Análisis estadístico de datos obtenidos mediante qPCR y RT-qPCR utilizando métodos de remuestreo.

Álvaro Franquet
Máster Universitario en Bioinformática y Bioestadística
Estadística y Bioinformática

Sergi Civit Vives y Mireia Vilardell Nogales
Alexandre Sánchez Pla

26/12/2016


Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Análisis estadístico del fold-change en qPCR y RT-PCR utilizando métodos de remuestreo*
Nombre del autor:	*Álvaro Franquet Bonet*
Nombre del consultor/a:	*Sergi Civit Vives*
Nombre del PRA:	Alexandre Sánchez Pla
Fecha de entrega (mm/aaaa):	12/2016
Titulación::	*Máster Universitario en Bioinformática y Bioestadística*
Área del Trabajo Final:	*Estadística y Bioinformática*
Idioma del trabajo:	Español
Palabras clave	*qPCR, Bootstrap, R*

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

La reacción en cadena de la polimerasa en tiempo real (qPCR) y sus variantes cómo la retrotranscripción PCR en tiempo real (RT-qPCR) son técnicas de la biología molecular altamente utilizadas en la investigación biomédica para comparar las expresiones relativas (RE) de un grupo tratamiento contra un grupo control debido a su alta sensibilidad. Estas técnicas permiten ampliar fragmentos de ADN a partir de muestras biológicas incluso cuando la concentración de ADN presente en ellas sea baja. Así es posible comparar de forma eficiente, por ejemplo las RE, entre un grupo control y tratamiento.

Sin embargo, en algunas ocasiones, asumir una distribución de probabilidad concreta (usualmente la distribución Normal) para datos procedentes de estas técnicas o similares puede ser difícil de justificar debido a que habitualmente se obtienen un número de observaciones bajo y a la utilización de cocientes .

En este trabajo se presenta una metodología estadística para el análisis de RE obtenidas mediante RT-qPCR o para datos con una estructura similar, así cómo su implementación en la librería `testratio` creada con el lenguaje de programación R. La metodología aquí empleada se basa en técnicas de remuestreo, que evitan tener que asumir una distribución de probabilidad y, en concreto, las que aquí se presentan, admiten trabajar con un reducido número de observaciones.

Además se detalla la aplicación de estas técnicas y de la librería `testratio` con dos ejemplos reales.

La librería `testratio` tiene implementados los métodos aquí descritos y va

acompañada de dos herramientas más que consisten en un informe dinámico escrito en *Markdown* y una aplicación web *Shiny* para aquellos usuarios menos conocedores del entorno R. Esta aplicación web esta disponible en https://alvarofranquet.shinyapps.io/testrapptio/ o mediante la función `testrapptio()` incluida en el paquete `testratio`

Abstract (in English, 250 words or less):

The real-time polymerase chain reaction (qPCR) and its variants such as real-time PCR retrotranscription (RT-qPCR) are molecular biology techniques highly used in biomedical research to compare the relative expressions (RE) of a treatment group against a control group due to its high sensitivity. These techniques allow the amplification of DNA fragments from biological samples even when the concentration of DNA present in them is low. Thus it is possible to compare efficiently, for example RE, between a control and treatment group.

However, sometimes assuming a particular probability distribution (usually the Normal distribution) for data from these or similar techniques may be difficult to justify due the low number of observations obtained and the use of ratios.

This work presents a statistical methodology for the analysis of RE obtained by RT-qPCR or for data with a similar structure, as well as its implementation in the `testratio` library created with the programming language R. The methodology used here is based on techniques Of resampling, which avoid having to assume a probability distribution, and specifically those presented here, admit to working with a small number of observations.

In addition, the application of these techniques using the `testratio` library is described using two diferent examples. The `testratio` package implement the methods described here and is accompanied by two more tools consisting of a dynamic report written in Markdown and a Shiny web application for practitioners (those less knowledgeable about the R environment). This web application is available at https://alvarofranquet.shinyapps.io/testrapptio/ o via the `testrapptio()` function included in the `testratio` package.

Índice

Índice de figuras

Índice de tablas

1. Introducción

1.1 Contexto y Justificación del Trabajo

La técnica de *Real-time quantitative polymerase chain reaction* (qPCR) y sus variantes cómo la retrotranscripción en tiempo real (RT-qPCR) son altamente utilizadas en los estudios de investigación y diagnóstico. RT-qPCR se considera cómo un método fiable para determinar la expresión relativa (RE) de un gen en una muestra biológica en concreto. La RE se define cómo el cociente entre un gen candidato y una referencia (generalmente un gen de referencia), obteniendo, así, una RE para cada muestra biológica que se quiere estudiar. En este tipo de estudios, finalmente, lo que se desea conocer es si hay diferencias entre las RE de un grupo objetivo (llamado tratamiento) de las RE del grupo control. A pesar de la importancia que esta técnica ha alcanzado en la última década, aún queda por resolver la metodología estadística que se debería aplicar en estos casos. Es común observar, en la literatura científica relacionada, la asunción de distribuciones de probabilidad cómo la normalidad [1] subyacentes, sin embargo la utilización de cocientes implica considerar distribuciones no negativas y asimétricas que determinarían la no normalidad de los datos. Además el escaso número de datos (o observaciones) no permiten realizar con garantías los test de normalidad.

Teniendo en cuenta estas limitaciones, el Departamento de Estadística de la Universidad de Barcelona y el Grupo de investigación en prevención y control del cáncer - ICO del Instituto de Investigación Biomédica de Bellvitge (IDIBELL) se han basado en el uso de los métodos de remuestreo[2][3] para la selección y detección de genes estadísticamente diferenciados sin la necesidad de asumir una distribución probabílistica subyacente. En el presente trabajo se analiza y implementa la metodología desarrollada por el grupo de investigación y el departamento de estadística en el paquete de R `testratio` que contiene las funciones necesarias para llevar a cabo la obtención de la RE de los genes ,una función que permite generar de forma dinámica, automática y reproducible un *Markdown* con los datos introducidos y una aplicación *Shiny*[1] con el fin de facilitar el uso del paquete para aquellos investigadores no familiarizados con `R`.

1.2. Objetivos

I. Conocer los métodos propuestos por el grupo de investigación, analizarlos, completarlos y finalmente implementarlos para crear un paquete en R que permita realizar los estudios de significación.

II. Confeccionar una plantilla para el estudio de significación de forma automática.

III. Crear una aplicación interactiva basada en *Shiny* que permita facilitar al usuario la metodología descrita y poder emplearla sobre un conjunto de datos.

1.3. Enfoque y método a seguir

Para cumplir con los objetivos propuestos en el apartado anterior, se determinaron que técnicas estadísticas debían ser implementadas, como presentarlas al usuario final así como facilitar un conjunto de datos que permitiera al usuario el uso de las mismas y la familiarización con el paquete de R.

Desde la metodología estadística el proyecto planteo analizar varias posibilidades/estrategias como la de asumir una distribución para la expresión, evaluar diferentes distribuciones y seleccionar aquella que se ajuste más a unos criterios o utilizar métodos que no requieran de una distribución preestablecida.

Se decide enfocar este trabajo en el último punto, implementando los siguientes métodos de remuestreo (Bootstrap y Combn) que serán descritos y presentados con más detalle en el apartado 2.4.

- Bootstrap
- Combn

[1] http://shiny.rstudio.com/

En la parte referente a la creación del paquete se decie utilizar las herramientas en RStudio descritas en R Packages[4] y una combinación de Git[2] y Dropbox[3] para el control de versiones.

1.4. Planificación con hitos y temporización

La planificación de las tareas se ha realizado estimando una duración de 300 horas des de el inicio de la PEC1 (03/10/2016) hasta la entrega de la memoria (26/12/2016). El resto de actividades, aunque incluidas en la planificación no se consideran en ese computo de horas. Se excluyen de la planificación los fines de semana y los días festivos:

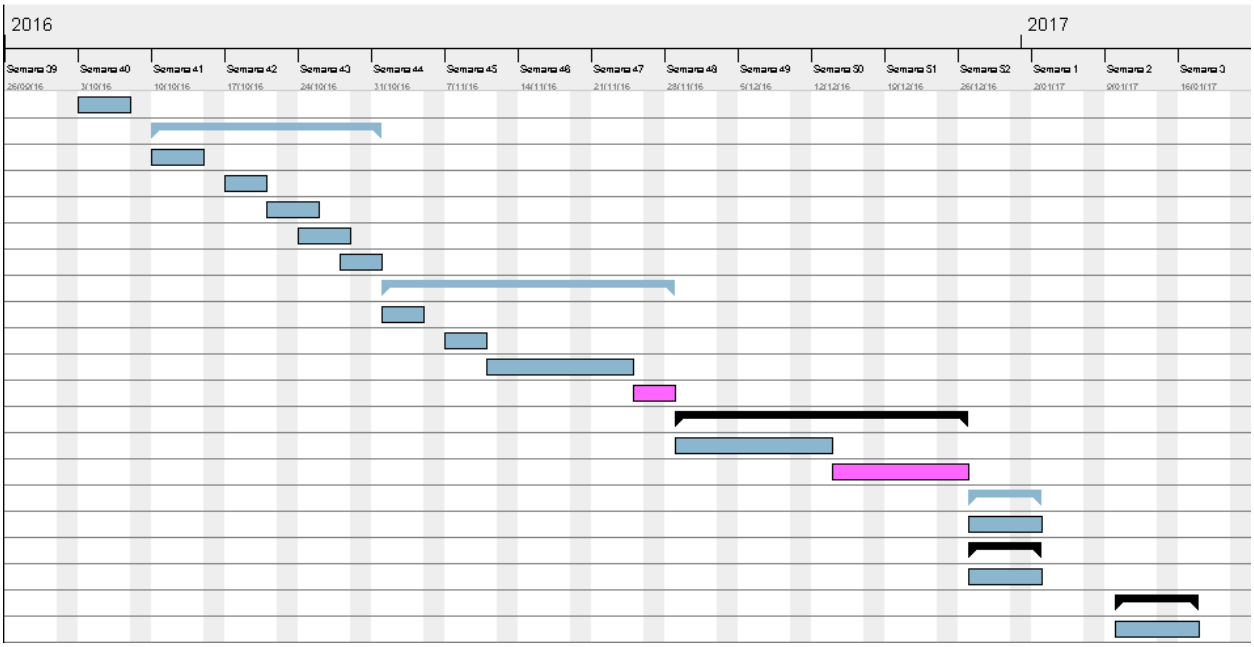


Figura 1: Diagrama de Gantt

[2]https://git-scm.com/
[3]https://www.dropbox.com/

Hitos y Tareas	Inicio	Fin	Horas	%
Plan de trabajo	03/10/16	09/10/16	8	2 %
Desarrollo del trabajo Fase 1	10/10/16	31/10/16	108	30 %
Búsqueda bibliográfica	10/10/16	14/10/16	30	8 %
Diseño de la estructura del paquete	17/10/16	20/10/16	24	7 %
Diseño de la plantilla del documento automático	21/10/16	25/10/16	18	5 %
Implementación de las funciones en R (I)	24/10/16	28/10/16	30	88 %
Documentar las funciones de R	28/10/16	31/10/16	6	2 %
Desarrollo del trabajo Fase 2	01/11/16	28/11/16	102	28 %
Creación de la plantilla del documento automático	01/11/16	04/11/16	24	7 %
Diseño de la aplicación en Shiny	07/11/16	10/11/16	18	5 %
Implementación de la aplicación en Shiny	11/11/16	24/11/16	60	17 %
Memoria del trabajo final	25/11/16	28/11/16	90	25 %
Redactar la memoria	29/11/16	26/12/16	66	18 %
Tiempo de reserva	29/11/16	13/12/16	24	6 %
Presentación del trabajo	14/12/16	26/12/16	30	8 %
Diseñar la presentación	27/12/16	02/01/17	30	8 %
Autoevaluación del trabajo	27/12/16	02/01/17	3	1 %
Completar el cuestionario de autoevaluación	27/12/16	02/01/17	3	1 %
Defensa pública	10/01/17	17/01/17	30	8 %
Preparar la exposición oral	10/01/17	17/01/17	30	8 %

Tabla 1: Temporización prevista.

Existen algunos factores de riesgo que pueden afectar a la temporización de etapas. Por esa razón se reservan algunas horas "Tiempo de reserva" en la tabla 1 para poder corregir las posibles desviaciones en los tiempos de entrega.

1.5. Resultados esperados

- Plan de trabajo
- Memoria
- Producto: Un paquete de R, un Markdown dinámico y una aplicación Shiny.
- Presentación virtual
- Autoevaluación del proyecto

1.6. Estructura del proyecto

A continuación se listan las partes que ha de contener la memoria final

- Resumen / Abstract
- Introducción
- Metodología
  - Bootstrap
  - Combn
- El paquete `testratio`
  - funcion `testratio`
  - function `testratio_report()`
  - la aplicación `testrapptio()`
  - Ejemplo de uso
- Ejemplo de uso del paquete `testratio` con datos reales
- Análisis de los tiempos de ejecución

- Conclusiones

2. Metodología

La técnica de la reacción en cadena de la polimerasa en tiempo real (RT-qPCR) es ampliamente utilizada en la investigación y el diagnóstico como método para cuantificar la cantidad de ácido nucleico. Cuando se utiliza en combinación con retrotranscripción (RT-qPCR) permite determinar la expresión relativa presente en una muestra biológica dada. Los resultados se basan en la expresión del gen candidato frente a una referencia (usualmente la expresión de otro gen) o una cuantificación absoluta basada en curvas de calibración internas o externas [5] por cada muestra biológica a analizar. RE es ampliamente utilizado por los investigadores, ya que evita las complicaciones de generar muestras de calibración y se mide como la relación entre la expresión de un gen objetivo y otro de referencia. Las directrices MIQE [7] para la publicación de los datos RT-qPCR sugiere que debe indicarse, en el análisis de datos, de forma clara los procedimientos y métodos estadísticos. Se han desarrollado métodos estadísticos para los análisis de datos RT-qPCR [7] - [12], pero la mayoría de estos han basado la comparación entre un tratamiento y un control a partir de tests que asumen normalidad de los datos [7] - [12]. Dicha asunción no debería ser común, ya que se evalúan cocientes (ratios entre el gen candidato y el de referencia en cada muestra biológica) de RE, los cuáles no pueden tomar valores negativos y presentan una distribución asimetrica . A pesar de ello, se ha comprobado que se aplica de forma común en varios estudios [11]. Se han sugerido alternativas a la estimación de cocientes basadas en aproximaciones asintóticas [3], pero su uso cuando se dispone de datos con un número reducido de observaciones (n<=20) puede no ser adecuado.

En este sentido, los métodos de remuestreo, como el descrito por BootstRatio [2] son una alternativa a la distribución de probabilidad de los cocientes.

NOTA: En este trabajo las expresiones Ratio y Expresión relativa serán utilizadas indistintamente.

2.1 Significación estadística

La función `testratio` permite obtener una estimación de la probabilidad de que el ratio calculado ($\frac{E(X_T)}{E(X_C)}$) sea superior o inferior a 1. Fijando una probabilidad de referencia (threshold) al que designamos como α es posible asumir que la RE entre el tratamiento y el control para un gen determinado es superior o inferior a 1 (1), valorando este resultado desde un punto de vista probabilístico. De esta manera al trabajar con probabilidades, el investigador puede valorar si el resultado es relevante o no, en lugar de determinar su significación estadística desde el punto de vista clásico.

$$\begin{aligned} 1 - P(RE > 1) \\ P(RE > 1) \end{aligned} \tag{1}$$

Si alguno de los valores obtenidos es superior al threshold (α) prefijado se considera que el resultados es relevante para el investigador.

Dado un vector $\hat{R}_G$ que contiene las medias de las RE para un gen G obtenidas mediante remuestro es posible estimar $P(RE > 1)$ mediante su aproximación a $P(\mu_{R_G>1})$ calculada como (2)

$$P(\hat{R}_G > 1) = \frac{\sum_{j=1}^{L} I(\hat{R}_G^j > 1)}{L} \text{ dónde } I(\hat{R}_G^j > 1) = \begin{cases} 1 & \text{si } \hat{R}_G^j > 1 \\ 0 & \text{si } \hat{R}_G^j \leq 1 \end{cases} \tag{2}$$

En el caso de que haya más de una tratamiento a evaluar respecto el mismo grupo control (es decir, en el mismo test, se evalúan distintos tratamientos siendo el grupo de referencia o control el mismo para todos ellos), el procedimiento que el investigador debería seguir sería comparar en primera instancia y de manera independiente, cada tratamiento con el grupo control. Ello permitiría determinar qué tratamientos se diferencían del control. En la mayoría de las situaciones estas comparaciones serán las únicas de interés, pero entendemos que en algunos casos también puede ser de interés comparar tratamientos (cuando estos hayan sido previamente significativos respecto el control) entre sí.

2.2. Ratios previamente calculados

Sea G un gen de interés para el cual disponemos de un conjunto de m expresiones a evaluar y sea $X_G = \{X_{G1}, \cdots, X_{Gm}\}$ el vector de ratios obtenidos de la división de las RE del grupo tratamiento entre las RE del grupo control assumiendo $X_{Gi} \geq 0 \quad \forall i = 1, \cdots, m$ y $m \geq 7$. Nos interesa contrastar las hipótesis (3). Dónde $E(R_G) = \mu_{R_G}$ por tanto es posible estimar $\hat{\mu}_{R_G} = \hat{R}_G = (1/m) \sum_{i=1}^{m} X_{Gi}$.

$$\begin{aligned} \mu_{R_G} &\leq 1 \\ \mu_{R_G} &> 1 \end{aligned} \tag{3}$$

Se generan L remuestras de longitud m con reposición del vector X_G. Para cada remuestra se calcula su media aritmética $\hat{R}_G^j \ \forall j = 1, \cdots, L$. Una vez obtenidas las L medias se estima $P(\mu_{R_G} > 1)$ tal como se describe en (2).

2.3. Datos separados por grupo Tratamiento y Control

Dados dos vectores $X_{G,C} = \{X_{G,C1}, \cdots, X_{G,Cm}\}$ y $X_{G,T} = \{X_{G,T1}, \cdots, X_{G,Tm}\}$, con las m expresiones obtenidas en los grupos control y tratamiento respectivamente para el gen G de interés, se quiere contrastar (3). Se generan $2L$ remuestras (L del vector $X_{G,C}$ y L del vector $X_{G,T}$) y se calcula , para cada remuestra, el estadístico de interés (en este caso la media aritmética) (4) siendo $X_{G,C}^j$ y $X_{G,T}^j \quad \forall j = 1, \cdots, L$ son las j-ésimas remuestras de los vectores $X_{G,C}$ y $X_{G,T}$ respectivamente, L el número de remuestras.

$$\begin{aligned} \hat{\mu}_C^j &= 1/m \sum_{i=1}^{m} X_{Ci}^j \quad \forall j = 1, \cdots, L \\ \hat{\mu}_T^j &= 1/m \sum_{i=1}^{m} X_{Ti}^j \quad \forall j = 1, \cdots, L \end{aligned} \tag{4}$$

Obtenidas las $2L$ medias se calculan los L ratios (5) y se calcula la significación (1) mediante (2).

$$\hat{R}_G^j = \frac{\hat{\mu}_{G,T}^j}{\hat{\mu}_{G,C}^j} \quad \forall j = 1, \cdots, L \tag{5}$$

2.4 Métodos de remuestreo implementados

En la aplicación `testratio` se han implementado dos métodos distintos para los casos en los que disponemos de los datos separados por grupo control y tratamiento. El método *Bootstrap* que genera las L remuestras con reposición y el método al que hemos llamado *Combn* que genera todas las posibles remuestras con reposición de los vectores X_C y X_T y las compara entre ellas. Este último método tiene un coste computacional más elevado y por tanto su uso es recomendable en aquellos casos en los que disponemos de un número reducido de observaciones por grupo.

Según el tipo de datos introducidos y el método seleccionado las remuestras se generan de una forma diferente. Se presenta un ejemplo para cada una de las posibles opciones.

Ejemplo 1: Método Bootstrap con RE previamente calculadas

Supongamos que se dispone de un vector $X_G = \{1, 2, 3, 4, 5, 6, 7\}$ que contiene las 7 observaciones de RE obtenidas para un gen particular. Si se desea estimar la distribución *Bootstrap* de dicho vector, se generan L remuestras del mismo tamaño. Para el ejemplo supongamos una $L = 2$ las remuestras obtenidas podrían ser las que se presentan en (6).

$$
\begin{aligned}
X^1 &= \{7,7,7,6,6,2,6\} \\
X^2 &= \{3,3,4,1,1,4,7\}
\end{aligned}
\tag{6}
$$

Ejemplo 2: Método Bootstrap con grupo tratamiento y control

Supongamos que se dispone de los vectores $X_T = \{1,2,3,4,5,6,7\}$ y $X_C = \{8,9,10,11,12,13,14\}$. Supongamos una primera remuestra $X_T^1 = \{1,2,3,4,4,7,7\}$ y $X_C^1 = \{8,8,8,9,9,9,13\}$, para las cuáles se calcula la media de X_T^1, $\bar{X_T^1} = 4$, y la media de X_C^1, $\bar{X}_C^1 = 9{,}14$, y se estima un primer cociente $\hat{R}^1 = 4/9{,}14 = 0{,}437$. Se lleva a cabo un segundo remuestreo obteniendo $\hat{R}^2$, y un tercero $\hat{R}^3$, y así sucesivamente hasta un máximo de L. Se sugiere un máximo de L=9999 para obtener una distribución Bootstrap de las $\hat{R}$

Ejemplo 3: Método Combn con grupo tratamiento y control

Supongamos que se dispone de los vectores $X_T = \{1,2,3,4\}$ i $Xc = \{5,6,7,8\}$. El método *Combn* calcula las $2 \cdot 4^4$ posibles remuestras con reposición, en el ejemplo aquí propuesto serían 256 remuestras con los valores del vector X_T y 256 más con los valores del vector X_C.

$$
\begin{aligned}
X_T^1 &= \{1,1,1,1\} \qquad X_C^1 = \{5,5,5,5\} \\
X_T^2 &= \{2,1,1,1\} \qquad X_C^2 = \{6,5,5,5\} \\
&\vdots \qquad\qquad\qquad \vdots \\
X_T^{256} &= \{4,4,4,4\} \qquad X_C^{256} = \{8,8,8,8\}
\end{aligned}
\tag{7}
$$

Se calculan las 512 (256 + 256) medias de las remuestras: $\bar{X}_T^1 = 1$, $\bar{X}_T^2 = 1{,}25$, ..., $\bar{X}_T^{256} = 4$, $\bar{X}_C^1 = 5$, $\bar{X}_C^2 = 5{,}25$, ..., $\bar{X}_C^{256} = 8$. Obtenidas las medias se divide cada una de las 256 medias de T entre las 256 medias de C obteniendo un total de 65536 $(256 * 256)$ ratios.

Una vez obtenidas las remuestras en los tres casos anteriores se calcula el estadístico de interés tal como se describe en los apartados 2.2 y 2.3.

3. La librería `testratio`

La librería `testratio` esta pensada con una triple intención. Por un lado el conjunto de funciones implementadas en `R` para aquellos que esten familiarizados con el lenguaje de programación, por otro lado la función `testratio_report()` que permite al usuario generar el informe dinámico usando la consola y finalmente la aplicación `testrapptio` ,a la que se puede acceder mediante el enlace https://alvarofranquet.shinyapps.io/testrapptio/ o la función `teestrapptio()`.

3.1 Estructura y funciones

En el momento de plantear la estructura de la librería `testratio` el objetivo principal era crear una herramienta útil y sencilla para aquellos investigadores de áreas muy diversas que pueden no tener conocimientos del lenguaje R. Con esto en mente, se decidió implementar la estructura de la Figura 2.

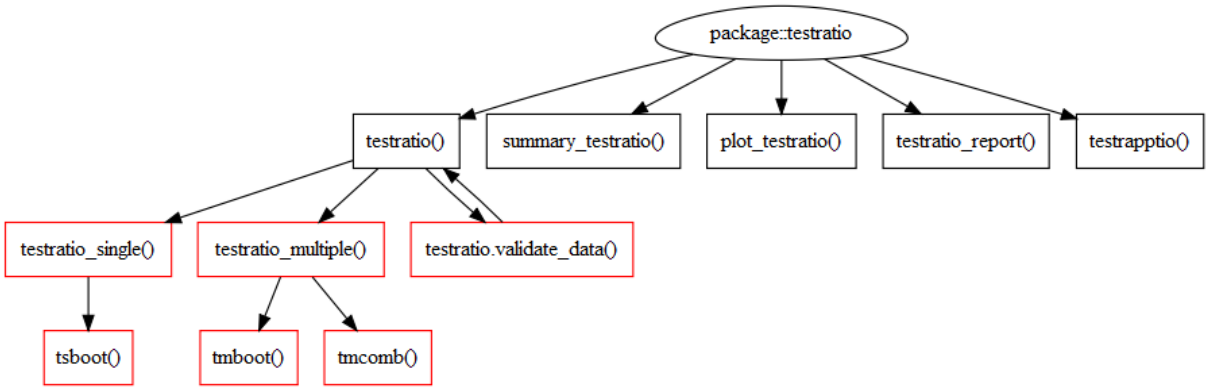


Figura 2: Diagrama con la estructura de la libreria `testratio`

La Figura 2 muestra algunas funciones de color rojo, estas funciones, que denotaremos cómo ocultas, son funciones que, aunque incluidas en el paquete, no pueden ser usadas por el usuario y no tienen documentación. La ocultación de estas funciones tiene como objetivo evitar que el usuario final se pierda en un mar de funciones. Por esa razón sólo cinco de las once se muestran al usuario, de estas las más importantes son `testratio()`, `testratio_report()` y `testrapptio()`.

- `testratio()`: se puede considerar el núcleo de la librería. Es la función que llevará a cabo el análisis estadístico de las expresiónes. A nivel interno `testratio()` ejecuta la función oculta del método seleccionado por el usuario. Esta estructura nos permite añadir en un futuro tantos métodos como queramos creando su respectiva función en cógido R. El parámetro `fun` permite al usuario contrastar cualquier estadístico o función propia.
- `testratio_single()`: Función auxiliar que selecciona, cuando disponemos de un único vector con los datos de las RE previamente cálculadas, el método indicado por el usuario.
- `testratio_multiple()`: Función auxiliar que selecciona, cuando disponemos de un grupo control y n tratamientos, el método seleccionado por el usuario.
- `tsboot()`: Lleva a cabo el test estadístico descrito en el apartado *2.2. Ratios previamente calculados* .
- `tmboot()`: Lleva a cabo el test estadístico descrito en el apartado *2.3. Datos separados por grupo* cuando se generan las remuestras mediante el método *Bootstrap*.

- `tmcomb()`: Lleva a cabo el test estadístico descrito en el apartado *2.3. Datos separados por grupo* cuando se generan las remuestras mediante el método *Combn*.
- `plot_testratio()`: Dibuja la función de densidad de las remuestras obtenidas.
- `summary_testratio()`: Retorna una tabla con los principales estadísticos descriptivos de interés sobre las remuestras generadas.
- `testratio_report()`: genera un documento con extensión .pdf dividido en tres apartados principales. El primero es un análisis descriptivo de los datos introducidos. En la actualidad se consideran grupos aptos para el análisis aquellos que tengan más de 7 observaciones no negativas. Aquellos conjuntos de expresiones que no cumplan estos requisitos no serán evaluadas. La segunda parte del documento es el análisis estadístico de la expresión relativa para cada gen. Es común utilizar ficheros de datos con un número muy elevado de genes, por esta razón la función testratio_report() permite no incluir en el informe aquellos que no alcancen una probabilidad fijada (α). La tercera y última parte del documento es un resumen de los resultados obtenidos.
- `testrapptio()`: abre una aplicación shiny en el navegador. Esta aplicación pretende ser una primera aproximación a la librería, sobre todo para aquellas personas no familiarizadas con el entorno R. La aplicación esta estructurada en tres pasos: Cargar los datos, seleccionar los parámetros a utilizar y generar el informe. Por defecto la aplicación nos generará una semilla que nos permitirá reproducir el análisis en el futuro.

A demás de las funciones descritas anteriormente se incluye un conjunto de datos (`testratio_example_data`) que ha sido proporcionados por el grupo investigador. Estos datos contienen las expresiones relativas obtenidas de los genes *arf1*, *odc1* y *s100a11* del riñon de ratones. Para cada gen se han analizado 25, 16 y 17 RE divididos en tres grupos: NP, P y control (CTRL).

Tabla 2: Seis primeras filas del conjunto de datos testratio_example_data incluido en el paquete testratio.

gene	group	value
arf1	CTRL	1.542
arf1	CTRL	0.904
arf1	CTRL	1.737
arf1	CTRL	0.878
arf1	CTRL	1.444
arf1	CTRL	0.786

Este *dataset* esta formado por 3 columnas y 144 filas. A continuación se describen las distintas variables:

- `gene`: Columna de tipo factor que indica a que gen pertenece cada valor.
- `group`: Columna de tipo factor que indica a que grupo pertenece cada valor.
- `value`: Columna con los valores obtenidos en el qPCR.

3.2 La función testratio()

La función `testratio()` es el núcleo de la librería. Esta función permite al usuario llevar a cabo el test para los ratios de las expresiones obtenidas. En este apartado se describe su funcionamiento y los distintos usos que permite la función en su estado actual, para lo que se usará un conjunto de datos simulado.

Se crean dos vectores de longitud 20 `x_c` y `x_t` que simulan los datos obtenidos para el grupo control y tratamiento respectivamente. El grupo control tendrá una media de 5 y el grupo tratamiento de 3, la varianza será de 1.5 igual para los dos grupos.

```
> set.seed(1479735781) # escogemos una semilla
> x_c = rnorm(n = 20, mean = 5, sd = 1.5) # grupo control
> x_t = rnorm(n = 20, mean = 3, sd = 1.5) # grupo tratamiento
```

3.2.1 Ejemplo de uso con ratios previamente calculados

Sea `s` el vector con los ratios calculados a partir de los vectors `x_t` y `x_c`.

```
> s <- x_t/x_c
```

Se carga la librería `testratio` y se computa el `testratio()` para el vector `s` mediante el método bootstrap, el número de remuestras a generar se puede modificar mediante el parámetro `m`. En este caso se generan `9999` remuestras.

```
> results <- testratio(value = s, group = 'default', gene = 'default',
+                      fun = 'mean', m = 9999)
```

Usando la función `str()` del paquete `utils` podemos ver la estructura del objeto `results` devuelo por la función `testratio()`. La función `testratio()` devuelve una lista de listas.

```
> str(results)
```

```
List of 1
 $ default:List of 1
  ..$ default:List of 2
  .. ..$ R     : num [1:9999] 0.927 0.898 0.751 0.773 0.858 ...
  .. ..$ method: chr "bootstrap"
```

Para cada uno de los distintos genes en el conjunto de datos obtenemos una lista con los resultados de cada comparación entre dos grupos y para cada comparación se dispone de un vector `R` que contiene las remuestras obtenidas y una variable de tipo carácter (`method`) con la descripción del método empleado para generar las remuestras.

- `R`: Un vector númerico con el valor del estadístico para cada una de las remuestras.
- `method`: Un valor de tipo caracter indicando el método utilizado para generar las remuestras.

Se han incluido dos funciones para poder interpretar los resultados obtenidos. La función `plot_testratio()` dibuja la distribución de las remuestras obtenidas y la funcion `summary_testratio()` imprime en pantalla una tabla con la estimación de $P(\mu_{R_G} > 1)$

```
> plot_testratio(results)
```

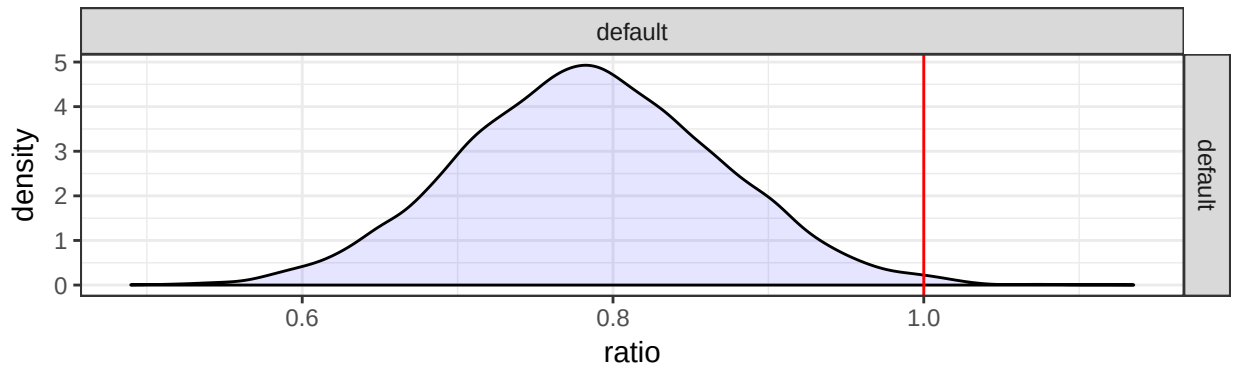


Figura 3: Gráfico de densidad obtenido mediante la función plot_testratio() en un ejemplo simulado de datos con los ratios previamente calculados.

```
> summary_testratio(results)
```

Tabla 3: Resumen numérico con los principales estadísticso descriptivos obtenidos en la simulación de datos con los ratios previamente calculados.

gene	group	mean	sd	min	Q1	Q2	Q3	max	P(A>B)	1-P(A>B)
default	default	0.7854	0.0828	0.4897	0.729	0.7842	0.8408	1.1351	0.0055	0.9945

3.2.2 Ejemplo de uso con grupos tratamiento y control

En este ejemplo se muestra como evaluar los ratios cuando tenemos los datos separados por grupo. La Tabla 4 muestra la estructura de los datos simulados.

```
> # creación del formato adecuado
> value = c(x_c,x_t)
> group = rep(c('ctrl','treatment'), each = 20)
> gene = rep('gene1', 40)
> md <- data.frame(value = value, group = group, gene = gene)
> knitr::kable(head(md), align = 'c', digits =4, caption ='Seis primeras filas
+ del data.frame simulado de datos con grupos tratamiento y control.\\label{tablaratios1}')
```

Tabla 4: Seis primeras filas del data.frame simulado de datos con grupos tratamiento y control.

value	group	gene
5.3714	ctrl	gene1
4.8490	ctrl	gene1
5.2832	ctrl	gene1
4.4214	ctrl	gene1
3.1426	ctrl	gene1
4.8296	ctrl	gene1

```
> # Remuestreo de los datos.
> results <- testratio(value = md$value, group = md$group, gene = md$gene,
+                      ctrl_name = 'ctrl', fun = 'mean', m = 9999)
```

```
> plot_testratio(results)
```

```
> summary_testratio(results)
```

Tabla 5: Resumen numérico de los principales estadísticos descriptivos obtenidos para la media en la simulación de datos con grupos tratamiento y control

gene	group	mean	sd	min	Q1	Q2	Q3	max	P(A>B)	1-P(A>B)
gene1	treatment	0.729	0.0776	0.445	0.6755	0.7281	0.7802	1.0708	2e-04	0.9998

Mediante la función `testratio()`, tal y como se ha comentado en el apartado 3.1, también es posible hacer un contraste sobre otros estadísticos como la desviación estándar, modificando el parámetro `fun` tal como se

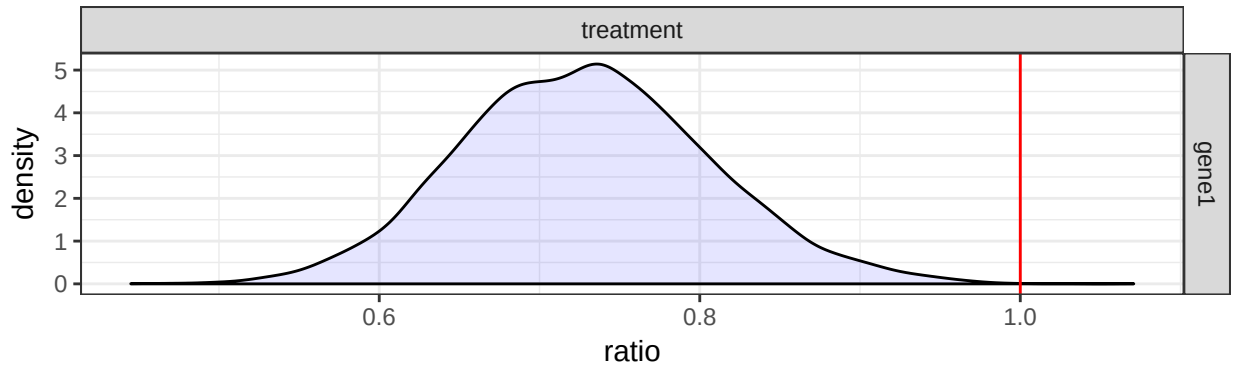


Figura 4: Gráfico de densidad de las remuestras obtenidas para la media en datos separados por grupo tratamiento y control

muestra a continuación:

```
> results <- testratio(value = value, group = group, gene = gene,
+                      ctrl_name = 'ctrl', fun = 'median', m = 9999)
```

```
> plot_testratio(results)
```

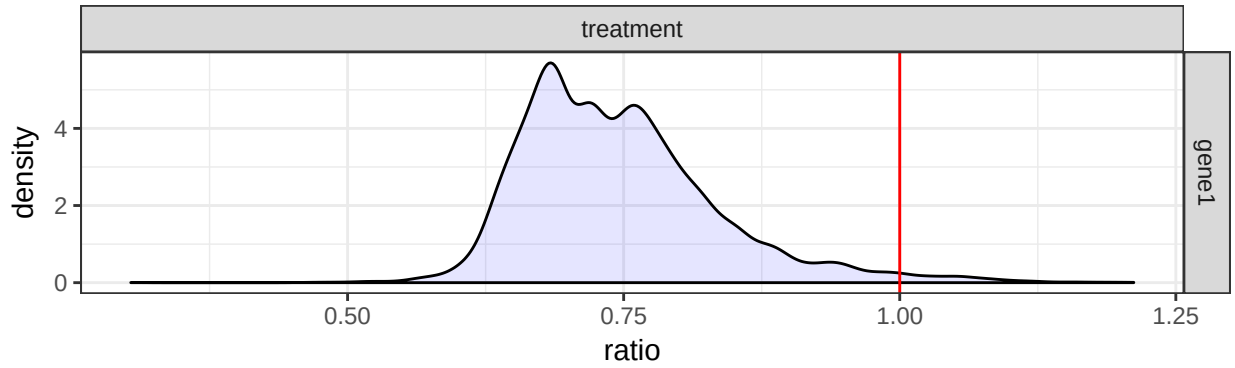


Figura 5: Gráfico de densidad de las remuestras obtenidas para la mediana en datos separados por grupo tratamiento y control

```
> summary_testratio(results)
```

Tabla 6: Resumen numérico de los resultados obtenidos para la mediana en la simulación de datos separados por grupo tratamiento y control.

gene	group	mean	sd	min	Q1	Q2	Q3	max	P(A>B)	1-P(A>B)
gene1	treatment	0.7465	0.0892	0.3037	0.6827	0.7327	0.7924	1.2119	0.0182	0.9818

3.3 La función testratio_report()

Una de las funcionalidades de la librería `testratio` es la posibilidad de generar un informe dinámico que permita de forma rápida y sencilla una primera visión sobre los resultados que vamos a obtener.

Este documento ha de ser de utilidad para las personas que componen el grupo de investigación y es por eso que en el desarrollo del documento ha sido necesaria la participación y revisión de todo el grupo para poder comprender mejor qué es exactamente aquello que se pretende realizar des de el punto de vista biológico y cómo se representa generalmente en el mundo de la medicina / biología.

Teniendo todo lo anterior en cuenta se ha decidio crear un documento con la siguiente estructura:

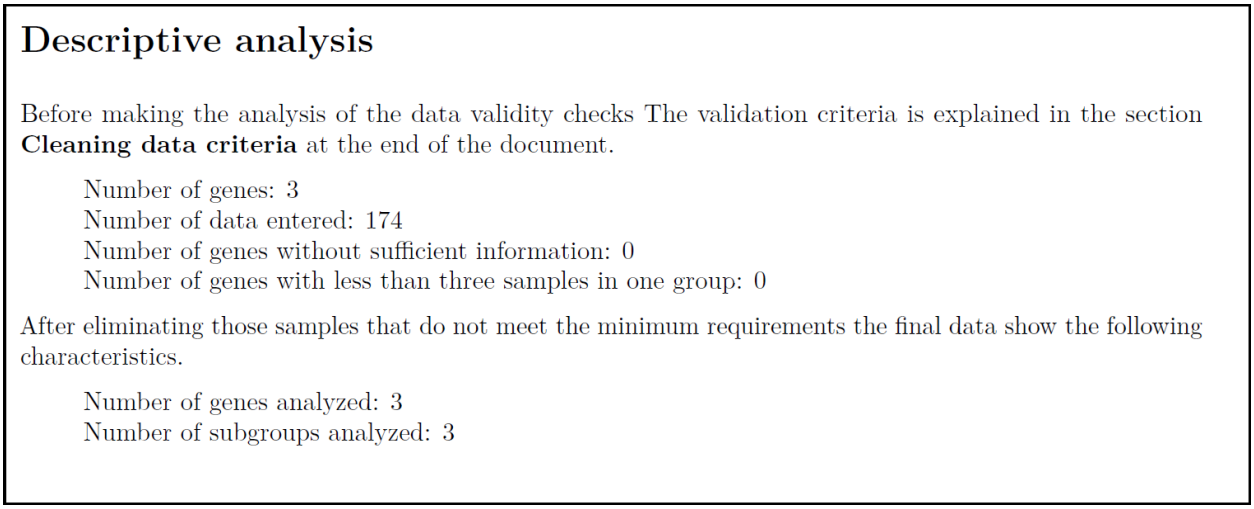
- Portada.
- Validación de los datos y análisis descriptivo del dataset.
- Estudio de significación de la expresión relativa.
- Resumen.
- Criterios de validación de los datos.

En los siguientes apartados se procede a una descripción más detallada de alguna de las partes. También se incluye un ejemplo del documento en los anexos.

3.3.1 validación de los datos y análisis descriptivo del dataset

En el documento se han aplicado algunos filtros a la hora de llevar a a cabo el estudio, estos filtros se han implementado para evitar usos inadecuados de la aplicación. En el mismo documento se presenta información sobre estas validaciones y al final del documento hay una pequeña descripción de los mismos.

La primera información (Figura 6) es **Number of genes** que nos informa de la cantidad total de genes que hay en el dataset, esta información ha de ser concocida previamente por el investigador y es una rápida evaluación de que los datos se han leído de forma correcta. **Number of data entered indica el número total de filas**. **Number of genes without sufficient information** indica el número de genes que han sido excluidos del estudio por falta de datos (se considera gen sin información necesaria aquel que tiene menos de siete observaciones). Finalmente **Number of genes with less than three samples in one group** indica el número de genes de los que hemos excluido uno de los grupos. En algunas ocasiones puede darse el caso de que sólo uno de los grupos no presente suficiente información pero el resto sí que se puede usar.



Descriptive analysis

Before making the analysis of the data validity checks The validation criteria is explained in the section **Cleaning data criteria** at the end of the document.

Number of genes: 3
Number of data entered: 174
Number of genes without sufficient information: 0
Number of genes with less than three samples in one group: 0

After eliminating those samples that do not meet the minimum requirements the final data show the following characteristics.

Number of genes analyzed: 3
Number of subgroups analyzed: 3

Figura 6: Validación de los datos y análisis descriptivo global del dataset obtenido en el informe de la funcion testratio_report().

Finalmente una vez filtrados los datos el documento nos informará sobre el número final de genes analizados y del número máximo de sub grupos analizados.

3.3.2 Estudio de significación de la expresión relativa.

Este apartado es el núcleo del documento, el lugar dónde se aplican los métodos descritos en este trabajo. Para cada gen se crea una página nueva que queda dividida en tres sub-apartados (Figura 7).

- Análisis descriptivo
- Gráficos de los ratios
- Tabla de resultados con: El valor de la media, cuantiles de las expresiones relativas obtenidas y estimación puntual de la probabilidad de $\mu_{R_G} > 1$.

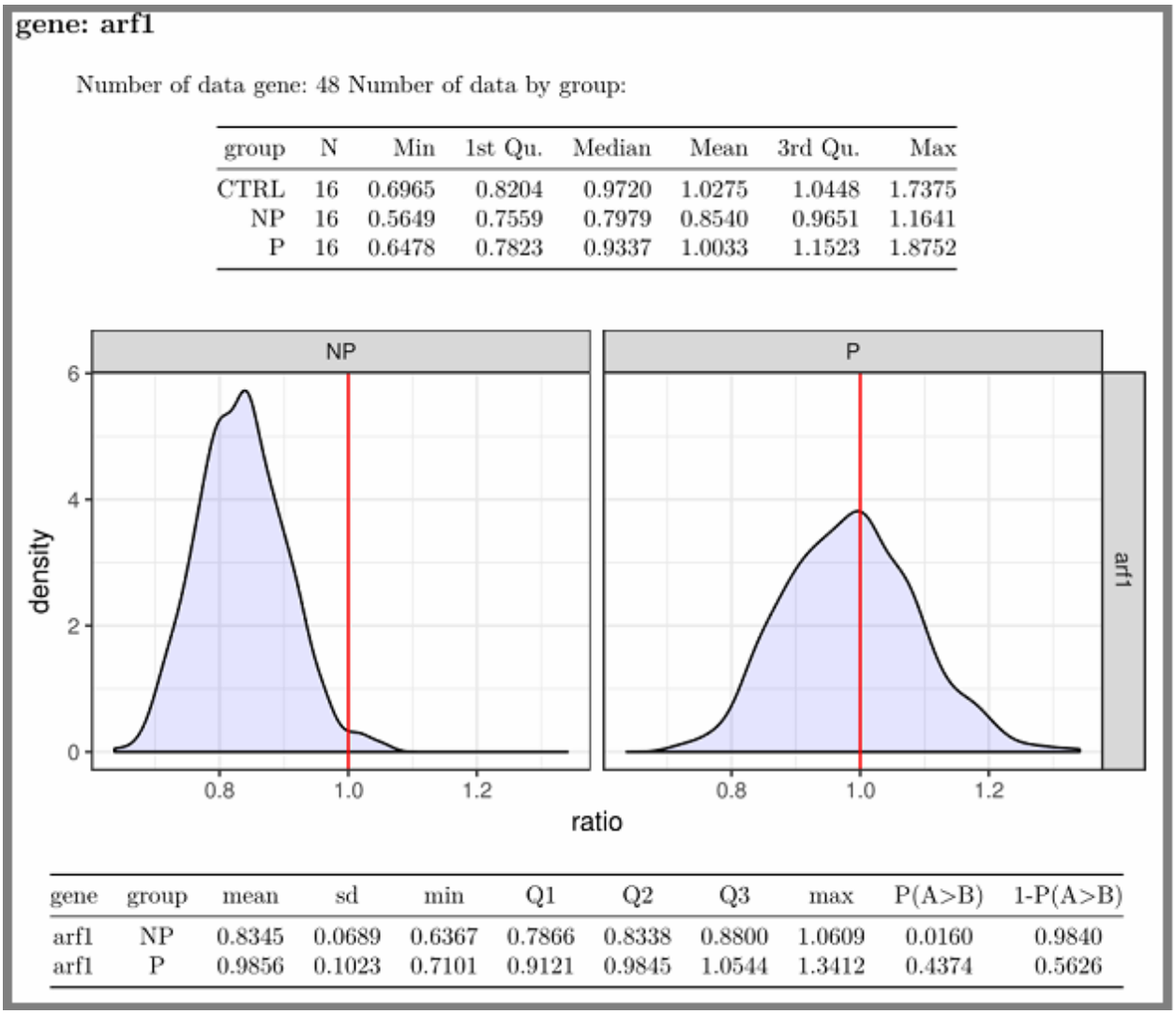


Figura 7: Ejemplo de un estudio de significación de la expresión relativa.

En ocasiones puede ser necesario evaluar un nómbre muy elevado de genes, por esa razón y para evitar la generación de un documento con un gran nombre de páginas, la mayoría de las cuales no van a ser relevantes para el estudio, se decide incluir la opción de eliminar todas aquellas que no sean inferiores a un nivel de significación α (Tal como se describe en (1)).

3.3.4 Resumen final

El resumen final cuenta con una tabla de todos los estadísticos descriptivos obtenidos a lo largo del documento. En esta tabla se muestran todos los genes, tanto si cumplen (1) cómo si no. El primer gráfico de barras muestra la estimación de los todos los ratios calculados y el segundo gráfico sólo aquellos genes con una RE estadísticamente relevante.

3.4 La función testrapptio()

La función `testrapptio()` permite iniciar la aplicación web `testrapptio` creada en *shiny*. Esta aplicación pretende simplificar el uso del paquete `testratio` para aquellas personas que no tienen porqué saber de `R`. La aplicación se compone de una interfaz de usuario dividida en tres pestañas con objetivos muy concretos. *File Upload* permite cargar el conjunto de datos, *Select Parameters* ayuda a definir los valores introducidos como parámetros en la función `testratio_report()` que se ejecuta en el servidor y *Get report* permite al usuario descargar el informe en formato PDF.

Esta nueva aplicación presenta algunas mejoras importantes respecto a la creada por Cleries et al. Una de las más relevantes es la unificación de las dos aplicaciones en una sola. Esto la hace más sencilla para el usuario que no tiene que acceder a una u otra en función de los datos de los que disponga. En la nueva versión también se genera una semilla que permite al investigador reproducir los mismos resultados en el futuro y la posibilidad de seleccionar el grupo control (que no tiene porqué llamarse 'CTRL').

A continuación se describen los menús y opciones de cada una de las pestañas junto a un ejemplo haciendo uso de los datos incluidos en paquete `testratio` y que han sido previamente guardados en formato TXT.

El primer paso es iniciar la aplicación, esto se puede hacer mediante la función `testrapptio()` o usando la versión en línea disponible en el siguiente enlace: https://alvarofranquet.shinyapps.io/testrapptio/. Al iniciar la aplicación se verá una pantalla como la que se muestra en la imagen Figura 8 dónde se puede ver un menú en el lateral izquierdo que permite al usuario ajustar diversos parámetros a la hora de introducir los datos. Esto permite más flexibilidad en el formato de los datos.

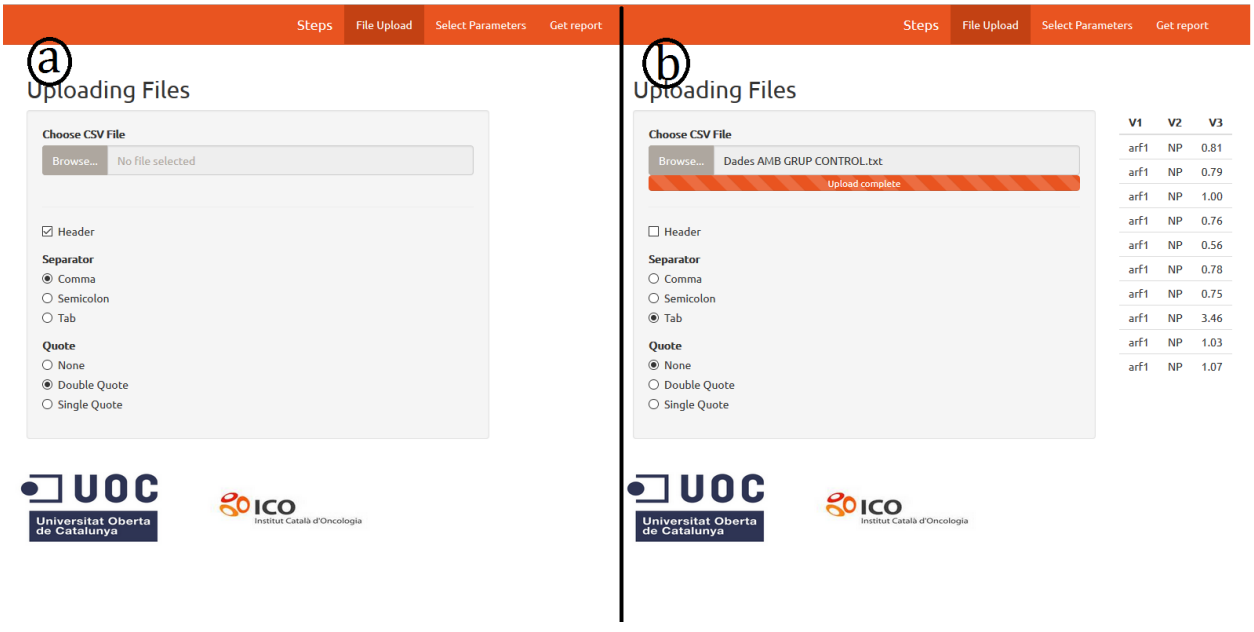


Figura 8: Interfaz de usuario de la aplicación testrapptio al ser iniciada (a) y la misma pestaña con los datos cargados (b)

Mediante el botón *Browse* podemos seleccionar el archivo deseado. El selector header indica si los datos introducidos tienen o no cabecera. El menú *Separator* permite seleccionar el tipo de separador utilizado (comas, semicolon o tabulador) y finalmente el submenú *Quote* que nos permite definir como están identificados los caracteres en el archivo. Una vez seleccionado el archivo se irá mostrando en pantalla como se han abierto los datos, de esta forma el usuario puede controlar en todo momento que se ha hecho de forma correcta. Si todo ha sido satisfactorio es el momento de cambiar a la pestaña Select Parameters (Figura 8).

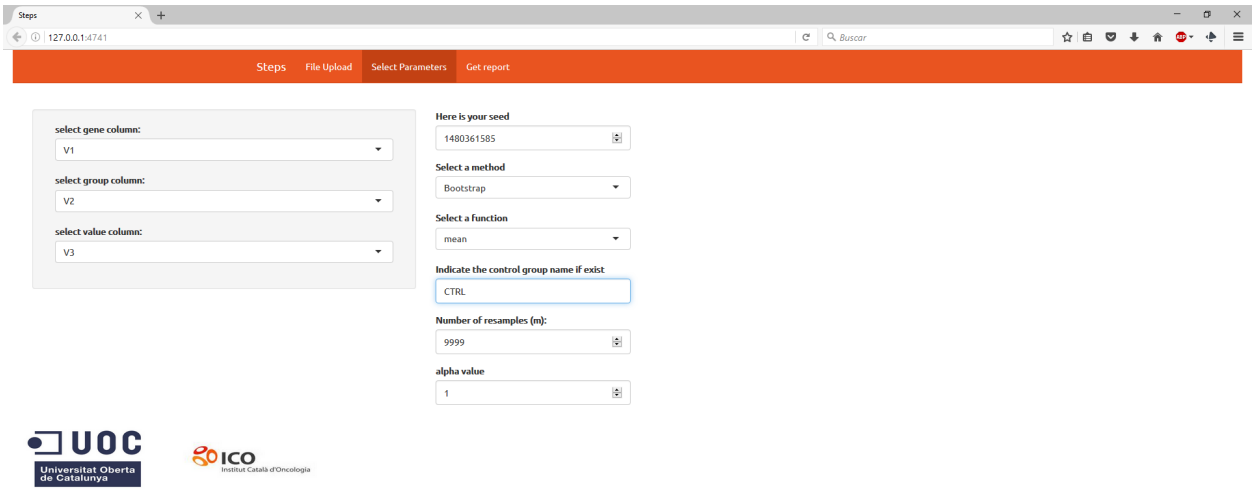


Figura 9: Interfaz de usuario de la pestaña Select Parameters

La pestaña Select Parameters permite modificar los parámetros de la función `testratio_report()`. En esta pestaña podemos ver dos menús. Uno en el lado lateral izquierdo con fondo gris y otro con el fondo blanco en el centro de la pantalla. El primero es un menú dinámico en el que se encuentran tres selectores dónde indicar las columnas de interés. El menú blanco contiene seis campos diferentes.

- **Here is your seed**: Internamente se genera una semilla que permite reproducir los mismos resultados en el futuro. Este campo indica qué semilla se ha generado y también permite usar una semilla diferente.
- **Select a method**: Este menú desplegable permite seleccionar qué método queremos usar para generar las remuestras (*Bootstrap* o *Combn*).
- **Select a function**: Este menú desplegable permite al usuario seleccionar qué estadístico quiere contrastar (en la actualidad *mean* o *median*).
- **Indicate the control group name, if exist.**: En este campo se ha de indicar el nombre del grupo control (*CTRL* en nuestro ejemplo).
- **Number of resamples (m)**: El número de remuestras que queremos generar.
- **Alpha value**: En este campo hay que indicar cual es el valor de α seleccionado, sólo es necesario si queremos que en el informe dinámico se muestren únicamente aquellos genes con expresiones relativas relevantes. Si este no es el caso se pone en 1 por defecto.

Finalmente, en la tercera pestaña *Get report* hay un botón que nos permite descargar el documento en format PDF. Basta con seleccionar para descargarlo (Figura 10).

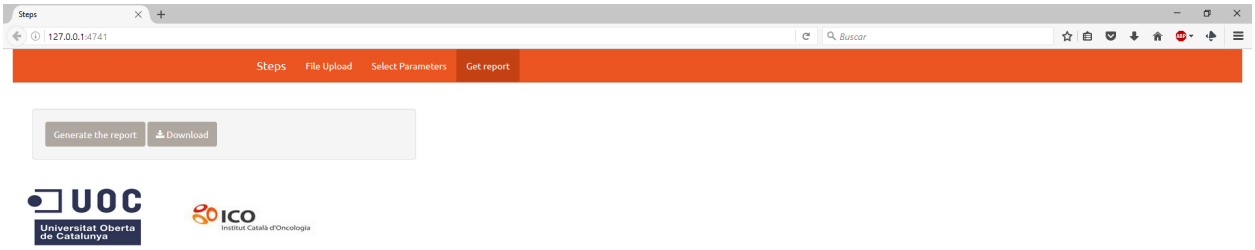


Figura 10: Interfaz de usuario de la pestaña Get report

3.5 Evaluación de los tiempos de ejecución

Este apartado tiene como objetivo evaluar los tiempos de ejecución de las función `testratio()`. Para ello se hace uso de los datos con las RE calculadas previamente. Se han seleccionado las RE obtenidas en pacientes con cáncer de próstata para el gen `12s/MT-RNR1` que se describen con más detalle en el apartado 4.1. El código `R` relativo a este subapartado puede encontrarse en el ANEXO B

Concretamente interesa evaluar las siguientes cuestiones:

- Qué efecto tiene en el tiempo de ejecución la longitud m del vector S_G?
- Qué efecto tiene en el tiempo de ejecución el número de genes análizados? denotaremos esta longitud como ng.
- Qué efecto tiene en el tiempo de ejecución el número de remuestras (L)?

Para responder a estas cuestiones se realizarán simulaciones con las siguientes características. Laslongitudes del vector S_G evaluadas son 7, 13 y 19, Para el número de genes se crea un vector con los valores de 1 a 1000 en intervalos de 100 genes de diferencia. Finalmente para estimar el efecto de incrementar el número de remuestras generadas se seleccionan los siguientes valores $L = 1000, 2000, 3000$.

Tabla 7: Seis primeras filas del conjunto de datos para los tiempos de ejecución de la función testratio()

m	ng	L	iter	user	system	elapsed
7	1	1000	1	0.03	0	0.02
7	1	1000	2	0.02	0	0.01
7	1	1000	3	0.00	0	0.00
7	1	1000	4	0.00	0	0.00
7	1	1000	5	0.01	0	0.01
7	1	1000	6	0.00	0	0.00

La Figura 11 muestra el gráfico con los resultados obtenidos para las simulaciones, en el eje X el número de genes y en el eje Y el tiempo de ejecución. En la parte superior del gráfico se indica la longitud del vector con los m Ratios (7, 13 o 19) y en la parte derecha del gráfico la cantidad de remuestras hechas (1000, 2000 o 3000).

Se puede observar un cambio en la pendiete según el valor m. Cuanto más grande sea m más rápidamente aumentará el tiempo de ejecución. Otro de los factores que parece ser relevante es el número de genes analizados. Sin embargo el tiempo de ejecucción más alto es de 27.42 segundos cuando analizamos una cantidad de 1000 genes con 19 observaciones.

Estos resultados son altamente satisfactorios sin embargo, es conveniente evaluar métodos de computación en paralelo cómo trabajo a futuro para intentar mejorar la *performance* del paquete `testratio`.

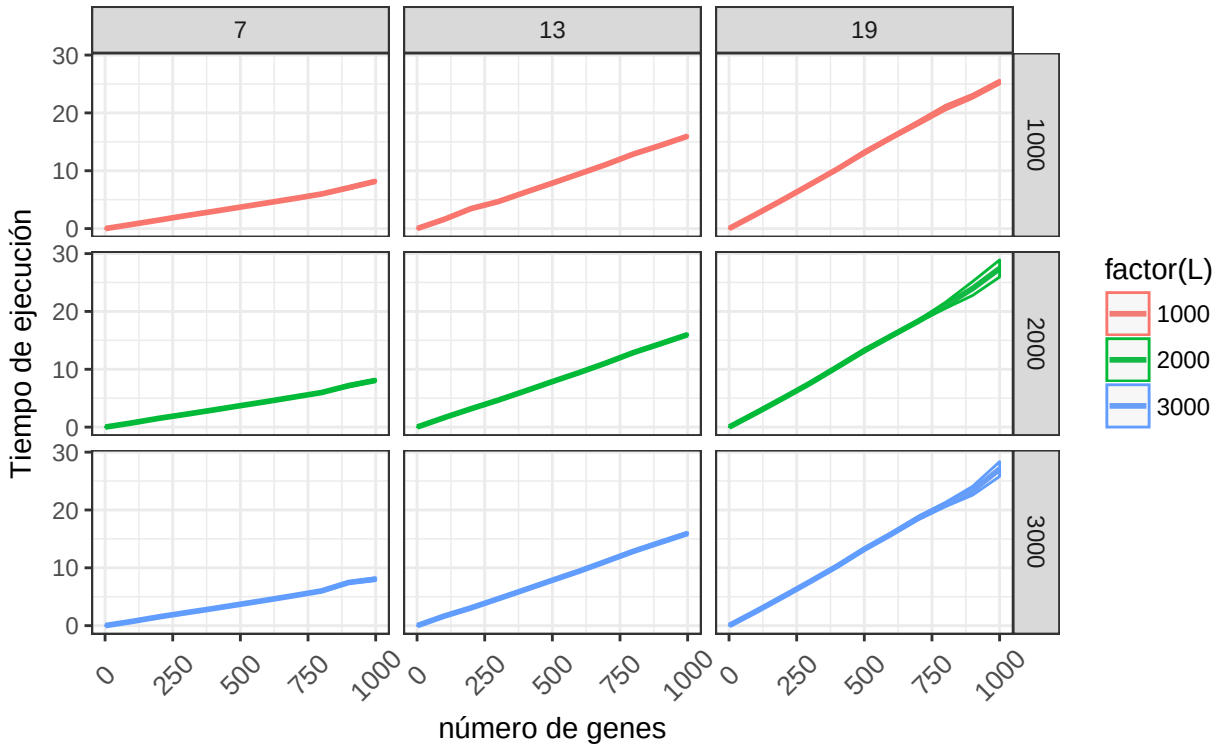


Figura 11: Tiempos de ejecución de la función testratio() separados por número de remuestras y tamaño del vector a evaluar.

4. Caso real

En este apartado se llevan a cabo los análisis de las expresiones para dos conjuntos de datos reales mediante las funciones implementadas en el paquete `testratio`.

4.1 Datos con los ratios previamente calculados

Se han recogido las expresiones relativas de los genes `12S/MT-RNR1`, `MT-CO2/COX2`, y `MT-ATP6` en el cáncer de próstata. Los Ratios corresponden a las expresiones relativas calculadas mediante dos muestras del mismo paciente una de la zona afectada por el cáncer (objetivo) y la otra a zonas no afectadas por este (control). En total se dispone de 19 ratios para cada gen.

Tabla 8: Primeras seis filas de los datos con los ratios calculados obtenidos de pacientes con cáncer de próstata

gene	value
12s/MT-RNR1	0.74475
12s/MT-RNR1	0.84080
12s/MT-RNR1	1.44750
12s/MT-RNR1	1.15882
12s/MT-RNR1	0.99075
12s/MT-RNR1	0.47821

Como se puede ver en la tabla 8 los datos contienen dos columnas: `gene` y `value`. La columna `gene` es una columna de tipo carácter con el nombre del gen al cual pertenece la observación. La columna `value` contiene el valor del Ratio previamente calculado. La Figura 12 muestra el gráfico de violín y boxplot para cada uno de los genes en el conjunto de datos.

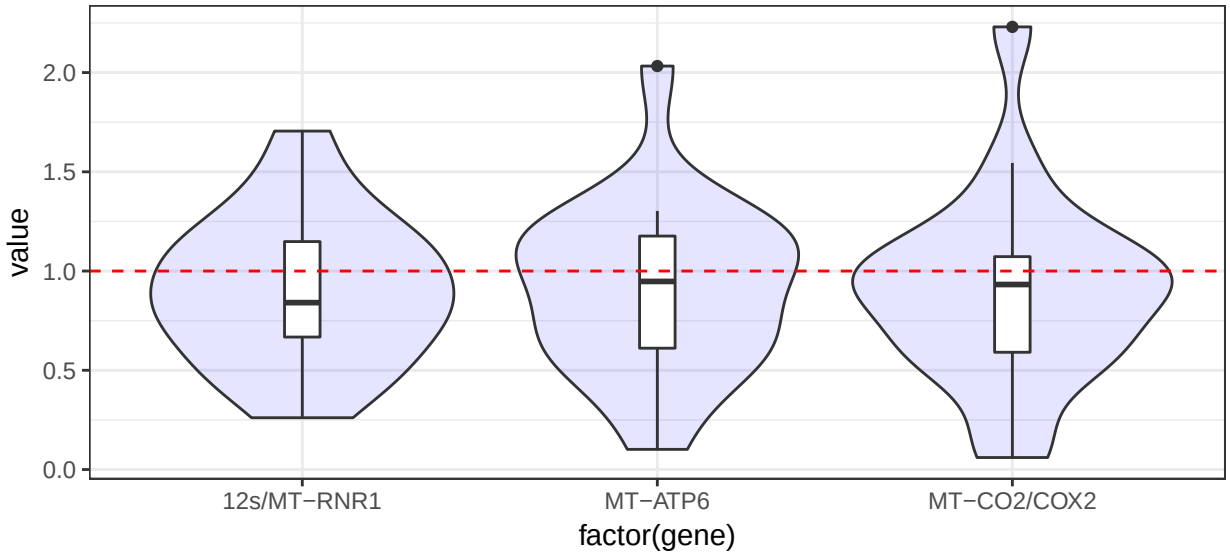


Figura 12: Gráfico de violín y boxplot por gen

La Figura 12 muestra el gráfico de violín [14] de los ratios para cada uno de los genes en el conjunto de datos. Este gráfico combina las características de un diagrama de caja (en el centro) con un gráfico de densidad (area coloreada). El gráfico de violín permite al usuario identificar la mediana y cuantiles (visibles en el

diagrama de caja) y detectar, por ejemplo, si los datos son bimodales. En el ejemplo aquí expuesto no se detecta bimodalidad pero si dos posibles datos atípicos en los genes *MT-ATP6* y *MT-CO2/COX2*.

Se procede a estimar la $1 - P(\mu_{R_G} > 1)$ mediante la funcion `testratio()`.

```
> set.seed(09051995)
> resultados <- testratio(value = md$value, gene = md$gene, ctrl_name = NULL, m = 9999,
+                         fun = 'mean', method = 'boot')
```

La Figura 13 muestra el gráfico de densidad de las remuestras por gen obtenido mediante la función `plot_testratio()`.

```
> plot_testratio(resultados)
```

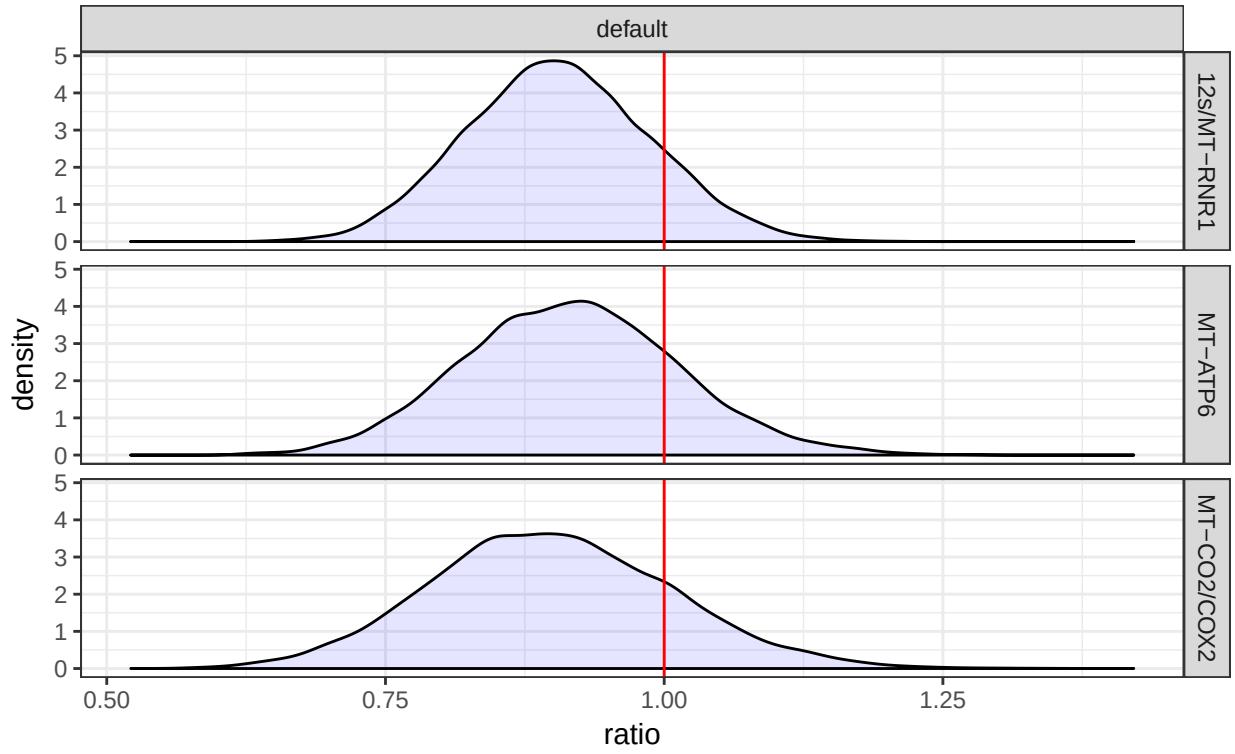


Figura 13: Gráfico de densidades de probabilidad de los ratios obtenidos mediante la función testratio()

Tabla 9: Resumen numérico de los resultados obtenidos mediante la función testratio en datos con los ratios previamente calculados

gene	group	mean	sd	min	Q1	Q2	Q3	max	P(A>B)	1-P(A>B)
12s/MT-RNR1	default	0.905	0.082	0.637	0.849	0.903	0.959	1.213	0.130	0.870
MT-ATP6	default	0.916	0.096	0.617	0.851	0.915	0.979	1.283	0.189	0.811
MT-CO2/COX2	default	0.899	0.108	0.522	0.825	0.895	0.970	1.421	0.175	0.825

En la Tabla 9 se muestra el resumen numérico de las remuestras obtenidas para los tres genes. La última columna muestra, en todos los casos, una probabilidad superior a 0,05 que es el valor límite de referencia i por tanto no se puede asumir que la expresión entre el gen y la referencia sea superior a 1 de forma estadísticamente relevante para los genes analizados en las zonas afectadas por el cáncer de próstata.

4.2 Datos con grupo tratamiento y control

En el siguiente ejemplo se hace uso de los datos incluidos en el paquete `testratio` a los que es posible acceder mediante el comando `data(testratio_example_data)`. Los datos han sido recogidos de diversas muestras de ARN de riñones de ratón usando el método de trizol (Invitrogen). El diseño experimental incluye tres grupos distintos de ratones: con Litiasis (Presencia de calcio, P), sin Litiasis (NP) y ratones salvajes cómo control (CTRL). Para cada grupo se obtubo un total de 16 muestras. La pureza y concentración de los datos fueron evaluados utilizando un espectrofotómetro (NanoDrop). La cantidad de RNA fué calculada mediante el método $2^{-\Delta\Delta Ct}$ [15]. Los genes analizados son: *arf1*, *odc1* y *s100a11*.

La Tabla 10 muestra las seis primeras filas de este conjunto de datos.

```
> data(testratio_example_data)
> md <- testratio_example_data
```

Tabla 10: Primeras seis filas de los datos obtenidos de ratones con y sin Litiasis

gene	group	value
arf1	CTRL	1.5418500
arf1	CTRL	0.9040843
arf1	CTRL	1.7374825
arf1	CTRL	0.8784755
arf1	CTRL	1.4444899
arf1	CTRL	0.7861940

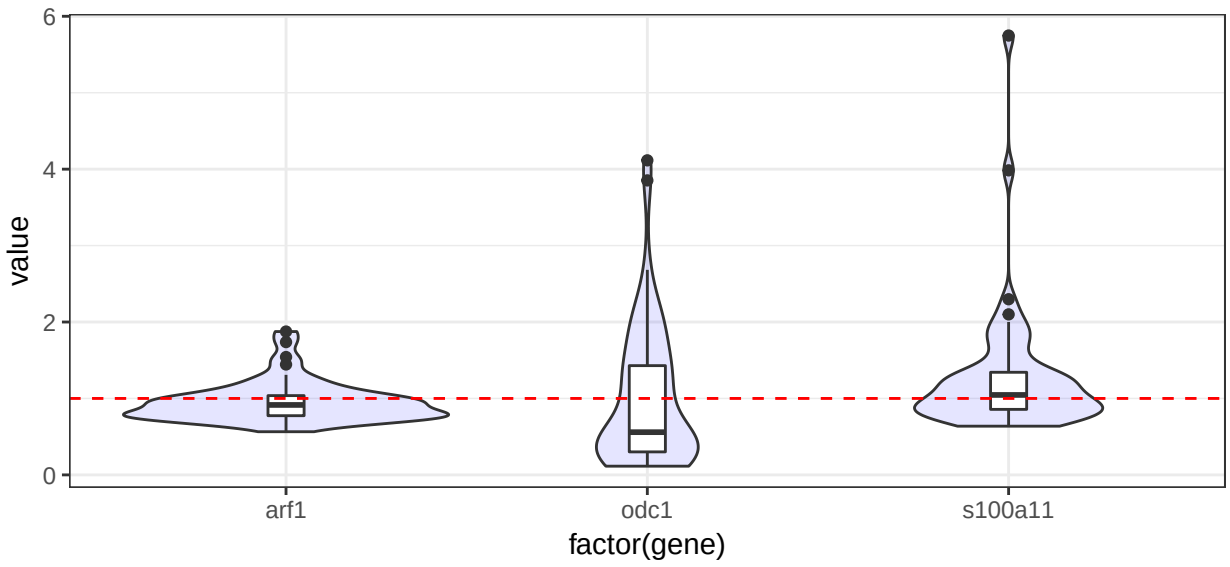


Figura 14: Gráfico de violín boxplot por gen datos Litiasis

```
> set.seed(09051995)
> resultados <- testratio(value = md$value, group = md$group, gene = md$gene,
+                         ctrl_name = 'CTRL', m = 9999, fun = 'mean',
+                         method = 'boot')
```

```
> plot_testratio(resultados)
```

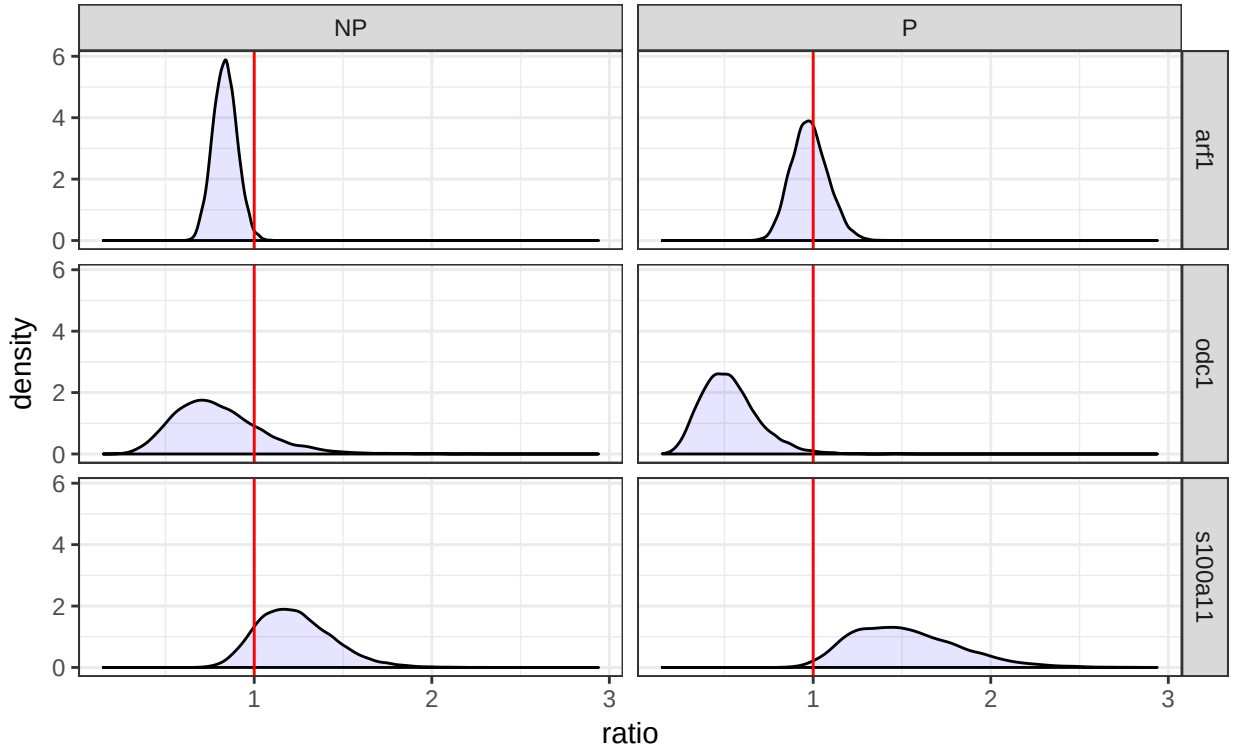


Figura 15: Gráfico de densidades para los ratios obtenidos mediante la función testratio()

Tabla 11: Resumen numérico de los resultados obtenidos mediante la función testratio en datos con los ratios previamente calculados

gene	group	mean	sd	min	Q1	Q2	Q3	max	P(A>B)	1-P(A>B)
arf1	NP	0.836	0.067	0.617	0.789	0.835	0.880	1.095	0.009	0.991
arf1	P	0.531	0.161	0.150	0.417	0.513	0.622	1.548	0.010	0.990
odc1	NP	1.231	0.211	0.684	1.077	1.211	1.360	2.290	0.874	0.126
odc1	P	0.980	0.102	0.645	0.909	0.976	1.047	1.406	0.406	0.594
s100a11	NP	0.797	0.249	0.246	0.620	0.763	0.937	2.257	0.187	0.813
s100a11	P	1.523	0.299	0.798	1.293	1.487	1.712	2.937	0.990	0.010

La Tabla 11 muestra los resultados obtenidos de analizar las remuestras generadas. En la columna 1-P(A>B) podemos ver un valor inferior a 0,05 para el contraste en el gen *s100a11* entre el grupo *P* y el grupo *CTRL*. Lo que nos permite asumir que la expresión entre el gen *s100a11* y la referencia es superior a 1 de forma relevante en el grupo con Litiasis. También se puede ver un valor inferior a 0,05 en la columna P(A>B) para los dos grupos del gen *arf1*. Esto se puede interpretar como que la expresión entre los grupos tratamientos y el grupo *CTRL* es menor a 1 de forma relevante lo que se traduce cómo que los grupos *P* y *NP* se expresan de forma más moderada que en el grupo *CTRL*.

5. Conclusiones

En lo referente a las competencias adquiridas durante la ejecución de este trabajo, es destacable la mejora en la capacidad para comprender conceptos propios de la Bioinformática y su posterior difusión, para ello ha sido muy necesaria la constante revisión y aportación de sugerencias de los tutores del trabajo que han sabido guiarme y centrarme en las cuestiones más relevantes del trabajo.

Se han alcanzado todos los objetivos propuestos desde un inicio, se ha creado el paquete en R y la aplicación en *Shiny* para que los futuros usuarios tengan una forma más atractiva de acercarse al funcionamiento del paquete y a los métodos que en este trabajo se proponen. Sin embargo el factor tiempo no ha permitido alcanzar en el grado deseado alguno de los objetivos y dejando algunas tareas en el tintero que se describen a continuación:

- Evaluación y implementación de métodos para el control de calidad de los datos así como detección de datos atípicos y su tratamiento en el análisis posterior.
- El análisis de los tiempos de ejecución muestra un aumento importante de estos cuando se lleva a cabo un estudio sobre un alto número de genes, por esa razón se plantea la posibilidad de incluir métodos de computación en paralelo o la posibilidad de usar mátrices de permutaciones previamente generadas por el usuario.
- Con la intención de llegar al máximo número de usuarios potenciales es necesaria la creación de un manual de uso online con ejemplo prácticos y incluirlo en la misma aplicación.
- Creación de un conjunto de herramientas que permitan analizar los usos y abusos de la aplicación para su posterior corrección.

Aún así los resultados obtenidos han sido, a nivel personal, muy gratificantes y el desarrollo de este trabajo me deja con ganas de seguir adentrandome en un mundo tan interesante como es el de la Bioinformática.

6. Glosario

- **Paquete**: Conjunto de funciones implementadas en un lenguaje de programación que están ordenadas y documentadas en un formato determinado. En el caso de R la estructura que han de seguir estos se encontrar en http://r-pkgs.had.co.nz/package.html
- **Ratio**: Relación entre dos expresiones. La palabra Ratio y Expresión relativa son usadas indistintamentes en este trabajo.

Referencias

[1] Jacobs, J. L. and Dinman, J. D. (2004). Systematic analysis of bicistronic reporter assay data. Nucleic Acids Research, 32(20), e160. http://doi.org/10.1093/nar/gnh157

[2] Clèries R, Galvez J, Espino M, Ribes J, Nunes V, de Heredia ML. BootstRatio: A web-based statistical analysis of fold-change in qPCR and RT-qPCR data using resampling methods. Comput Biol Med. 2012 Apr;42(4):438-45

[3] Phillip I.,Resampling Methods A practical Guide to Data Analysis, Birkhäuser, Boston-Basel-Berlin, 3rd edition, 2006.

[4] Hadley Wickham and Winston Chang (2016). devtools: Tools to Make Developing R Packages Easier. R package version 1.12.0. https://CRAN.R-project.org/package=devtools

[5] M.W. Pfaffl, M. Hageleit, Validities of mRNA quantification using recombinant RNA and recombinant DNA external calibration curves in real-time RT-PCR, Biotechnol. Lett. 23 (2001) 275-282.

[6] S.A. Bustin, Why the need for qPCR publication guidelines?-The case for MIQE, Methods 50 (2010) 217-226.

[7] M.W. Pfaffl, A new mathematical model for relative quantification in realtime RT-PCR, Nucleic Acids Res. 29 (9) (2001) e45.

[8] P.Y. Muller, H. Janovjak, A.R. Miserez, Z. Dobbie, Processing of gene expression data generated by quantitative real-time RT-PCR, Biotechniques 32 (2002) 1372-1379.

[9] M.W. Pfaffl, G.W. Horgan, L. Dempfle, Relative expression software tool (REST) for group-wise comparison and statistical analysis of relative expression results in real-time PCR, Nucleic Acids Res. 30 (2002) e36.

[10] S.N. Peirson, J.N. Butler, R.G. Foster, Experimental validation of novel and conventional approaches to quantitative real-time PCR data analysis, Nucleic Acids Res. 31 (2003) e73.

[11] R. Gilsbach, M. Kouta, H. Bonisch, M. Bruss, Comparison of in vitro and in vivo reference genes for internal standardization of real-time PCR data, Biotechniques 40 (2006) 173-177.

[12] J.S. Yuan, A. Reed, F. Chen, C.N. Stewart Jr, Statistical analysis of real-time PCR data, BMC Bioinformatics 7 (2006) 85.

[13] Jacobs J., Dinman, JD. Systematic analysis of bicistronic reporter assay data. Nucleic Acids Research, 2004, Vol. 32, No. 20

[14] Hintze, J. L., Nelson, R. D., Violin Plots: A Box Plot-Density Trace Synergism. The American Statistician 52 (1998) 181-184.

[15] K.J. Livak, T.D. Schmittgen, Analysis of relative gene expression data using real-time quantitative PCR and the 2(-delta delta C(T)) method, Methods. 25 (2001) 402-408.

[16] J. Felsenstein, Confidence limits on phylogenies: an approach using the bootstrap, Evolution 39 (1985) 783-791.

[17] S.N. Peirson, J.N. Butler, R.G. Foster, Experimental validation of novel and conventional approaches to quantitative real-time PCR data analysis, Nucleic Acids Res. 31 (2003) e73.

[18] Hadley Wickham, ggplot2: Elegant Graphics for Data Analysis, Springer-Verlag New York, (2009), http://ggplot2.org

[19] J. Abril, M.L. de Heredia, L. Gonzalez, R. Cleries, M. Nadal, E. Condom et al., Altered expression of 12 S/MT-RNR1, MT-CO2/COX2, and MT-ATP6 mitochondrial genes in prostate cancer, Prostate 68 (2008) 1086-1096.

[20] R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[21] L. Feliubadalo, M.L. Arbones, S. Manas, J. Chillaron, J. Visa, M. Rodes, et al., Slc7a9-deficient mice develop cystinuria non-I and cystine urolithiasis, Hum. Mol. Genet. 12 (2003) 2097-2108.

[22] E.J.G. Pitman, Significance tests which may be applied to samples from any population, J. R. Stat. Soc. Suppl. 4 (1937) 119-130.

[23] P.D.W. Kirk, P.H. Stumpf, Gaussian process regression bootstrapping: exploring the effects of uncertainty in time course data, Bioinformatics 25 (10) (2009) 1300-1306.

Anexo A

Testrapptio relative expresion analysis

date: 2017-01-02 13:04:24

Descriptive analysis

Before making the analysis of the data validity checks The validation criteria is explained in the section **Cleaning data criteria** at the end of the document.

Number of genes: 3
Number of data entered: 144
Number of genes without sufficient information: 0
Number of genes with less than three samples in one group: 0

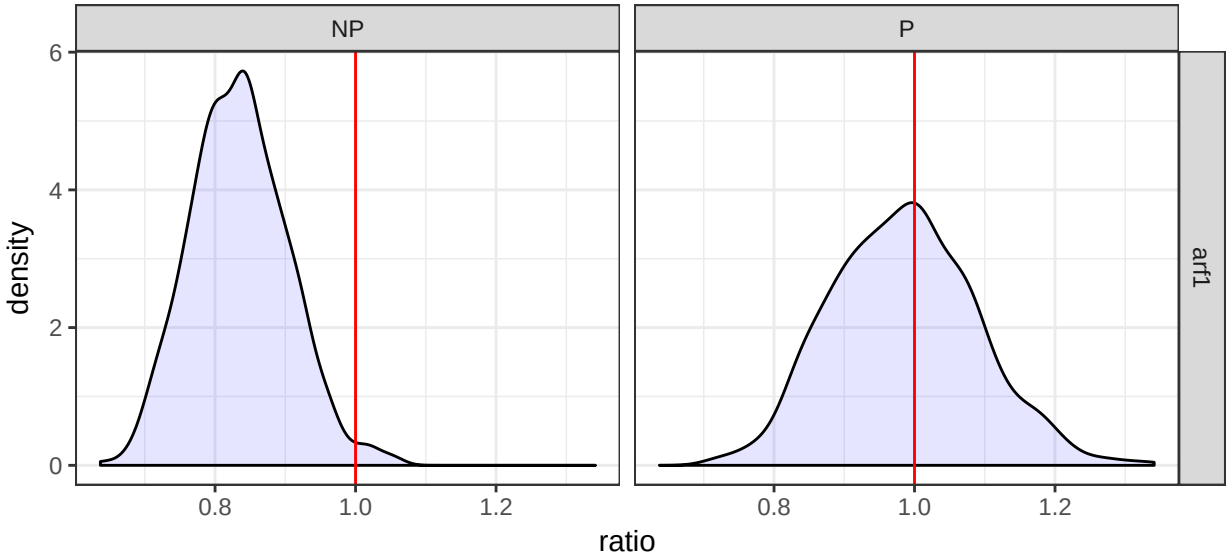
After eliminating those samples that do not meet the minimum requirements the final data show the following characteristics.

Number of genes analyzed: 3
Number of subgroups analyzed: 3

gene: arf1

Number of data gene: 48 Number of data by group:

group	N	Min	1st Qu.	Median	Mean	3rd Qu.	Max
CTRL	16	0.6965	0.8204	0.9720	1.0275	1.0448	1.7375
NP	16	0.5649	0.7559	0.7979	0.8540	0.9651	1.1641
P	16	0.6478	0.7823	0.9337	1.0033	1.1523	1.8752



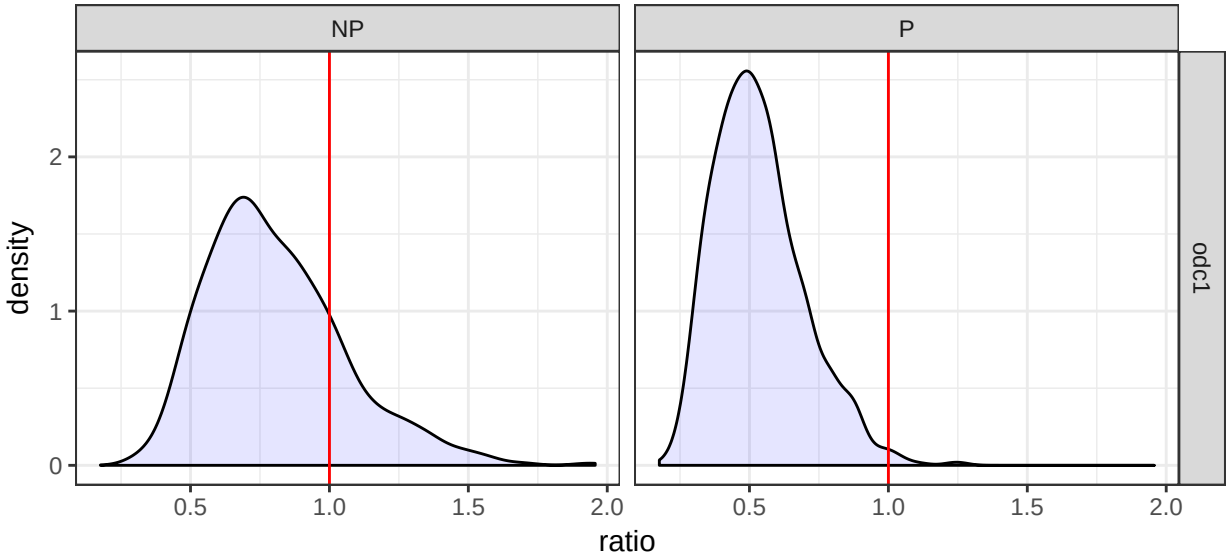
gene	group	mean	sd	min	Q1	Q2	Q3	max	P(A>B)	1-P(A>B)
arf1	NP	0.8345	0.0689	0.6367	0.7866	0.8338	0.8800	1.0609	0.0160	0.9840
arf1	P	0.9856	0.1023	0.7101	0.9121	0.9845	1.0544	1.3412	0.4374	0.5626

Gene: Name of the Gene. **Group**: Name of the group. **Mean**: The mean of the resamples. **Sd**: The standard devition of the resample's means. **Min**: The minimum value obtained in the resamples. **Q1**: The 1st quantile. **Q2**: The 2nd quantile. **Q3**: The 3rd quantile. **max**: The maximum value obtained in the resamples. **p**: The probability that ratio > 1.

gene: odc1

Number of data gene: 48 Number of data by group:

group	N	Min	1st Qu.	Median	Mean	3rd Qu.	Max
CTRL	16	0.3449	0.4196	1.1953	1.3333	1.8749	4.1152
NP	16	0.1479	0.2780	0.7961	1.0251	1.4308	3.8527
P	16	0.1141	0.2704	0.3812	0.6823	0.8276	2.0538



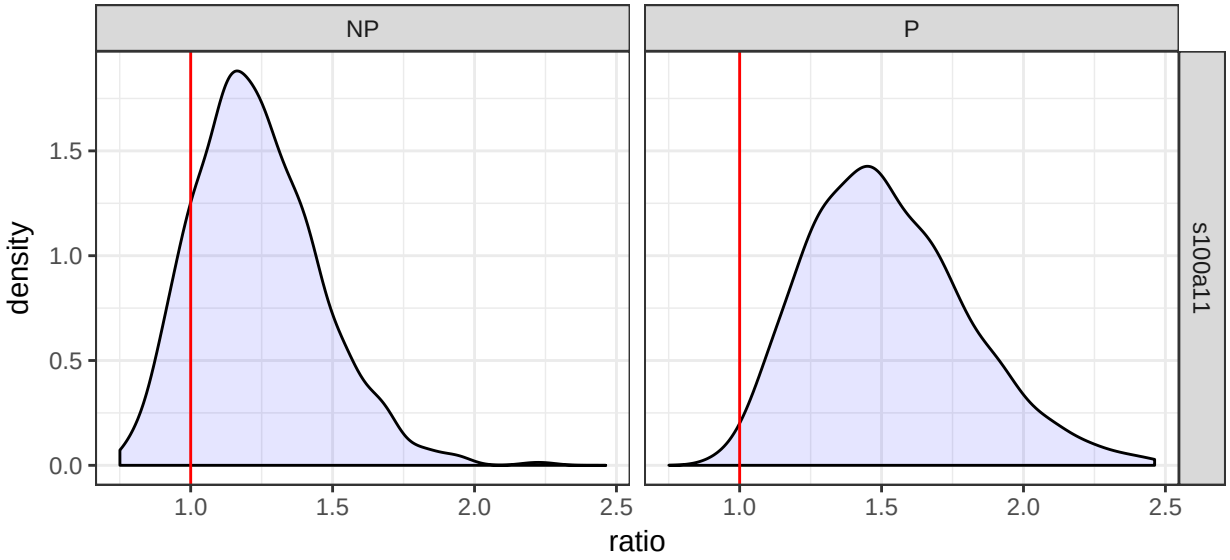
gene	group	mean	sd	min	Q1	Q2	Q3	max	P(A>B)	1-P(A>B)
odc1	NP	0.8069	0.2503	0.2937	0.6269	0.7690	0.9522	1.9567	0.1942	0.8058
odc1	P	0.5340	0.1614	0.1755	0.4169	0.5126	0.6288	1.2625	0.0080	0.9920

Gene: Name of the Gene. **Group**: Name of the group. **Mean**: The mean of the resamples. **Sd**: The standard devition of the resample's means. **Min**: The minimum value obtained in the resamples. **Q1**: The 1st quantile. **Q2**: The 2nd quantile. **Q3**: The 3rd quantile. **max**: The maximum value obtained in the resamples. **p**: The probability that ratio > 1.

gene: s100a11

Number of data gene: 48 Number of data by group:

group	N	Min	1st Qu.	Median	Mean	3rd Qu.	Max
CTRL	16	0.6986	0.8350	0.9768	1.0426	1.2253	1.8575
NP	16	0.6373	0.7550	1.1237	1.2823	1.3071	3.9844
P	16	0.7386	0.9332	1.1736	1.5835	1.7219	5.7480



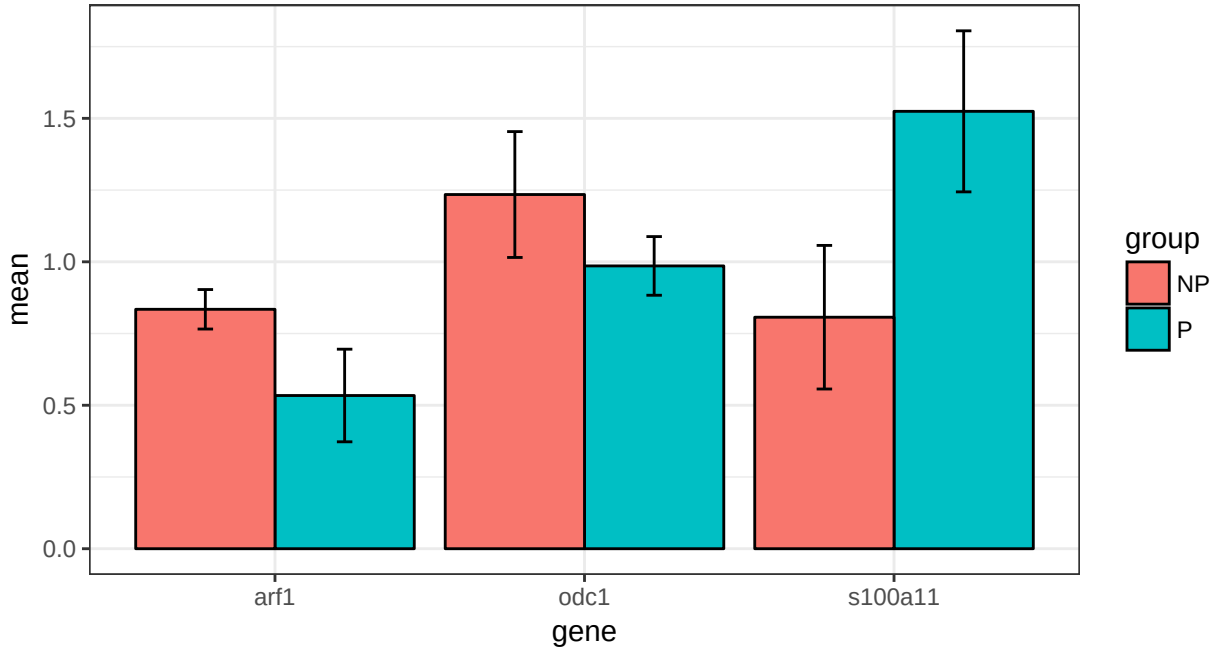
gene	group	mean	sd	min	Q1	Q2	Q3	max	P(A>B)	1-P(A>B)
s100a11	NP	1.2345	0.2194	0.7507	1.0811	1.2130	1.3708	2.2528	0.8649	0.1351
s100a11	P	1.5244	0.2806	0.9154	1.3162	1.4919	1.6964	2.4621	0.9940	0.0060

Gene: Name of the Gene. **Group**: Name of the group. **Mean**: The mean of the resamples. **Sd**: The standard devition of the resample's means. **Min**: The minimum value obtained in the resamples. **Q1**: The 1st quantile. **Q2**: The 2nd quantile. **Q3**: The 3rd quantile. **max**: The maximum value obtained in the resamples. **p**: The probability that ratio > 1.

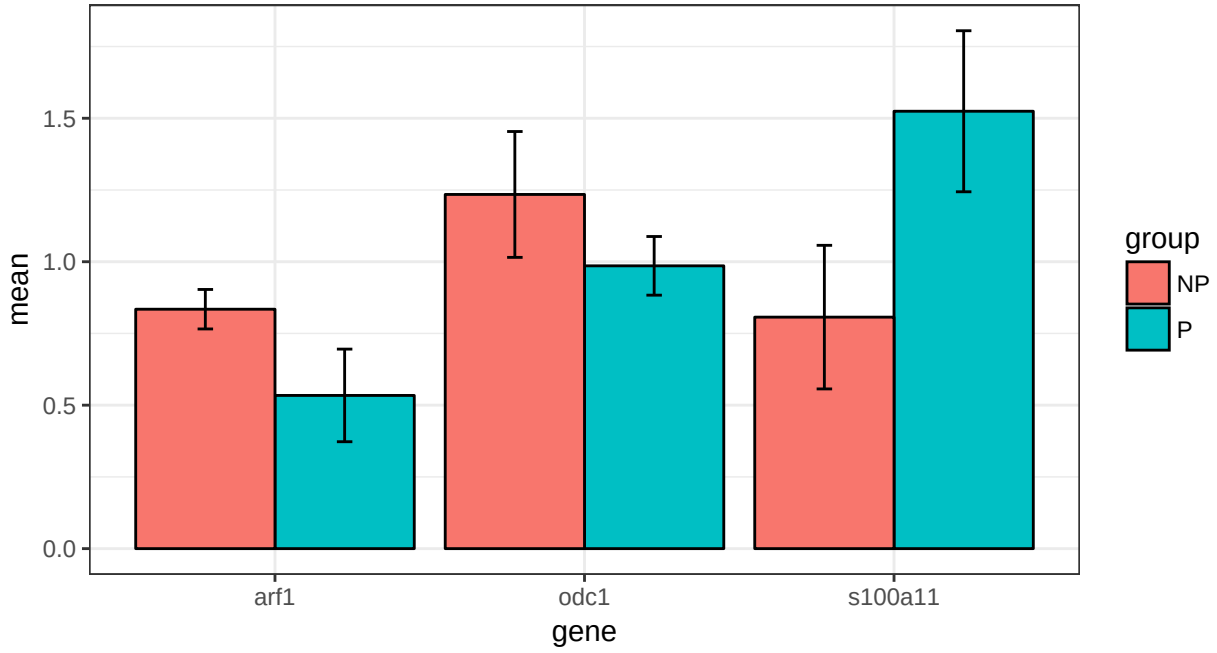
Resume

gene	group	mean	sd	min	Q1	Q2	Q3	max	P(A>B)	1-P(A>B)
arf1	NP	0.834	0.069	0.637	0.787	0.834	0.880	1.061	0.016	0.984
arf1	P	0.534	0.161	0.176	0.417	0.513	0.629	1.262	0.008	0.992
odc1	NP	1.234	0.219	0.751	1.081	1.213	1.371	2.253	0.865	0.135
odc1	P	0.986	0.102	0.710	0.912	0.984	1.054	1.341	0.437	0.563
s100a11	NP	0.807	0.250	0.294	0.627	0.769	0.952	1.957	0.194	0.806
s100a11	P	1.524	0.281	0.915	1.316	1.492	1.696	2.462	0.994	0.006

All results



Only genes with significant relative expresion



Cleaning data criteria

To ensure the robustness of the implemented results the the testrapptio team has decided to don't perform the test in that cases in which the number of samples is smaller than 7 when computed ratio is given. If two groups are provided then the minimum samples needed is 3 for each group with a minimum of 7 total samples. All the samples values shoud be bigger or equal to 0.

Anexo B

```
# Carga de los datos.
md <- read.delim("Dades SENSE GRUP CONTROL.txt", header=FALSE, stringsAsFactors = FALSE)
colnames(md) <- c('gene','value')
# Selección de los datos de interés
datos <- md[md$gene == '12s/MT-RNR1',]

# Asignación de los valores a los parámetros de evaluación
m = c(7,13,19)
ng = c(1, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000)
L = c(1000,2000,3000)
iter = 25

# Creación de tres datasets de longitudes iguales a las indicadas en m
de = lapply(m, function(m) datos[sample(1:nrow(datos), size = m),])

resultados =
  do.call('rbind',lapply(1:length(m), function(i){
    do.call('rbind',lapply(ng, function(j){
      do.call('rbind', lapply(L, function(k){
        do.call('rbind',lapply(1:iter, function(l){

          value = rep(de[[i]]$value, times = j)
          gene = rep(1:j, each = length(de[[i]]$value))

          tiempo <- system.time(testratio(value = value,gene = gene, m = L[i]))
          resultados = data.frame(m = m[i],
                                  ng = j,
                                  L = k,
                                  iter = l,
                                  user = tiempo[1],
                                  system = tiempo[2],
                                  elapsed = tiempo[3]
          )

          print(resultados)

          return(resultados)

        }))
      }))
    }))
  }))

require(dplyr)
r = resultados %>%
  group_by(m, ng, L) %>%
  summarise(user.sd = sd(user), system.sd = sd(system), elapsed.sd = sd(elapsed),
            user = median(user), system = median(system), elapsed = median(elapsed))

ggplot(data = r, aes(y = elapsed, x = ng, color = factor(L))) +
  geom_line(lwd =1) +
```

```
  geom_ribbon(aes(ymin = elapsed - elapsed.sd, ymax = elapsed + elapsed.sd),
              fill = "grey70", alpha = 0.1) +
  facet_grid(L ~ m) +
  theme_bw()



saveRDS(object = r, file = "r.RDS")
saveRDS(object = resultados, file = "resultados.RDS")
```