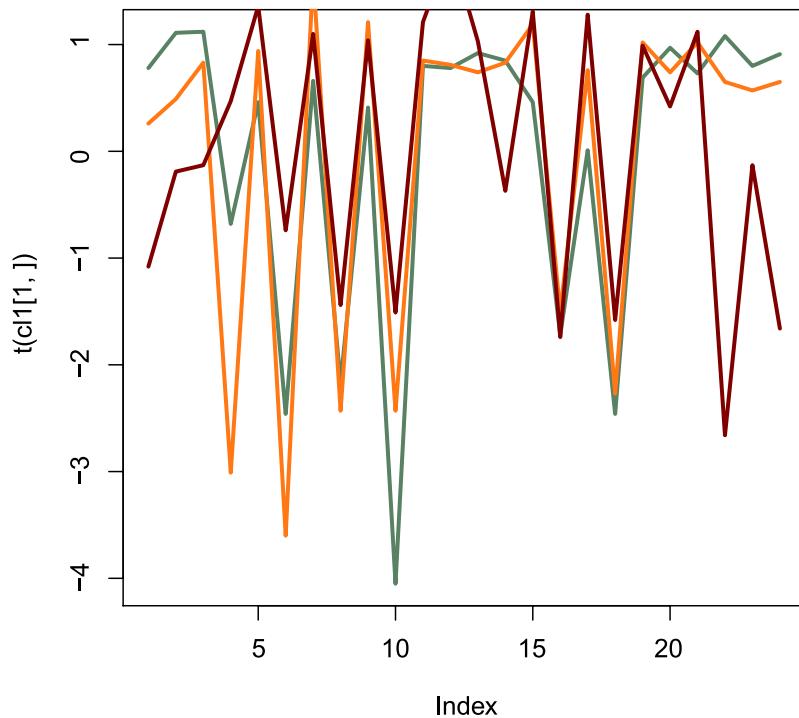


Figura 41

Si hay más de un grupo, podemos visualizarlos todos en el mismo gráfico utilizando un pequeño bucle `for`:

```
> km <- kmeans(dat.sel, 4, iter.max=1000)
> par(mfrow=c(2,2))
> for(i in 1:4) {
+   matplot(t(dat.sel[km$cluster==i,]), type="l",
+   main=paste("cluster:", i), ylab="log expression", xlab="time")
+ }
```

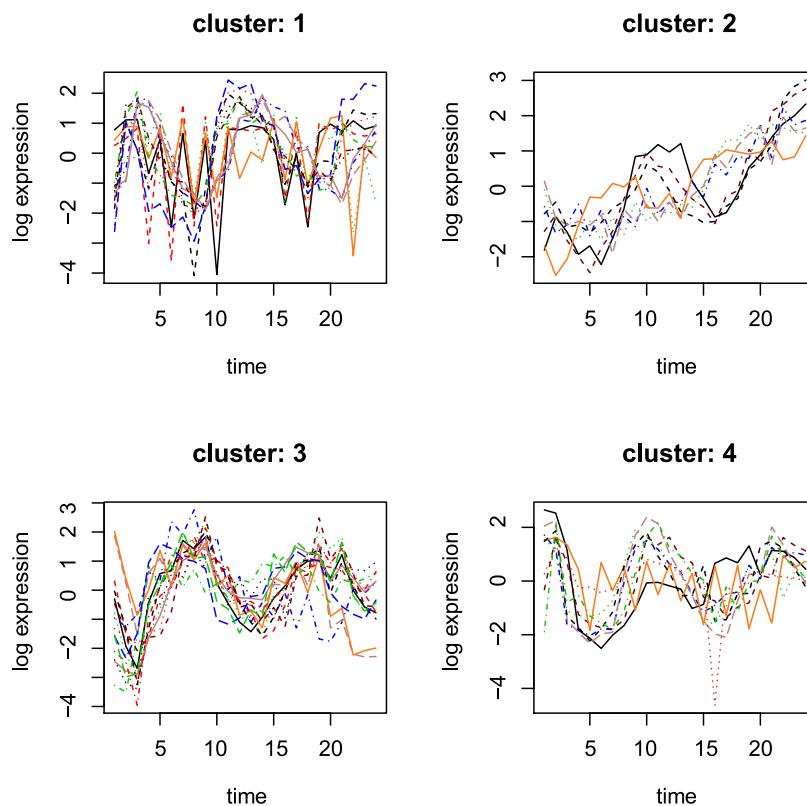


Figura 42

8.1.4. Anotación de resultados

Una vez que se han identificado algunos grupos de interés, se puede comprobar qué genes constituyen de cada uno de los clústeres. Esto puede hacerse con el paquete de anotaciones para levadura, de donde podremos extraer información sobre nombres, descripciones y otras anotaciones en múltiples bases de datos:

```
> if(!require(org.Sc.sgd.db)){
+ source("http://bioconductor.org/biocLite.R")
+ biocLite("org.Sc.sgd.db")
+
> require("org.Sc.sgd.db")
> anotData <- capture.output(org.Sc.sgd())
> print(anotData[1:15])

[1] "Quality control information for org.Sc.sgd:"
[2] ""
[3] ""
[4] "This package has the following mappings:"
[5] ""
[6] "org.Sc.sgdALIAS has 2559 mapped keys (of 8702 keys)"
[7] "org.Sc.sgdCHR has 8702 mapped keys (of 8702 keys)"
[8] "org.Sc.sgdCHRLLENGTHS has 17 mapped keys (of 17 keys)"
[9] "org.Sc.sgdCHRLOC has 8072 mapped keys (of 8702 keys)"
```

```
[10] "org.Sc.sgdCHRLOCEND has 8072 mapped keys (of 8702 keys)"
[11] "org.Sc.sgdCOMMON2ORF has 9565 mapped keys (of 9565 keys)"
[12] "org.Sc.sgdDESCRIPTION has 8348 mapped keys (of 8702 keys)"
[13] "org.Sc.sgdENSEMBL has 5881 mapped keys (of 8702 keys)"
[14] "org.Sc.sgdENSEMBL2ORF has 5888 mapped keys (of 5888 keys)"
[15] "org.Sc.sgdENSEMBLPROT has 5881 mapped keys (of 8702 keys)"

> cat ("... output continues until ", length(anotData), " lines.\n")

... output continues until 47 lines.
```

Resumen

Los microarrays de DNA, junto con otras tecnologías de alto rendimiento, han revolucionado el estudio de la biología a partir de la primera década del siglo XXI. El motivo principal es que estas tecnologías han permitido un cambio de enfoque en el estudio de procesos, como por ejemplo la expresión génica, pudiéndose utilizar para analizar cambios de expresión en cientos o miles de genes a la vez. El análisis de datos de microarrays ha evolucionado rápidamente como una disciplina que integra la biología, la estadística y la bioinformática, y combina las técnicas tradicionales de análisis de datos con nuevos desarrollos puestos a punto exclusivamente para estos nuevos tipos de datos. El análisis de datos de microarrays puede verse como un proceso que va desde la concepción del experimento hasta la generación de los resultados y su interpretación biológica.

Sus etapas principales son:

- 1) el diseño experimental, en donde se decide cómo asignar muestras a tratamientos para optimizar recursos;
- 2) la preparación de los microarrays, desde la hibridación de las muestras hasta las imágenes, base del análisis;
- 3) el control de calidad, que define la validez de los datos, y el preprocesado para eliminar ruido y generar datos comparables;
- 4) los diversos análisis estadísticos entre los que destaca la selección de genes diferencialmente expresados entre varias condiciones o el descubrimiento de grupos de muestras o de genes y
- 5) el análisis de significación biológica que sirve de soporte a la interpretación de los resultados.

Los microarrays de expresión génica han representado una revolución, pero parece que nuevas técnicas, como la ultrasecuenciación, podrían en poco tiempo permitir mejorar sus capacidades e incluso llegar a sustituirlos como herramientas de análisis.

Bibliografía

- [1] A. Alizadeh; M. B. Eisen; E. Davis; C. Ma; I. Lossos; A. Rosenwald; J. Boldrick; H. Sabet; T. Tran; X. Yu; J. I. Powell; L. Yang; G. E. Marti; J. Hudson Jr; L. Lu; D.B. Lewis; R. Tibshirani; G. Sherlock; W. C. Chan; T. C. Greiner; D. D. Weisenburger; J. O. Armitage; R. Warnke; R. Levy; W. Wilson; M. R. Grever; J. C. Byrd; D. Botstein; P. O. Brown; L. M. Staudt (2000). "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling". *Nature* (núm. 403, págs. 503-511).
- [2] D. B. Allison; X. Cui; G. P. Page; M. Sabripour (2006). "Microarray data analysis: from disarray to consolidation and consensus". *Nat. Rev. Genet.* (vol. 1, núm. 7, págs. 55-65).
- [3] P. Baldi; A. D. Long (enero, 2001). "A Bayesian Framework for the Analysis of Microarray Expression Data: Regularized t -test and Statistical Inferences of Gene Changes". *Bioinformatics* (vol. 17, núm. 6, págs. 509-519). doi:10.1093/bioinformatics/17.6.509.
- [4] Y. Benjamini; Y. Hochberg (1995). "Controlling the false discovery rate: A practical and powerful approach to multiple testing". *Journal of the Royal Statistical Society B* (núm. 57, págs. 289-300).
- [5] L. Breiman (1996). "Bagging Predictors". *Machine Learning* (vol. 2, núm. 24, págs. 123-140).
- [6] R. L. Chelvarajan; Y. Liu; D. Popa; M. L. Getchell; T. V. Getchell; A. J. Stromberg; S. Bondada (2006). "Molecular basis of age-associated cytokine dysregulation in LPS-stimulated macrophages". *Journal of Leukocyte Biology* (vol. 6, núm. 79, págs. 1314).
- [7] MAQC Consortium (2010). "The MicroArray Quality Control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models". *Nat. Biotech.* (vol. 8, núm. 28, págs. 827-838).
- [8] C. M. Cuadras (1989). *Análisis Multivariante*. EUNIBAR.
- [9] R. Díaz-Uriarte (2005). "Supervised Methods with Genomic Data: a Review and Cautious View". *Data analysis and visualization in genomics and proteomics* (págs. 193-214). Nueva York: Wiley.
- [10] R. Duda; P. Hart; D. G. Stork (2001). *Pattern recognition* (2.^a ed.). Ed. John Wiley and Sons.
- [11] S. Dudoit; J. Fridlyand (2002). "A prediction-based resampling method for estimating the number of clusters in a dataset". *Genome Biology* (vol. 7, núm. 3, págs. 1-21).
- [12] S. Dudoit; J. Fridlyand; T. P. Speed (2002). "Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data". *Journal of the American Statistical Association* (núm. 97, págs. 457).
- [13] S. Dudoit; J. P. Shaffer; J. C. Boldrick (2003). "Multiple hypothesis testing in microarray experiments". *Statistical Science* (núm. 18, págs. 71-103).
- [14] A. Dupuy; R. M. Simon (2007). "Critical Review of Published Microarray Studies for Cancer Outcome and Guidelines on Statistical Analysis and Reporting". *JNCI Journal of the National Cancer Institute* (vol. 2, núm. 99, págs. 147).
- [15] M. B. Eisen; P. T. Spellman; P. O. Brown; D. Botstein (1998). "Cluster analysis and display of genome-wide expression patterns". *Proceedings of the National Academy of Sciences USA* (vol. 25, núm. 95, págs. 14863-14868).
- [16] J. J. Faraway (2004). *Linear Models with R* (1.^a ed.). Chapman and Hall/CRC.
- [17] Y. Freund; R. Schapire (1996). "Experiments with a new boosting algorithm, in Machine Learning: Proceedings of the Thirteenth International Conference" (págs. 148-156). San Francisco: Morgan Kauffman.
- [18] R. Gentleman; V. Carey; W. Huber; R. Irizarry; S. Dudoit (2005). *Bioinformatics and Computational Biology Solutions using R and Bioconductor*. Nueva York: Springer.
- [19] D. H. Geschwind; J. P. Gregg (2002). *Microarrays for the neurosciences: an essential guide*. MIT Press.
- [20] T. R. Golub; D. K. Slonim; P. Tamayo; C. Huard; M. Gaasenbeek; J. P. Mesirov; H. Coller; M. L. Loh; J. R. Downing; M. A. Caligiuri; C. D. Bloomfield; E. S. Lan-

der (1999). "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring". *Science* (núm. 286, págs. 531-537).

[21] **T. Hastie; R. Tibshirani; G. Walther** (2001). "Estimating the number of clusters in a dataset via the gap statistic". *Journal of the Royal Statistical Society, B* (vol. 41, núm. 63, págs. 1-423).

[22] **S. Imbeaud; Ch. Auffray** (2005). "The 39 steps' in gene expression profiling: critical issues and proposed best practices for microarray experiments". *Drug Discovery Today* (vol. 17, núm. 10, págs. 175-1182). PMID: 16182210.

[23] **R. A. Irizarry; B. Hobbs; F. Collin; Y. D. Beazer-Barclay; K. J. Antonellis; U. Scherf; T. P. Speed** (2003). "Exploration, normalization, and summaries of high density oligonucleotide array probe level data". *Biostatistics* (núm. 4, págs. 249-264).

[24] **M. van Iterson; J. M. Boer; R. X. Menezes** (2010). "Filtering, FDR and power". *BMC Bioinformatics* (núm. 11, pág. 450).

[25] **S.-H. Jung; S. S. Young** (2012). "Power and sample size calculation for microarray studies". *J Biopharm Stat* (núm. 22, págs. 30-42).

[26] **L. Kaufman; P. J. Rousseeuw** (1990). *Finding Groups in Data. An Introduction to Cluster Analysis*. John Wiley and Sons.

[27] **M. K. Kerr; G. A. Churchill** (2001). *Experimental design for gene expression microarrays*.

[28] **M. K. Kerr; G. A. Churchill** (2001). "Bootstrapping cluster analysis: Assessing the reliability of conclusions from microarray experiments". *Proceedings of the National Academy of Sciences*. PMID: 161273698.

[29] **P. Khatri; M. Sirota; A. J. Butte** (2012). "Ten Years of Pathway Analysis: Current Approaches and Outstanding Challenges". *PLoS Comput Biol* (núm. 8). e1002375.

[30] **P. Khatri; S. Draghici** (2005). "Ontological analysis of gene expression data: current tools, limitations and problems". *Bioinformatics* (núm. 18, págs. 3587-3595).

[31] **P. Larrañaga; B. Calvo; R. Santana; C. Bielza; J. Galdiano; I. Inza; J. A. Lozano; R. Armañanzas; G. Santafé; A. Pérez; V. Robles** (2006). "Machine learning in Bioinformatics". *Briefings in Bioinformatics* (vol. 1, núm. 7, págs. 86-112).

[32] **M. L. T. Lee; G. A. Whitmore** (2002). "Power and sample size for DNA microarray studies". *Statistics in Medicine* (vol. 23, núm. 21, págs. 3543-3570).

[33] **W.-J. Lin; H.-M. Hsueh; J. J. Chen** (2010). "Power and sample size estimation in microarray studies". *BMC Bioinformatics* (núm. 11, pág. 48).

[34] **G. W. Milligan; M. C. Cooper** (1985). "An examination of procedures for determining the number of clusters in a data set". *Psychometrika* (vol. 2, núm. 50, págs. 159-179).

[35] **Sh. Moore; J. Vrebalov; P. Payton; J. Giovannoni** (2002). "Use of genomics tools to isolate key ripening genes and analyse fruit maturation in tomato". *Journal of Experimental Botany* (vol. 377, núm. 53, págs. 2023-2030).

[36] **J.-L. Mosquera; A. Sánchez-Pla** (2005). "A comparative study of GO mining programs". X Conferencia Española de Biometría. Sociedad Española de Biometría.

[37] **J. Quackenbush** (2002). "Microarray data normalization and transformation". *Nature Genet.* (núm. 32, págs. 496-501).

[38] **P. Rousseeuw; E. Trauwaert; L. Kaufman** (1989). "Some silhouette-based graphics for clustering interpretation". *Belgian Journal of Operations Research, Statistics and Computer Science* (vol. 3, núm. 29, págs. 35-55).

[39] **A. Sánchez-Pla; J. L. Mosquera** (2007). "The Quest for Biological Significance". En: L. L. Bonilla; M. Moscoso; G. Platero; J. M. Vega (eds.). *Progress in Industrial Mathematics at ECMI 2006*. Nueva York: Springer.

[40] **D. Scholtens; A. Miron; F. M. Merchant; A. Miller; P. L. Miron; J. Dirk Iglehart; R. Gentleman** (2004). "Analyzing factorial designed microarray experiments". *J. Multivar. Anal.* (vol. 1, núm. 90, págs. 19-43).

[41] **R. M. Simon; E. L. Korn; L. M. McShane; M. D. Radmacher; G. W. Wright; Y. Zhao** (2003). *Design and Analysis of DNA Microarray Investigations*. Springer-Verlag.

[42] **G. K. Smyth; J. Michaud; H. S. Scott** (2005). "Use of within-array replicate spots for assessing differential expression in microarray experiments". *Bioinformatics* (vol. 9, núm. 21, págs. 2067-75).

[43] **G. K. Smyth** (febrero, 2004). "Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments". *Statistical Applications in Genetics and Molecular Biology* (vol. 3, núm. 1). <http://www.degruyter.com/view/j/sagmb.2004.3.1/sagmb.2004.3.1.1027/sagmb.2004.3.1.1027.xml>.

[44] **T. Speed** (2003). *Statistical Analysis of Gene Expression Data*. Boca Raton, Fla.: Chapman & Hall/CRC.

[45] **R. Tibshirani** (2006). "A simple method for assessing sample sizes in microarray experiments". *BMC Bioinformatics* (vol. 1, núm. 7, págs. 106).

[46] **V. G. Tusher; R. Tibshirani; G. Chu** (abril, 2001). "Significance analysis of microarrays applied to the ionizing radiation response". *Proceedings of the National Academy of Sciences of the United States of America* (vol. 98, núm. 9, págs. 5116-5121). doi:10.1073/pnas.091062498.

[47] **L. J. van't Veer; H. Dai; M. J van de Vijver; Yudong D. He; A. A. M. Hart; Mao Mao; H. L. Peterse; K. van der Kooy; Matthew J. Marton; Anke T. Witteveen; George J. Schreiber; Ron M. Kerkhoven; Chris Roberts; Peter S. Linsley; R. Bernards; Stephen H. Friend** (2002). "Gene expression profiling predicts clinical outcome of breast cancer". *Nature* (vol. 6871, núm. 415, págs. 530-536).

[48] **J. Won Lee; J. Bok Lee; M. Park; S. Song** (2005). "An extensive comparison of recent classification tools applied to microarray data". *Computational Statistics & Data Analysis* (vol. 4, núm. 48, págs. 869-885).

[49] **Y. H. Yang; M. J. Buckley; S. Dudoit; T. P. Speed** (2002). "Comparison of methods for image analysis on cDNA microarray data". *Journal of Computational and Graphical Statistics* (vol. 1, núm. 11).

[50] **K. Y. Yeung; C. Fraley; A. Murua; A. E. Raftery; W. L. Ruzzo** (2001). *Model-based clustering and data transformations for gene expression data*.

[51] **Q. Zhu; J. Miecznikowski; M. Halfon** (2010). "Preferred analysis methods for Affymetrix GeneChips. II. An expanded, balanced, wholly-defined spike-in dataset". *BMC Bioinformatics* (vol. 1, núm. 11, pág. 285).

