El significado de la “significancia”: Usando R para comprenderlo

Octavio Martínez de la Vega
Biología Computacional
Langebio – Cinvestav, Febrero 2011

Es una verdad cierta que, cuando no está en nuestra mano determinar lo que es verdad, debemos seguir lo que es más probable.
René Descartes

Resumen
Un concepto universalmente utilizado en investigación es la "significancia" estadística. Investigaremos, utilizando R, que queremos decir cuando afirmamos que una prueba de hipótesis fue "significativa" y damos un valor asociado "P". Utilizaremos datos reales de expresión genética discreta (contando etiquetas génicas en datos de SOLID) en dos tipos de genotecas. Aprenderemos a analizar "tablas de contingencia" por varios métodos, simularemos y obtendremos gráficas de los resultados.

Los datos
Utilizaremos datos obtenidos por Sandra González, del Lab. del Dr. Luis Herrera. Brevemente, ella creo y secuenció genotecas (conjunto de etiquetas) del transcriptoma de diversos genotipos de *A. thaliana*, algunos de los cuales eran “Tolerantes” (**To**) y otros “Intolerantes” (**In**) a la desecación (en semilla). Las genotecas fueron secuenciadas por SOLiD. No veremos los detalles del experimento que tiene un diseño complejo, pero uno de los objetivos es detectar genes que tuvieran una expresión diferencial con respecto a la tolerancia o intolerancia a la desecación. De los más de 21,000 genes detectados en ese trabajo, solo utilizaremos el locus AT4G28520, que corresponde a “cruciferina 3”, una proteína de almacenamiento de la semilla, característica del final de la maduración de esta. El número de etiquetas mapeadas al locus AT4G28520 en las genotecas **To** sumó **587,941** mientras que en las genotecas **In** la suma fue de **89,933** etiquetas. El número total de etiquetas mapeadas en cada condición fue de **9,160,694** para las **To** y de **5,743,943** para las **In**. La pregunta que queremos contestar es: ¿Qué tanta evidencia existe de “expresión diferencial” para el locus AT4G28520 en este experimento?. Por conveniencia llamaremos simplemente “A” al locus AT4G28520.

Capturando los datos en R
NOTAS: Lo que se tiene que teclear en R se presenta en **`Courier Negrita`** mientras que los resultados que arroja R se presentan en `Courier` (pueden copiar y pegar para ir siguiendo la práctica). Omitiré el cursor (*prompt*) de R, “>” y será conveniente que le den una vista a la ayuda de cada función que utilicemos; por ejemplo para ver que puede hacer la función “matrix” deberán utilizar “**`? matrix`**”. Por favor pregunten cualquier duda que vaya surgiendo. Por costumbre yo uso el operador “<-“ para asignación, aunque como han visto esto es equivalente al “=”.

La forma más adecuada para el análisis de los datos es introducirlos en una matriz de dos renglones y dos columnas, primero pongámoslos en un simple vector o cadena “`c`”

```
A <- c(587941, 89933, 9160694-587941, 5743943-89933)
# vector que contiene los cuatro números que necesitamos
```

Notemos que introducimos los datos de expresión de A en **To** y en **In**, seguidos por los números de expresión del resto de los genes (que podemos pensar como la expresión de "No-A") en las genotecas. Por ello restamos de los totales (9160694 y 5743943) la expresión de A. Ahora, podemos convertir A en una matriz:

```
A <- matrix(A, nrow=2, ncol=2, byrow=T, dimnames =
list(c("A","NoA"), c("To","In")))
```

```
A
          To      In
A     587941   89933
NoA 8572753 5654010
```

Notemos que reasignamos al objeto "A" el resultado de "matrix"; El vector original "A" es ahora un objeto del tipo "matrix" con dos renglones y dos columnas. Por "default" (omisión) "matrix" acomodará los datos POR COLUMNA, sin embargo nosotros queremos que los acomode "por renglón"; experimentemos para entender:

```
matrix(c(1,2,3,4), nrow=2, ncol=2)
     [,1] [,2]
[1,]    1    3
[2,]    2    4

matrix(c(1,2,3,4), nrow=2, ncol=2, byrow="T")
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

Además usamos el argumento "**dimnames = list(c("A","NoA"), c("To","In"))**" para definir los nombres de los renglones (primer componente de la lista) y de las columnas (segundo componente). La función "**list**" nos permite tener objetos complejos en forma ordenada; vea "**? list**" y experimente:

```
list( c("A","NoA"), c("To","In"))
[[1]]
[1] "A"   "NoA"

[[2]]
[1] "To" "In"
```

Note el doble corchete, que implica que podemos tener una LISTA de arreglos ordenados (vectores)… Regresando a nuestros datos, podemos usar la función "sum" para obtener la suma total (número total de etiquetas) que se mapearon en el experimento:

```
sum(A)
[1] 14904637
```

Ahora, podemos utilizar la función "apply" para obtener la suma por renglones (1) o columnas (2):

```
apply(A,1,sum) # "sum" se aplica a la primera dimensión (renglones)
       A      NoA
  677874 14226763
apply(A,2,sum) # "sum" se aplica a la segunda dimensión (columnas)
     To      In
9160694 5743943
```

En realidad estamos interesados en la frecuencia relativa con la cual se expresa el gen A en **To** e **In**; podemos obtener esas frecuencias relativas dividiendo A por la suma de sus columnas, digamos:

```
f.A <- A[1,]/apply(A,2,sum) # Frecuencia relativa de A
f.A
         To          In
0.06418084 0.01565701
```

Notemos que en la operación anterior ambos objetos (`A[1,]` y `apply(A,2,sum)`) tienen la misma dimensión (son vectores de tamaño 2); si tratamos de obtener la matriz A dividida por la suma de sus renglones DIRECTAMENTE obtenemos un resultado **erróneo**:

```
A/apply(A,2,sum)
              To          In
A   0.06418084 0.00981727
NoA 1.49248574 0.98434299
```

Esto se debe a la manera que R interpreta el operador "/" en esa operación; R esta efectuando la operación "/" POR CULUMNAS, "reciclando" el vector **apply(A,2,sum)** dos veces (una para cada columna). Para obtener el resultado correcto (es decir, dividir cada número entre el respectivo total de columna) podemos utilizar la función "**rep**" (repeat); veamos:

```
rep(apply(A,2,sum), each=2) # Cada elemento del vector repetido dos veces
     To      To      In      In
9160694 9160694 5743943 5743943
```

```
f.A <- A/rep(apply(A,2,sum), each=2) # Ahora si, nos da f[A|To], ... etc
```

```
f.A
              To          In
A   0.06418084 0.01565701
NoA 0.93581916 0.98434299
```

Notemos que la suma de **f.A** por columnas es uno (como debe ser):

```
apply(f.A,2,sum)
```

```
To In
 1  1
```

Comenzando el análisis de los datos

Regresando al problema que nos ocupa, ¿Es la frecuencia relativa de expresión de A "igual" (o cuando menos "parecida") en ambas condiciones (**To** e **In**)?. ¡Evidentemente NO! La frecuencia relativa de "A" en **To** es más de cuatro veces la frecuencia relativa de A en **In**:

```
f.A[1,1]
[1] 0.06418084

f.A[1,2]
[1] 0.01565701

f.A[1,1]/f.A[1,2]
[1] 4.099175
```

¿Será esta diferencia en frecuencias relativas "significativa"?; ¿qué queremos decir con eso?...

Un poquito de probabilidad

En lo que sigue asumiremos que las "frecuencias relativas" estiman las "probabilidades" de los eventos. Evidentemente lo que nos indican los datos es que existe una relación entre la expresión del gen "A" y el genotipo de los individuos (**To** o **In**). Si no existiera relación, es decir si los dos criterios de clasificación fueran INDEPENDIENTES, ¿qué frecuencias relativas esperaríamos?. La definición de "independencia de dos eventos C y D nos dice que C y D son independientes si y solo sí P[C ∩ D] = P[C] P[D]; en palabras, si la probabilidad de que C y D ocurran "al mismo tiempo" es el producto de sus probabilidades. Nosotros tenemos un estimado de las probabilidades de cada una de las celdas, digamos los eventos "A ∩ **To**", **"**A ∩ **In**", "NoA ∩ **To**" y "NoA ∩ **In**"; estas probabilidades estimadas son:

```
pe.A <- A/sum(A) # Dividimos cada celda por el total...

pe.A
             To          In
A   0.03944685 0.006033894
NoA 0.57517355 0.379345703
```

Es decir, P[A ∩ **To**]= 0.03944685, P[A ∩ **In**]= 0.006033894,
P[NoA ∩ **To**]= 0.57517355y P[NoA ∩ **In**]= 0.379345703.

También podemos obtener las probabilidades estimadas de cada evento individual, sumando por renglones y por columnas:

```
apply(pe.A,1,sum) # Probabilidades estimadas de A y NoA...
         A        NoA
0.04548075 0.95451925
```

```
apply(pe.A,2,sum) # Probabilidades estimadas de To e In...
       To        In
0.6146204 0.3853796
```

Es decir, estimamos que:

P[A]= 0.04548075, P[NoA]= 0.95451925, P[**To**]= 0.6146204 y P[**In**]= 0.3853796

Proponiendo una hipótesis

Ahora, supongamos que proponemos la HIPÓTESIS de independencia, digamos H0; entonces -si los criterios de clasificación "no tienen nada que ver", ESPERARIAMOS que:

P[A ∩ **To** | H0] = P[A] P[**To**] = 0.04548075 x 0.6146204 = 0.02795340,
P[A ∩ **In** | H0] = P[A] P[**In**] = 0.04548075 x 0.3853796 = 0.01752735,
P[NoA ∩ **To** | H0] = P[NoA] P[**To**] = 0.95451925 x 0.6146204 = 0.586667
y
P[NoA ∩ **In**|H0] = P[NoA] P[**In**] = 0.95451925 x 0.3853796 = 0.3678522

Estas cuentas se pueden hacer en R utilizando dos operadores de matrices (%*% y t):

```
apply(pe.A,1,sum) %*% t(apply(pe.A,2,sum))
             To         In
[1,] 0.02795339 0.01752735
[2,] 0.58666701 0.36785225
```

De manera que podemos definir las probabilidades esperadas en H0 como:

```
pe.A.H0 <- apply(pe.A,1,sum) %*% t(apply(pe.A,2,sum))
# Ahora le ponemos el atributo de nombres de renglones:
attributes(pe.A.H0)$dimnames[[1]] <- c("A","NoA")

pe.A.H0
             To         In
A   0.02795339 0.01752735
NoA 0.58666701 0.36785225
```

Recordemos que

```
pe.A
             To          In
A   0.03944685 0.006033894
NoA 0.57517355 0.379345703
```

Una definición importante es la de probabilidad condicional; la probabilidad condicional de C dado que D a ocurrido, denotada como P[C | D] está definida como:
P[C | D] = P[C ∩ D] / P[D] siempre que P[D] > 0. En la matriz f.A tenemos las probabilidades CONDICIONALES observadas de A dado **To** y de A dado **In**, etc.

```
f.A
             To          In
A   0.06418084 0.01565701 # P[A|To], P[A|In]
NoA 0.93581916 0.98434299 # P[NoA|To], P[NoA|In]

apply(f.A,2,sum) # Suma por columnas de f.A
To In
 1  1
```

Probabilidades si la hipótesis nula es cierta

Ahora obtengamos las probabilidades condicionales de A dado **To** y de A dado **In** **asumiendo que nuestra hipótesis nula es cierta**; estas probabilidades las podemos obtener dividiendo las probabilidades `pe.A.H0` por sus totales de columna…

```
rep(apply(pe.A.H0,2,sum),each=2) # Para hacer la división…
       To        To        In        In
0.6146204 0.6146204 0.3853796 0.3853796

pe.A.H0 / rep(apply(pe.A.H0,2,sum),each=2)
             To          In
A   0.04548075 0.04548075
NoA 0.95451925 0.95451925
```

Notemos que la hipótesis de independencia, H0, implica que la frecuencia relativa (esperada) de expresión de A es idéntica en ambos tipos de genotipos (**To** e **In**). Podemos obtener los "valores esperados" de A dado que H0 es cierta multiplicando `pe.A.H0` por el la suma total de etiquetas: 14,904,637; es decir:

```
EA.H0 <- pe.A.H0*sum(A)
EA.H0
            To         In
A    416635.2  261238.8
NoA 8744058.8 5482704.2
```

Esas frecuencias de etiquetas son las que se esperaría encontrar si es que nuestra hipótesis H0 es cierta; comparémoslas visual y numéricamente con las frecuencias observadas:

```
A # Frecuencias observadas (matriz original de datos)
         To      In
A    587941   89933
NoA 8572753 5654010

A-EA.H0 # La diferencia entre observados y esperados
            To         In
A    171305.8 -171305.8
NoA -171305.8  171305.8
```

Podemos notar que tenemos un exceso muy grande de etiquetas observadas para el gen A en la genotecas **To** y lo inverso es cierto para la genoteca **In**.

Ahora lo importante: Significancia

Ahora la pregunta es ¿es esa diferencia "significativa"?. O más claramente: ¿Es posible que esa diferencia sea dada por "azar" siendo la hipótesis nula cierta?. Para contestar esta pregunta necesitamos encontrar un "estadístico de prueba" cuya distribución sea conocida cuando H0 es cierta; es decir, un valor que nos permita evaluar la probabilidad de que los datos observados hayan ocurrido por azar… Antes de continuar, podemos detenernos a pensar un momento sobre que tipos de error podemos cometer. Supongamos que queremos "rechazar" H0 o "aceptarla", en primer lugar podemos plantear una alternativa para nuestra hipótesis H0, digamos Ha, formalmente:

H0: "Los criterios de clasificación (tipo de genoteca y expresión génica) son independientes", i.e., $P[A \cap \mathbf{To}] = P[A]\,P[\mathbf{To}]$.
Ha: "Los criterios de clasificación son dependientes", i.e., $P[A \cap \mathbf{To}] \neq P[A]\,P[\mathbf{To}]$

Ahora los tipos de error que podemos cometer:

	Estado de la Naturaleza (realidad)	
Decisión:	H0 es verdadera	H0 es falsa
Rechazar H0	Error tipo I, Probabilidad α	Correcto, Probabilidad $1-\alpha$
Aceptar H0	Correcto, Probabilidad $1-\beta$	Error tipo II, Probabilidad β

α = P[Rechazar H0 cuando H0 es verdadera]
β = P[Aceptar H0 cuando H0 es falsa]
A "1-β", la probabilidad de aceptar H0 cuando H0 es verdadera se le conoce como "potencia de la prueba" y en algunos casos $1-\alpha$ se conoce como la "significancia de la prueba".

Existen varias formas de poner a prueba nuestra hipótesis en este caso. La primera y más simple es la conocida como "prueba de chi cuadrada" (o ji-cuadrada), en ese caso el "estadístico de prueba" esta dado por:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \qquad (1)$$

en donde Oi son los valores observados (en cada i-casilla de la tabla) y Ei son los valores esperados correspondientes. Notemos que cuando los valores observados y esperados son muy parecidos, el valor del estadístico tenderá a ser pequeño, mientras que si las diferencias son grandes (independientemente del signo, pues elevamos al cuadrado), el valor del estadístico será grande… Lo importante es que, bajo la hipótesis nula H0 la distribución del estadístico es conocida y resulta ser la distribución chi cuadrada con un "grado de libertad". Obtengamos una gráfica de esa distribución en R:

```
plot(c(1:1000)/100, dchisq(c(1:1000)/100,df=1), type="l",
xlab="x", ylab="f(x) - Densidad Chi Cuadrada",
main="Grafica de densidad Chi Cuadrada con un grado de
libertad", col="red")
```

De la gráfica anterior vemos que los valores mayores a seis son muy poco probables… De echo podemos calcular que valores acumulan una proporción dada (área bajo la curva) de la probabilidad, veamos:

```
qchisq(.95,df=1) # Valor para 1-alfa=0.95; alfa=0.05
[1] 3.841459
qchisq(.99,df=1) # Valor para 1-alfa=0.99; alfa=0.01
[1] 6.634897
qchisq(.999,df=1) # Valor para 1-alfa=0.999; alfa=0.001
[1] 10.82757
```

Ahora podemos calcular el valor del estadístico en nuestros datos usando la formula en (1), pero ANTES podemos decidir que probabilidad de error tipo I, α, estamos dispuestos a considerar adecuada. Supongamos que queremos ser "muy estrictos" y poner $\alpha = 0.000001 = 1e\text{-}6$ (solo una probabilidad de 1 en un millón de cometer un error tipo I); entonces R nos ayuda a formular una regla de decisión calculando el valor crítico de chi cuadrada:

```
qchisq(1-0.000001,df=1) # Valor "crítico" de chi cuadrada
[1] 23.92813
```

De acuerdo con esto, formulamos una regla de decisión:
"Rechazar H0 si y solo sí $\chi^2 \geq 23.93$"
Ahora hagamos las cuentas:

```
A-EA.H0
            To         In
A    171305.8 -171305.8
NoA -171305.8  171305.8

(A-EA.H0)^2 # Nos da los valores de (Oi - Ei) al cuadrado
              To          In
A   29345680168 29345680168
NoA 29345680168 29345680168

((A-EA.H0)^2)/EA.H0 # Nos da los valores de (Oi - Ei)^2 / Ei
            To         In
A   70434.953 112332.77
NoA  3356.071   5352.41

sum(((A-EA.H0)^2)/EA.H0) # Nos da el valor del estadístico chi cuadrada
[1] 191476.2
```

Obtenemos un valor ENORME (191,476.2) para chi cuadrada que evidentemente es mucho mayor que el valor "crítico" que habíamos predeterminado, 23.93, de manera que rechazamos H0 con "toda confianza". Es más, si tratamos de calcular (con R) el valor de significancia:

P = P[probabilidad de error de tipo I dados los datos] = P[$\chi^2 \geq 191476.2$]

```
1-pchisq(191476.2, df=1) # 1 - P[Chi<=191476.2]
[1] 0
pchisq(191476.2, df=1, lower.tail=FALSE) # P[Chi>191476.2]
[1] 0
```

(esto es un error de "redondeo" porque la probabilidad no es exactamente 0, sino posiblemente un valor extremadamente pequeño, menor que 1e-200)
Por supuesto R tiene pre-programada una función para realizar la prueba de Chi cuadrada:

```
chisq.test(A) # Realizando la prueba sobre la matriz A

     Pearson's Chi-squared test with Yates' continuity
correction

data:  A
X-squared = 191475.1, df = 1, p-value < 2.2e-16
```

```
chi.A <- chisq.test(A) # Guardando los resultados para
posterior análisis

attributes(chi.A)$names
[1]   "statistic"   "parameter"   "p.value"        "method"
"data.name" "observed"
[7] "expected"  "residuals"

chi.A$expected
           To        In
A    416635.2  261238.8
NoA 8744058.8 5482704.2

EA.H0
           To        In
A    416635.2  261238.8
NoA 8744058.8 5482704.2
```

La pequeña diferencia con nuestro resultado es porque R utiliza la corrección de Yates para continuidad:

```
chisq.test(A, correct=FALSE) # Realizando la prueba sin la
correccion de Yates.

     Pearson's Chi-squared test

data:  A
X-squared = 191476.2, df = 1, p-value < 2.2e-16
```

La prueba que acabamos de hacer es "aproximada" en el sentido de que la distribución del estadístico de prueba es solamente parecida a la distribución chi cuadrada. En este

caso existe una prueba "exacta" y no-paramétrica que permite calcular la suma de la probabilidad de TODAS las tablas más extremas que la que se presenta. Esta prueba usa bastantes recursos de cómputo. En R podemos hacerla con:

```
fish.A <- fisher.test(A) # Prueba exacta de Fisher para nuestros datos

fish.A
     Fisher's Exact Test for Count Data
data:  A
p-value < 2.2e-16
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 4.281098 4.343020
sample estimates:
odds ratio
  4.311691
```

Aquí se utiliza como hipótesis nula el hecho de que la razón de los momios ("odds ratio") es igual a 1. Esto es TOTALMENTE equivalente a la hipótesis de independencia y la prueba de Fisher nos calcula un intervalo de confianza para la razón muestral de las probabilidades; en este caso el intervalo va de 4.28 a 4.34 con un valor medio de 4.31 de manera que podemos decir que la probabilidad de A en **To** es aproximadamente 4.31 veces mayor que en **In** (nuestro estimado muestral fue igual a `4.099175`)

Como interpretación de lo anterior concluimos que podemos estar muy seguros de que existe una diferencia de expresión significativa en el locus A entre las genotecas **To** e **In**. Estamos "tan seguros" porque la probabilidad de que eso haya ocurrido por azar es tremendamente pequeña; redondeada a 0 en nuestros cálculos y seguramente menor a 1e-16, es decir nuestra probabilidad de "error" $\alpha < 1 / 10{,}000{,}000{,}000{,}000{,}000$.

¿Por qué nuestra prueba tiene una probabilidad de error tipo I tan pequeña?. Esto se debe a dos factores: a) El número total de etiquetas que estamos usando es grande (sum(A) = 14,904,637) y b) La razón entre las probabilidades condicionales P[A|To] / P[A|In] = f.A[1,1] / f.A[1,2] = 4.099175 es, también muy grande (lejana a 1). Tratemos de experimentar con los datos para entender mejor el significado de la "significancia".

Supongamos primero que reducimos el número total de etiquetas por un factor de 100,000:

```
A.dis <- A/100000 # Nuestra nueva tabla "disminuida"

A.dis # Aún cuando no tiene valores enteros (como debería)
          To       In
A    5.87941  0.89933
NoA 85.72753 56.54010

(5.87941/(5.87941+85.72753))/(0.89933/(0.89933+56.54010))
[1] 4.099175
```

```
chisq.test(A.dis)

     Pearson's Chi-squared test with Yates' continuity
correction

data:  A.dis
X-squared = 0.9601, df = 1, p-value = 0.3272

Warning message:
In chisq.test(A.dis) : Chi-squared approximation may be
incorrect

fisher.test(A.dis)

     Fisher's Exact Test for Count Data

data:  A.dis
p-value = 0.2493
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
   0.4602213 185.9179737
sample estimates:
odds ratio
   3.94794

Warning message:
In fisher.test(A.dis) :
  'x' has been rounded to integer: Mean relative
difference: 0.00639821
```

Aún cuando obtenemos una tabla con un "conteo" no entero en las celdas y valores muy pequeños en dos de las celdas (por eso los "Warnings"), el argumento que quiero mostrar es que ahora una diferencia (razón) de probabilidades del mismo tamaño NO resulta "significativa", ni por la prueba de chi cuadrada (P=0.33) ni por la prueba exacta de Fisher (P=0.25); en otras palabras, con ese número pequeño de etiquetas, sum(A.dis) = 149 NO podemos rechazar ni siquiera una diferencia tan grande como la presente en las probabilidades condicionales. Podemos explorar lo que sucede al ir reduciendo el tamaño de la muestra por factores de, digamos 1, 10, 100, …, 100000; veamos:

```
factor <- 10^c(0:5) # Factor de reducción de la muestra 1,
10, 100, 1000, 10000, 100000.
n <- sum(A)/factor # Aqui guardaremos el tamaño de muestra
val.chi <- c() # Aqui guardaremos el valor del estadistico
chi cuadrada
val.P <- c() # Aqui guardaremos el valor de P
(significancia)

for(i in c(1:6)){ # Iniciamos un "bucle" para hacer las
cuentas en cada caso.
temp <- chisq.test(A/factor[i]) # Guardamos en un objeto
"temp" el resultado de la prueba
```

```
val.chi <- c(val.chi, temp$statistic) # Recuperamos el valor del estaditico
val.P <- c(val.P, temp$p.value ) # Recuperamos el valor de P
} # Cerramos el bucle, terminando la ejecucion
varNinA <- data.frame(n,factor,val.chi,val.P) # Reunimos las columnas en un marco de datos

varNinA
  n             factor val.chi     val.P
1 1.490464e+07  1e+00 1.914751e+05 0.000000e+00
2 1.490464e+06  1e+01 1.914650e+04 0.000000e+00
3 1.490464e+05  1e+02 1.913644e+03 0.000000e+00
4 1.490464e+04  1e+03 1.903601e+02 2.652959e-43
5 1.490464e+03  1e+04 1.804619e+01 2.156099e-05
6 1.490464e+02  1e+05 9.601384e-01 3.271520e-01
```

Ahora podemos obtener una gráfica que nos muestre la relación en una escala adecuada…

```
plot(log10(varNinA$n), log10(varNinA$val.chi), ylim=c(-0.5, 5.5), xlim=c(2,7.5), xlab="Log 10 del tamaño de muestra", ylab="Log 10 del valor del estadístico Chi cuadrada",main="Relación entre el tamaño de la muestra y el valor del\nestadístico chi cuadrada",sub="Datos para el gen A")

points(log10(varNinA$n), log10(varNinA$val.chi), type="l", col="red")

text(log10(varNinA$n), log10(varNinA$val.chi), labels=round(varNinA$val.P,7), pos=3, cex=.7, col="blue")

text(log10(varNinA$n), log10(varNinA$val.chi), labels=round(varNinA$n), pos=1, cex=.7, col="orange")

legend("topleft", legend=c("P - significancia","n - Tamaño de muestra"), text.col=c("blue","orange"),cex=0.7, bty="n")
```

En la figura anterior podemos ver como existe una relación lineal entre los logaritmos del valor del estadístico chi cuadrada con el tamaño de la muestra (número total de etiquetas muestreadas). Podemos explorar más finamente que sucede con muestras pequeñas, digamos entre 100 y 800 etiquetas en total. Ahora, para crear la matriz a analizar utilizaremos la matriz de frecuencias relativas estimadas, `pe.A`, multiplicándola por el tamaño de muestra deseado…

```
n <- c(100:800) # Los tamaños de muestra que exploraremos
val.chi <- c()
val.P <- c()
```

```
for(i in c(1:701)){ # Iniciamos un "bucle" para hacer las cuentas en cada caso.
temp <- chisq.test(n[i]*pe.A) # Guardamos en un objeto "temp" el resultado de la prueba
val.chi <- c(val.chi, temp$statistic) # Recuperamos el valor del estadístico
val.P <- c(val.P, temp$p.value ) # Recuperamos el valor de P
} # Cerramos el bucle, terminando la ejecución

varNinA2 <- data.frame(n,val.chi,val.P)

plot(varNinA2$n, varNinA2$val.P, type="l", col="red", xlab="n - Tamaño de muestra", ylab="P - significancia de la prueba", main="Valores de significancia para la prueba chi cuadrada\nen la tabla de contingencia",sub="(Datos del gen A)") # Todos los valores del dataset

plot(varNinA2$n[val.P<0.01], varNinA2$val.P[val.P<0.01], type="l", col="red", xlab="n - Tamaño de muestra", ylab="P - significancia de la prueba", main="Valores de significancia para la prueba chi cuadrada\nen la tabla de contingencia",sub="(Datos del gen A). Solamente valores de P<0.01") # Solo los valores con P<0.01
```

Hemos visto como, para una diferencia tan grande en las proporciones o frecuencias relativas de expresión en las genotecas, con un número relativamente pequeño de etiquetas podemos estar "muy seguros" de que la diferencia en frecuencia de expresión es real y no producto del azar. Ahora veremos que sucede si, con el mismo tamaño de muestra, n = 14,904,637 la diferencia entre las probabilidades es menor. Recordemos nuestra matriz original (los totales se los puse yo, no están en R):

A

	To	In	Total
A	587941	89933	677874
NoA	8572753	5654010	14226763
Total	9160694	5743943	14904637

Si restamos y sumamos n elementos de la manera que se muestra (no correr en R)

A.mod

	To	In	Total
A	587941-n	89933+n	677874
NoA	8572753+n	5654010-n	14226763
Total	9160694	5743943	14904637

NO modificamos los totales (sumas) de los renglones ni tampoco los de las columnas; tampoco modificamos la suma total de etiquetas, ¿verdad?. Lo que SI modificamos es la relación de las probabilidades condicionales estimadas, veamos numéricamente:

En A:

P[A|**To**] = 587941/9160694=0.06418084, P[A|**In**] = 89933/5743943= 0.01565701; P[A|**To**] / P[A|**In**] =0.06418084/0.01565701=4.099176

En A.mod:

P[A|**To**] = (587941-n)/9160694, P[A|**In**] = (89933+n)/5743943 y
P[A|**To**] / P[A|**In**] =((587941-n)/9160694)/((89933+n)/5743943)

Y este último valor será siempre menor que 4.099176 cuando n>0, comprobemos:

```
n <- 0 # Para n = 0
((587941-n)/9160694)/( (89933+n)/5743943)
[1] 4.099175
n <- 100 # Para n = 100
((587941-n)/9160694)/( (89933+n)/5743943)
[1] 4.093925
n <- 1000 # Para n = 1000
((587941-n)/9160694)/( (89933+n)/5743943)
[1] 4.0472
n <- 100000 # Para n = 100000
((587941-n)/9160694)/( (89933+n)/5743943)
[1] 1.610826
```

Programemos una funcioncita en R para hacer obtener A.mod (la matriz modificada) a partir de A dando el valor de n:

```
modA <- function(n) {A + c(-n, n, n, -n)}

A # La matriz original
         To      In
A    587941   89933
NoA 8572753 5654010

modA(10) # La matriz modificada con n=10
         To      In
A    587931   89943
NoA 8572763 5654000
```

Ahora programemos otra función para obtener el radio o razón de las probabilidades (P[A|**To**] / P[A|**In**]):

```
radio <- function(X) { (X[1,1]/(X[1,1]+X[2,1])) /
(X[1,2]/(X[1,2]+X[2,2])) }

radio(A)
[1] 4.099175
radio(modA(1000))
[1] 4.0472
radio(modA(10000))
[1] 3.626238
radio(modA(100000))
[1] 1.610826
```

Ahora estamos preparados para investigar como la diferencia en las probabilidades de expresión (o más exactamente la razón entre estas) afecta la “significancia”. Notemos

que en la prueba estadística siempre estamos comparando con los valores "esperados" bajo la hipótesis de independencia:

```
A
          To      In
A    587941   89933
NoA 8572753 5654010

EA.H0 # Valores esperados bajo independencia en nuestra matriz original
           To        In
A    416635.2  261238.8
NoA 8744058.8 5482704.2

radio(EA.H0) # radio de las probabilidades esperadas
[1] 1

587941 - 416635 # Cambio de A[1,1] en la observada respecto a la esperada
[1] 171306
```

De manera que podemos explorar el cambio de "significancia" para n desde 0 a 171,306…

```
n <- 1000*c(0:172) # Valores en que cambiaremos n (de 0 a 172000)
p.ATo <- c() # Vector para guardar P[A|To]
p.AIn <- c() # Vector para guardar P[A|In]
p.dif <- c() # Vector para guardar P[A|To]-P[A|In]
p.radio <- c() # Vector para guardar P[A|To] / P[A|In]
chi.cuad <- c() # Vector para guardar el valor del estadístico chi cudrada
P <- c() # Vector para guardar la significancia del estadístico chi cudrada

for(i in 1:173) { # Para todos los valores desde i a 173 (longitud de n)
temp <- modA(n[i]) # Matriz A, pero haciendo la operación con n[i]
p.ATo <- c(p.ATo, temp[1,1]/(temp[1,1]+temp[2,1])) # Prob estimada de A|To
p.AIn <- c(p.AIn, temp[1,2]/(temp[1,2]+temp[2,2])) # Prob estimada de A|In
p.dif <- c(p.dif, p.ATo[i]-p.AIn[i]) # Diferencias de las probabilidades
p.radio <- c(p.radio, radio(temp)) # Radio o razón de las probabilidades
temp.test <- chisq.test(temp) # Objeto con los resultados de la prueba estadistica
chi.cuad <- c(chi.cuad, temp.test$statistic) # Capturamos el valor del estadístico
```

```
P <- c(P, temp.test$p.value) # Capturamos el valor de P
} # cerramos el bucle para ejecutar...

sign.r <- data.frame(n, p.ATo, p.AIn, p.dif, p.radio,
chi.cuad, P) # Marco de datos

attach(sign.r) # Ponemos el marco de datos "a la vista"
```

Ahora podemos graficar la significancia como función del radio de las probabilidades…

```
plot(sign.r$p.radio, sign.r$P, type="l", xlab= "radio = P[A
| To] / P[A | In]", ylab="Significancia (P)", main =
"Significancia como función del radio\nen tablas con
14,904,637 etiquetas")
```

O bien como función de la diferencia en probabilidades…

```
plot(sign.r$p.dif, sign.r$P, type="l", xlab="diferencia =
P[A | To] - P[A | In]", ylab="Significancia (P)",
main="Significancia como función de la diferencia de
probabilidades\nen tablas con 14,904,637 etiquetas")
```

Nos damos cuenta que la prueba es muy potente, es decir, tiene capacidad de detectar significancia aún para radios cercanos a 1 o, equivalentemente para diferencias cercanas a 0. Podemos echarle un vistazo a las gráficas solamente para los valores de P que NO son cero…

```
plot(sign.r$p.radio[sign.r$P>0], sign.r$P[sign.r$P>0],
type="l",xlab="radio = P[A | To] / P[A | In]",  ylab =
"Significancia (P)", main="Significancia como función del
radio\nen tablas con 14,904,637 etiquetas", sub="Solo
valores P>0")

plot(sign.r$p.dif[sign.r$P>0], sign.r$P[sign.r$P>0],
type="l",xlab="diferencia = P[A | To] - P[A | In]",
ylab="Significancia (P)", main="Significancia como función
de la diferencia de probabilidades\nen tablas con
14,904,637 etiquetas", sub="Solo valores P>0")
```

A *groso modo* vemos que, con tal número de etiquetas, somos capaces de detectar diferencias en radios de probabilidades que están tan cerca de 1 como 1.02 o, diferencias que están tan cerca de 0 como 0.001

Dado que en el experimento de Sandra se detectaron las diferencias de más de 21,000 genes, es importante saber la significancia que obtendremos de tablas que tengan la estructura:

	To	**In**	Total
A	a	b	A+b
NoA	9,160,694 – a	5,743,943 – b	14,904,637 – (a+b)
Total	9,160,694=c	5,743,943=d	14,904,637

Variando a y b podemos crear una tabla con distintas probabilidades y estudiar sus propiedades, sobre todo la "significancia" de una prueba. Notemos que los totales de columnas están fijos y que tenemos mucho más etiquetas en las genotecas **To** que en las **In**. Podemos interesarnos en la diferencia entre P[A|**To**] y P[A|**In**] o bien en su sazón:
r = P[A|**To**] / P[A|**In**] = (a / c) / (b / d) = (a * d) / (b * c) = (a / b) * cons, en donde

```
cons <- 5743943 / 9160694 # Que es igual a 0.6270205
```

Esta constante nos da la equivalencia entre el cambio de las probabilidades condicionales estimadas de la tabla como función del número de etiquetas, en particular entre P[A|**To**] y P[A|**In**] y por lo tanto con respecto a su radio (razón). Por ejemplo cuando r=1 tenemos que b = a * cons ; por ejemplo para a=1

```
tot.col.A <- apply(A,2,sum) # Los totales de las columnas
de la matriz original

c(1, cons)/tot.col.A # Para b=1, a=to.in
To           In
1.091620e-07 1.091620e-07

c(10000, 10000*cons)/tot.col.A # Para a=10000,
 To          In
0.001091620 0.001091620
```

Así pues r > 1 implica que a*cons > b (genes sobre-expresados en **To**), mientras que r < 1 implica que a*cons < b (genes sobre-expresados en **In**). Programemos una función que nos forme automáticamente una matriz del tipo de A:

```
forma <- function(a, b) {matrix(c(a, 9160694-a, b, 5743943-
b), nrow=2, ncol=2,   dimnames = list(c("A", "NoA"),
c("To", "In")))}
```

Es fácil ver que para todos los casos en los cuales r=1 el valor del estadístico chi cuadrada será 0 y la "significancia" 1 (es decir, no hay discrepancia entre los valores observados y esperados y por tanto no hay "significancia"), por ejemplo:

```
A
         To      In
A    587941   89933
NoA 8572753 5654010

forma(a=587941, b=89933) # Formando una matriz idéntica a A
         To      In
A    587941   89933
NoA 8572753 5654010

B <- round(forma(a=100000, b=100000*cons)) # Una matriz con
r=1
```

```
B
          To      In
A    100000   62702
NoA 9060694 5681241
```

```
chisq.test(B)

     Pearson's Chi-squared test with Yates' continuity
correction

data:  B
X-squared = 0, df = 1, p-value = 0.998 # Por error de
redondeo no da p.value=1
```

Ahora podemos hacer un experimento doble, variando tanto el numero de etiquetas como el radio de probabilidades, r. Para el numero de etiquetas pongamos a de tal manera que P[A|**To**] = 1e-6, 1e-5, …, 1e-2;

```
prob.A.To <- 10^-c(6:2) # Las probabilidades P[A|To]

tags.a <- round(9060694 * prob.A.To) # Numeros de tags "a"
correspondientes a las P[A|To]

tags.a
[1]     9    91   906  9061 90607
```

Notemos que de la ecuación r = (a/b)*cons obtenemos b = (a*cons)/r. Ahora, podemos explorar radios (r) que estén relativamente "cerca" de 1 -puesto que sabemos que, para radios "grandes" (r>3) tendremos una α muy pequeña con un modesto numero de etiquetas. Exploremos los radios (aproximados[1]) r = 1, 1.01, 1.02, …, 3 y para cada valor de tags.a (9, 91, … 90607) –cinco valores, calcularemos el número correspondiente de etiquetas a, las matrices y sus estadísticos…

```
radios <- c(100:300)/100 # radios P[A|To]/P[A|In].
i1 <- c() # Indicador 1 (número de tags)
i2 <- c() # Indicador 2 (radios)
t.a <- c() # Para guardar el número de tags a
t.b <- c() # Para guardar el número de tags b
tags <- c() # para la suma de etiquetas t.a + t.b
r <- c() # Para guardar el radio
p.A.To <- c() # Para guardar P[A|To]
p.A.In <- c() # Para guardar P[A|In]
chi <- c() # Para guardar el valor del estadístico chi
P <- c() # Para guardar P (significancia)

k <- 0
for(i in c(1:5)){ # para cada valor en tags.a
for(j in c(1:201)) { # para cada valor de "radios"
```

[1] Dada la naturaleza discreta (conteos) de los datos, solamente podemos obtener radios (razones) aproximados.

```
i1 <- c(i1, i) # Indicador 1 (número de tags)
i2 <- c(i2, j) # Indicador 2 (radios)
k <- k+1
t.a <- c(t.a, tags.a[i])
t.b <- c(t.b, round(tags.a[i]*cons/radios[j]) )
tags <- c(tags, t.a[k]+t.b[k])
r <- c(r, t.a[k]*cons/t.b[k]) # radios REALES
#  tot.col.A
#      To      In
# 9160694 5743943
p.A.To <- c(p.A.To, t.a[k]/9160694)
p.A.In <- c(p.A.In, t.b[k]/5743943)
temp <- forma(a=t.a[k], b=t.b[k]) # Obtenemos la matriz
test <- chisq.test(temp) # Realizamos la prueba
chi <- c(chi, test$statistic) # Capturamos el estadístico
P <- c(P, test$p.value) # Capturamos la significancia
}
}

datos <- data.frame(i1, i2, t.a, t.b, tags, r, p.A.To,
p.A.In, chi, P)
```

Ahora podemos graficar nuestros resultados…

```
attach(datos) # Poniendo datos a la vista

plot(radios, datos$P[datos$i1==1], ylim=c(0,1), type="l",
col="black",
xlab = "Radios P[A|To]/P[A|In]", ylab = "Significancia
(P)",
main = "Valores de significancia de la prueba chi
cuadrada\ncuando se presentan distintos radios y
frecuencias de expresión",
sub = "(los radios son aproximados)")
points(radios, datos$P[datos$i1==2], type="l", col="red")
points(radios, datos$P[datos$i1==3], type="l", col="blue")
points(radios, datos$P[datos$i1==4], type="l",
col="purple")
points(radios, datos$P[datos$i1==5], type="l",
col="orange")
legend("topright", legend=c("1","10","100","1000","10000"),
bty="n", title="P[A|To] en transcritos por millon",
text.col=c("black", "red", "blue", "purple", "orange"))

# prob.A.To
# [1] 1e-06 1e-05 1e-04 1e-03 1e-02
# 1000000*prob.A.To # (expresadas en transcritos por
millón)
# [1]     1    10   100  1000 10000
```

En la gráfica anterior podemos ver como la significancia es una función muy sensible a el nivel de expresión de los genes (P[A|**To**]) y, secundariamente a la razón r = P[A|**To**] / P[A|**In**]. Examinemos con más detalle la región para la cual P<0.2 y r<2…

```
plot(radios[radios<=2], P[(i1==2)&(radios<=2)],
ylim=c(0,0.2), type="l", col="red",
xlab = "Radios P[A|To]/P[A|In]", ylab = "Significancia
(P)",
main = "Valores de significancia de la prueba chi
cuadrada\ncuando se presentan distintos radios y
frecuencias de expresión",
sub = "(los radios son aproximados)")
points(radios[radios<=2], P[(i1==3)&(radios<=2)], type="l",
col="blue")
points(radios[radios<=2], P[(i1==4)&(radios<=2)], type="l",
col="purple")
points(radios[radios<=2], P[(i1==5)&(radios<=2)], type="l",
col="orange")
legend("topright", legend=c("10","100","1000","10000"),
bty="n", title="P[A|To] en transcritos por millon",
text.col=c("red", "blue", "purple", "orange"),
title.col="black")
```

Otro acercamiento a la esquina inferior derecha…

```
plot(radios[radios<=1.2], P[(i1==3)&(radios<=1.2)],
ylim=c(0,0.05), type="l", col="blue",
xlab = "Radios P[A|To]/P[A|In]", ylab = "Significancia
(P)",
main = "Valores de significancia de la prueba chi
cuadrada\ncuando se presentan distintos radios y
frecuencias de expresión",
sub = "(los radios son aproximados)")
points(radios[radios<=1.2], P[(i1==4)&(radios<=1.2)],
type="l", col="purple")
points(radios[radios<=1.2], P[(i1==5)&(radios<=1.2)],
type="l", col="orange")
legend("topright", legend=c("100","1000","10000"),
bty="n", title="P[A|To] en transcritos por millon",
text.col=c("blue", "purple", "orange"),
title.col="black")
```

En las gráficas anteriores hemos visto como la "significancia" de una prueba de hipótesis en tablas de contingencia 2x2 depende de el tamaño de muestra (total de conteos en la tabla), las razones de las probabilidades condicionales (o "momios", "odd ratios") –entre más lejos de 1 está la razón (r), más "grande" es la significancia (menor P) y también de las frecuencias relativas de expresión de los genes (nuestras casillas con "a" y "b" datos en la tabla).

Realizando un análisis completo de todos los genes: El problema de "multiple testing"

Hemos ejemplificado aquí con solo uno de los genes del experimento de Sandra; en su experimento ella evaluó 21,588 genes en 24 diferentes genotecas –que, para este ejemplo, se clasificaron como procedentes de genotipos "Tolerantes" (**To**) o "Intolerantes" (**In**). Al realizar pruebas múltiples con los mismos datos (agrupándolos de manera diferente), tenemos el problema de que los resultados de las pruebas *no son independientes entre si*. Si nosotros queremos asegurarnos de tener un nivel de significancia α en *el total del experimento (experiment-wise)*, deberemos de hacer alguna corrección. Ejemplificaremos esto con un conjunto continuo de datos (independiente del ejemplo que hemos seguido) y después plantearemos una de las formulas generales (corrección de Bonferroni).

R tiene grandes capacidades para simular "números aleatorios" (pseudoaleatorios) procedentes de muchas distribuciones. Utilizaremos un conjunto arbitrario de 100 números procedentes de la distribución normal estándar (con media cero y varianza uno) para realizar un experimento. Notemos que cada uno de nosotros generará una muestra DIFERENTE de números, sin embargo, por regularidad estadística, las conclusiones a las que lleguemos deben de ser parecidas.

```
azar <- rnorm(100) # Genero una muestra de 100 números aleatorios de la distribución normal
summary(azar) # Estadísticas básicas
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.9460 -0.5877  0.2053  0.1143  0.8231  2.8640
sd(azar) # Desviación estándar
[1] 1.078289
hist(azar) # Obtengo un histograma de los datos
```

Ahora, si nosotros dividimos aleatoria y arbitrariamente a estos números en dos grupos de 50 datos cada uno y realizamos una prueba de hipótesis para H0: $\mu 1 = \mu 2$ (las medias de las poblaciones de las cuales procede cada grupo son iguales) contra Ha: $\mu 1 \neq \mu 2$ (son diferentes), lo "más probable" es que no rechacemos H0 (que en este caso ES CIERTA, pues todos los números proceden de la misma población).

```
i1 <- sample(c(1:100), size=50) # 50 números aleatorios muestreados del vector c(1:100)
i2 <- setdiff(c(1:100), i1) # Obtiene el RESTO de los números; i.e., los que no están en i1
sort(i1); sort(i2) # Para ver como dividimos la muestra
```

Ahora realicemos una prueba de t (t de Student) para la hipótesis H0 usando como índices los valores i1 & i2 en "azar"; esto es seleccionando dos conjuntos disjuntos y aleatorios de 50 números que existen en "azar"

```
test <- t.test(azar[i1],azar[i2]) # Prueba de t de dos grupos de 50 datos cada uno

test

     Welch Two Sample t-test
```

```
data:  azar[i1] and azar[i2]
t = -0.1998, df = 96.346, p-value = 0.8421
alternative hypothesis: true difference in means is not
equal to 0
95 percent confidence interval:
 -0.4734398  0.3868575
sample estimates:
 mean of x  mean of y
0.09262617 0.13591731
```

Ahora pensemos, si nosotros realizamos un gran número de veces el proceso anterior (escoger dos grupos disjuntos de 50 datos de "azar"), eventualmente obtendremos un resultado "significativo" con, por ejemplo, α=0.05 Más aún, "en promedio" deberemos de obtener aproximadamente 5% (P=0.05) de resultados "significativos" o "positivos". En este caso sabemos que esos casos son "falsos positivos" pues los números generados proceden de la misma distribución. Obtengamos los valores de P para 1000 repeticiones de ese proceso.

```
p.val <- c() # Aquí guardaremos el valor de P.
for( i in 1:1000) {
i1 <- sample(c(1:100), size=50) # 50 números aleatorios
muestreados del vector c(1:100)
i2 <- setdiff(c(1:100), i1) # Obtiene el RESTO de los
números; i.e., los que no están en i1
test <- t.test(azar[i1], azar[i2])
p.val <- c(p.val, test$p.value) # Capturamos el valor de P
}

summary(p.val)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
0.0001172 0.2406000 0.5049000 0.4984000 0.7386000 0.9998000

length(p.val[p.val<0.05]) # Número de elementos de p.val
menores de 0.05 (debe ser más o menos 50, es decir el 0.05
(5%) de los casos).
[1] 57
```

los resultados que ustedes obtengan serán distintos, PERO en aproximadamente 0.05 (5%) de los casos la probabilidad P será MENOR que 0.05; es decir "significativa" con α=0.05

¿Cómo protegernos de estos "falsos positivos"?, es decir ¿cómo "asegurar" que realmente los resultados son "significativos"?. La respuesta es relativamente fácil: Seamos mas estrictos. Si vamos a hacer k pruebas utilizando los mismos datos (clasificándolos de manera distinta) y queremos tener una proporción máxima de α "falsos positivos" entonces realicemos cada prueba individual con un valor $\alpha' = \alpha / k$. Esto se conoce como la "corrección de Bonferroni" para pruebas múltiples. En nuestro caso estamos realizando k = 1000 pruebas, de manera que debemos tomar $\alpha' = (0.05) / 1000 = 5e\text{-}05$ en cada prueba individual. Esto nos "asegurará" que tendremos una probabilidad "pequeña" (alrededor de 0.05) de tener falsos positivos. En nuestro caso:

```
length(p.val[p.val<5e-05]) # Número de elementos de p.val
menores de 5e-05
[1] 0
```

En el caso de Sandra, ella estará haciendo pruebas CON LOS MISMOS DATOS (las más de 14 millones de lecturas) agrupándolas de 21,588 distintas maneras (de acuerdo con a que gen mapearon), de manera que si quiere una α=0.05 deberá emplear en cada prueba individual una

$\alpha' = (0.05) / 21588 = 2.316102e\text{-}06$

Esta corrección es bastante "conservadora" o astringente –se pasa de la raya, pero en general asegura un nivel adecuado de falsos positivos.

Conclusiones y colofón

En el pasado se usaban, arbitrariamente, los valores fijos de α = 0.05 o 0.01, pues muchas de las tablas (impresas) en los libros de Estadística daban esos valores. Sin embargo ahora lo más común es que podamos calcular, exacta o aproximadamente

α = P = P[obtener un valor igual o más extremo que el encontrado en los datos dado que la hipótesis nula (H0) es cierta].

Es decir, podemos medir la "cantidad" de evidencia que nos dan los datos acerca de H0; mientras más pequeña es α, menos "creíble" (verosímil) es que H0 sea cierta y por lo tanto es más razonable rechazarla a favor de la hipótesis alternativa (Ha).

Este cálculo nos permite tomar decisiones que normarán el futuro de nuestros experimentos. Por ejemplo, si estamos "razonablemente seguros" de que un gen se expresa de manera diferencial en dos condiciones, podemos plantear otros experimentos basados en ese hecho.

Es importante remarcar que la "significancia estadística" (una α pequeña) NO implica relevancia biológica ni causalidad, sin embargo lo inverso –investigando si tal o cual fenómeno tiene relevancia (biológica o científica en general) en muchos casos implica realizar pruebas estadísticas que resulten "significativas".

Algo que no discutimos hoy es la "potencia de la prueba", P[Aceptar H0 dado que H0 es cierta], sin embargo, de lo presentado se infiere que entre mayor sea el tamaño de muestra, menores son las diferencias (o razones, r) que pueden detectarse y por lo tanto mayor es la potencia.

Comencé a escribir esto con dos propósitos: Tratar de dar con un ejemplo real una explicación del "significado de la significancia" e introducir con ejemplos la enorme potencia que tiene el lenguaje R para hacer cálculos en Probabilidad y Estadística. Me extendí mucho más de lo que pensaba, pero espero que sea de algún provecho.