

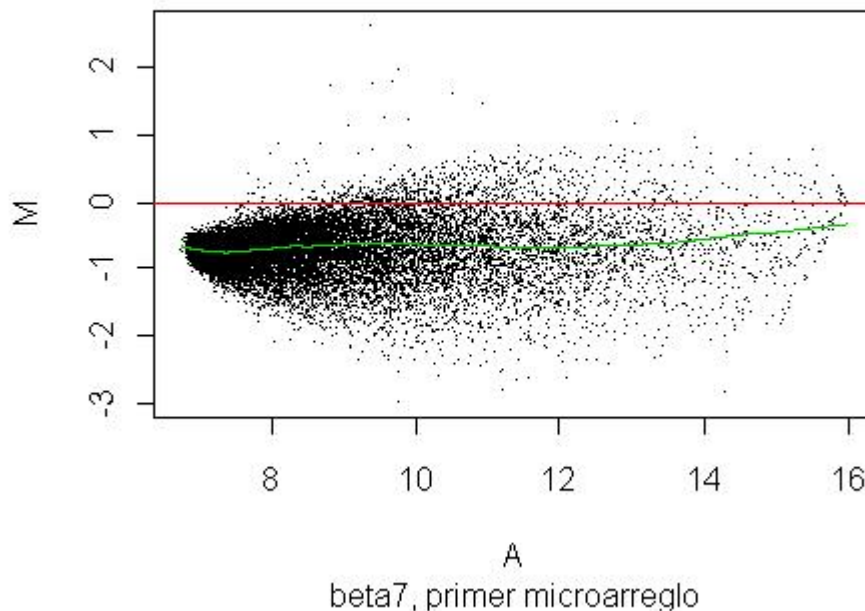
8. Normalización

El objetivo de la normalización es identificar y eliminar variaciones sistemáticas conservando la señal biológica. Las principales fuentes de estas variaciones se encuentran en: diferencias entre los fluores Cy3 y Cy5 respecto de sus eficiencias de etiquetado y propiedades de escaneado, diferencias en los parámetros de escaneado, efectos de la aguja (print-tip) y del plato. Mediante la normalización se pretende asegurar que las diferencias en intensidad realmente reflejan la expresión diferencial de los genes y que no haya sesgos artificiales debido a factores técnicos.

Recordemos la notación M-A:

$$M = \log_2 R/G$$
$$A = 0.5 \log_2 (RG),$$

Sin corrección por background

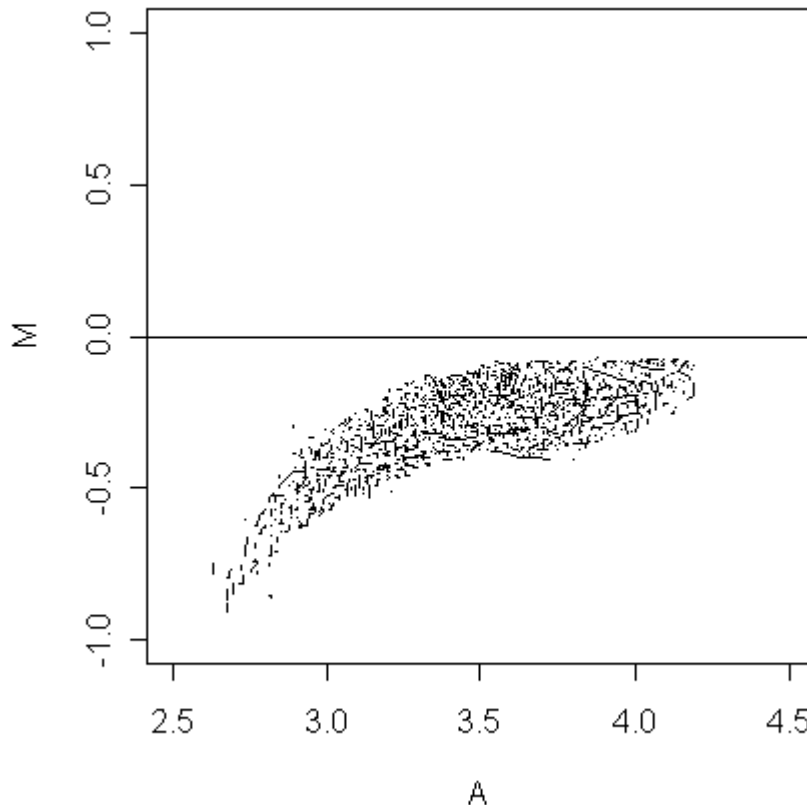


```
verde <- log2(beta7@maGf[,1])
rojo <- log2(beta7@maRf[,1])
M <- rojo-verde
A <- 0.5*(rojo +verde)
plot (A, M, pch=".")
title("Sin corrección por background")
title(sub ="beta7, primer microarreglo")
abline (h=0,col=2)
lines(lowess(A,M,f=0.10), col = 3)
```

Se espera que en general a lo sumo un 10 % de los genes se encuentren expresados en diferentes niveles entre los dos tipos de células, con beta 7 + y beta 7 -, sin embargo el

gráfico MA anterior muestra que los valores del canal verde son en su mayoría más altos que los del canal rojo.

Un MA plot de un experimento de calibración (self-self) en el cual la misma muestra ha sido hibridada al canal rojo y al canal verde de un mismo microarray, idealmente, la lectura del escaner debería ser la misma para ambos canales. En este caso se espera que la nube de puntos en un MA plot no presente tendencias y sea pareja alrededor de la recta $M=0$. Pero habitualmente los gráficos MA presentan estructuras como la que mostramos en la figura siguiente



En la figura se pone de manifiesto el sesgo del tinte hacia el canal verde Cy3 para intensidades bajas. Esto es que para intensidades bajas los valores de Cy3 son mayores de lo esperado.

Una característica para tener en cuenta es que las estructuras que muestran los gráficos MA dependen del microarreglo, esto es que, cuando un mismo lote de RNA es utilizado como muestra para hibridar distintos vidrios o chips pueden obtenerse patrones diferentes.

Consideraremos dos tipos de métodos de normalización:

- *Métodos de normalización de dos canales:* se ajustan los valores de M utilizando los valores de A y otros factores como print-tip, plato, posición espacial.

- *Métodos de normalización para cada canal por separado:* se aplican a las intensidades originales considerando los mismos factores que antes pero no A.

8.1 Normalización conjunta de dos canales

El proceso de normalización de dos canales puede separarse en dos componentes: posición ó locación (l) y escala (s). En general se estiman esas dos componentes de la distribución de los log-ratios (M) en función de las intensidades (A) de los dos canales o en función de las coordenadas espaciales del microarreglo.

Los log-ratios normalizados M_{norm} en general están dados por

$$M_{\text{norm}} = (M - l) / s$$

Los métodos difieren en cómo se obtienen l y s. Por ejemplo en la normalización global por la mediana el parámetro de localización se asume constante, es decir que es el mismo para todos los spots, mientras que en una normalización global dependiente de la intensidad (A) el parámetro de localización se supone que es una función suave de A y la función es estimada utilizando la opción robusta de la función **loess** para el suavizado en diagramas de dispersión.

La tabla siguiente muestra los procedimientos básicos de normalización para posición en microarreglos de dos canales.

Tabla: Procedimientos de normalización de localización

Nombre	Descripción
None	No se realiza normalización
Median	Normalización global por mediana
Loess	Normalización global dependiente de A utilizando el scatter plot smoother loess
Print-tip-loess	Normalización A-dependiente, dentro del grupo que determina cada aguja utilizando el scatter plot smoother loess
twoD loess	Normalización espacial, dentro del grupo que determina cada aguja, utilizando la función loess
scalePrintTipMAD	Normalización A-dependiente, dentro del grupo de cada aguja utilizando el scatter plot smoother loess, seguida de una normalización de escala dentro de cada grupo utilizando la función MAD

Dado un conjunto de números $x_1, \dots, x_n$, el MAD es la mediana de sus desvíos absolutos respecto de la mediana, es decir que si

$$m = \text{mediana}\{x_1, \dots, x_n\}$$

entonces

$$\text{MAD} = \text{mediana}\{|x_1 - m|, \dots, |x_n - m|\}.$$

La función de R para MAD es `mad`.

La función **maNorm** del paquete **marray** realiza los procedimientos de normalización descriptos en la tabla.

Del help del R tenemos

Uso:

```
maNorm(mbatch, norm=c("printTipLoess", "none", "median", "loess",  
  "twoD", "scalePrintTipMAD"), subset=TRUE, span=0.4, Mloc=TRUE,  
  Mscale=TRUE, echo=FALSE, ...)
```

Veamos los argumentos principales:

mbatch: Objeto de clase 'marrayRaw', con los datos lote de arreglos para normalizar. Un objeto de clase "'marrayNorm'" también puede ser pasado si la normalización se realiza en varios pasos.

norm: Una cadena de caracteres especificando el procedimiento. Puede especificarse mediante la primera letra de cada procedimiento: "p", "n", "m", "l", "t", "s".

subset: Un vector "lógico" o "numérico" indicando el subconjunto de puntos para calcular los valores normalizados.

span: Este argumento controla el grado de suavizado en la función loess.

Mloc: Si vale 'TRUE', los valores de la normalización de localización (l) que devuelve la función son guardados en el **slot** 'maMloc' del objeto de clase "'marrayNorm'", si vale 'FALSE', estos valores no son guardados

Mscale: Si vale 'TRUE', los valores de la normalización de escala (s) son guardados en el **slot** 'maMscale' del objeto de clase "'marrayNorm'", si vale 'FALSE', estos valores no son guardados.

echo: Si vale 'TRUE', el índice del arreglo siendo normalizado aparece en pantalla

La función **maNorm** devuelve un objeto de clase "'marrayNorm'" conteniendo los datos de intensidad normalizados.

La clase "'marrayNorm'" se utiliza para almacenar los datos pos-normalización de un lote de microarreglos de cDNA. Contiene slots para A, los log-ratios normalizados que seguiremos llamando M, y los valores de normalización de posición y escala, la geometría (layout) de los arreglos y la descripción de las muestras hibridadas y las secuencias spoteadas al arreglo.

La siguiente instrucción realiza una normalización Print-tip-loess

```
> beta7norm <- maNorm(beta7, norm= "p")
```

```
> summary(beta7norm)
```

Normalized intensity data: Object of class marrayNorm.

Call to normalization function:

```
maNormMain(mbatch = mbatch, f.loc = list(maNormLoess(x = "maA",
  y = "maM", z = "maPrintTip", w = NULL, subset = subset, span =
  span,
  ...)), Mloc = Mloc, Mscale = Mscale, echo = echo)
```

Number of arrays: 6 arrays.

A) Layout of spots on the array:

Array layout: Object of class marrayLayout.

Total number of spots: 23184
Dimensions of grid matrix: 12 rows by 4 cols
Dimensions of spot matrices: 23 rows by 21 cols

Currently working with a subset of 23184spots.

Control spots:

There are 5 types of controls :

Buffer	Empty	Negative	Positive	probes
3	1328	225	204	21424

Notes on layout:

B) Samples hybridized to the array:

Object of class marrayInfo.

	maLabels	FileNames	SubjectID	Cy3	Cy5	Date of	Blood Draw
1	6Hs.195.1.gpr	6Hs.195.1.gpr	1	b7 -	b7 +		2002.10.11
2	6Hs.168.gpr	6Hs.168.gpr	3	b7 +	b7 -		2003.01.16
3	6Hs.166.gpr	6Hs.166.gpr	4	b7 +	b7 -		2003.01.16
4	6Hs.187.1.gpr	6Hs.187.1.gpr	6	b7 -	b7 +		2002.09.16
5	6Hs.194.gpr	6Hs.194.gpr	8	b7 -	b7 +		2002.09.18
6	6Hs.243.1.gpr	6Hs.243.1.gpr	11	b7 +	b7 -		2003.01.13
	Date of Scan						
1	2003.07.25						
2	2003.08.07						
3	2003.08.07						
4	2003.07.18						
5	2003.07.25						
6	2003.08.06						

Number of labels: 6

Dimensions of maInfo matrix: 6 rows by 6 columns

Notes:

TargetBeta7.txt

C) Summary statistics for log-ratio distribution:

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

```
6Hs.195.1.gpr -6.47 -0.36 -0.01 -0.01 0.34 5.65 3415
6Hs.168.gpr -6.21 -0.50 -0.01 -0.02 0.49 5.91 2839
6Hs.166.gpr -6.68 -0.55 -0.02 -0.03 0.52 5.89 3440
6Hs.187.1.gpr -9.69 -0.28 -0.01 -0.01 0.26 5.20 2942
6Hs.194.gpr -8.22 -0.36 -0.01 -0.03 0.33 5.73 6090
6Hs.243.1.gpr -5.58 -0.36 -0.01 0.08 0.41 8.39 2227
```

D) Notes on intensity data:
GenePix Data
>

Comparemos con los resultados del

```
> summary(beta7)
```

para los log-ratios de las intensidades, de los datos sin normalizar

```
C) Summary statistics for log-ratio distribution:
      Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
6Hs.195.1.gpr -6.13 -1.00 -0.52 -0.50 -0.08 5.95 3415
6Hs.168.gpr -7.08 -0.80 -0.21 -0.23 0.34 5.19 2839
6Hs.166.gpr -7.07 -1.25 -0.64 -0.62 -0.02 6.15 3440
6Hs.187.1.gpr -9.81 -0.92 -0.60 -0.55 -0.25 5.00 2942
6Hs.194.gpr -5.93 0.00 0.44 0.53 0.90 7.74 6090
6Hs.243.1.gpr -6.38 -1.13 -0.69 -0.64 -0.21 7.05 2227
```

La normalización por localización centra los log-ratios alrededor de cero al tener en cuenta los sesgos espaciales y de intensidad. ¿Por qué?

Los siguientes son los resultados de la normalización por mediana

```
> beta7.normmm <- maNorm(beta7, norm="median")
> summary(beta7.normmm)
Normalized intensity data:      Object of class marrayNorm.
```

Call to normalization function:
maNormMain(mbatch = mbatch, f.loc = list(maNormMed(x = NULL,
y = "maM", subset = subset)), Mloc = Mloc, Mscale = Mscale,
echo = echo)

```
C) Summary statistics for log-ratio distribution:
      Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
6Hs.195.1.gpr -5.61 -0.48 0 0.02 0.44 6.47 3415
6Hs.168.gpr -6.87 -0.59 0 -0.02 0.55 5.40 2839
6Hs.166.gpr -6.43 -0.61 0 0.02 0.61 6.79 3440
6Hs.187.1.gpr -9.20 -0.32 0 0.05 0.35 5.60 2942
6Hs.194.gpr -6.38 -0.44 0 0.09 0.46 7.30 6090
6Hs.243.1.gpr -5.68 -0.44 0 0.05 0.48 7.75 2227
```

Las normalizaciones anteriores, por localización, no ajustan por diferencia en escala entre microarreglos. La función que realiza una normalización por escala es la **maNormScale** del paquete **marray**. Se recomienda evaluar la necesidad de

realizar esta normalización sobre la base de un control caso por caso. En los casos en que la diferencia en escalas es pequeña puede ser preferible realizar únicamente una normalización por localización.

El siguiente código realiza una normalización por escala

```
> beta7norm.escala <- maNormScale(beta7norm)
```

8.1.1 Normalización conjunta de dos canales. Algo más sobre la función maNorm

¿Qué devuelve la función maNorm cuando se elige “none” como método de normalización?

Por defecto toma “printTipLoess”

```
> maNorm
```

```
function (mbatch, norm = c("printTipLoess", "none", "median",  
"loess", "twoD", "scalePrintTipMAD"), subset = TRUE, span = 0.4,  
Mloc = TRUE, Mscale = TRUE, echo = FALSE, ...)  
{  
  opt <- list(...)  
  norm.method <- match.arg(norm)  
  if (echo)  
    cat(paste("Normalization method: ", norm.method, ".\n",  
      sep = ""))  
  switch(norm.method,  
    none = maNormMain(mbatch, f.loc = NULL,  
      Mloc = Mloc, Mscale = Mscale, echo = echo),  
    median = maNormMain(mbatch, f.loc = list(maNormMed(x = NULL, y = "maM",  
      subset = subset)), Mloc = Mloc, Mscale = Mscale, echo = echo),  
    loess = maNormMain(mbatch, f.loc = list(maNormLoess(x = "maA", y = "maM",  
      z = NULL, w = NULL, subset = subset, span = span, ...)), Mloc = Mloc,  
      Mscale = Mscale, echo = echo),  
    twoD = maNormMain(mbatch, f.loc = list(maNorm2D(x = "maSpotRow", y =  
      "maSpotCol", z = "maM", g = "maPrintTip", w = NULL, subset = subset,  
      span = span, ...)), Mloc = Mloc, Mscale = Mscale, echo = echo),  
    printTipLoess = maNormMain(mbatch, f.loc = list(maNormLoess(x = "maA",  
      y = "maM", z = "maPrintTip", w = NULL, subset = subset, span = span, ...)), Mloc  
      = Mloc, Mscale = Mscale, echo = echo),  
    scalePrintTipMAD = maNormMain(mbatch, f.loc = list(maNormLoess(x = "maA",  
      y = "maM", z = "maPrintTip", w = NULL, subset = subset,  
      span = span, ...)), f.scale = list(maNormMAD(x = "maPrintTip",  
      y = "maM", geo = TRUE, subset = subset)), Mloc = Mloc,  
      Mscale = Mscale, echo = echo))  
}  
<environment: namespace:marray>
```

```
> maNormMain
function (mbatch, f.loc = list(maNormLoess()), f.scale = NULL,
  a.loc = maCompNormEq(), a.scale = maCompNormEq(), Mloc = TRUE,
  Mscale = TRUE, echo = FALSE)
{
  if (is(mbatch, "marrayRaw")) {
    mnorm <- as(mbatch, "marrayNorm")
    M <- Ml <- Ms <- NULL
  }
  if (is(mbatch, "marrayNorm"))
    mnorm <- mbatch
  slot(mnorm, "maNormCall") <- match.call()
  if (length(f.loc) > 0)
    M <- Ml <- NULL
  if (length(f.scale) > 0)
    M <- Ms <- NULL
  for (i in 1:ncol(maM(mbatch))) {
    if (echo)
      cat(paste("Normalizing array ", i, ".\n", sep = ""))
    m <- mbatch[, i]
    M1 <- M2 <- NULL
    Mnorm <- maM(m)
    if (length(f.loc) > 0) {
      for (func in f.loc) M1 <- cbind(M1, func(m))
      if (length(f.loc) > 1)
        M1 <- rowSums(M1 * a.loc(maA(m), length(f.loc)))
      Ml <- cbind(Ml, M1)
      Mnorm <- (Mnorm - M1)
    }
    if (length(f.scale) > 0) {
      m <- mnorm[, i]
      slot(m, "maM") <- Mnorm
      for (func in f.scale) M2 <- cbind(M2, func(m))
      if (length(f.scale) > 1)
        M2 <- rowSums(M2 * a.scale(maA(m), length(f.scale)))
      Ms <- cbind(Ms, M2)
      Mnorm <- Mnorm/M2
    }
    M <- cbind(M, Mnorm)
  }
  slot(mnorm, "maM") <- M
  if (length(f.loc) > 0 & Mloc)
    slot(mnorm, "maMloc") <- Ml
  if (length(f.scale) > 0 & Mscale)
    slot(mnorm, "maMscale") <- Ms
  return(mnorm)
}
```



```
}
<environment: namespace:marray>
>
```

Cuando se aplica `maNorm` a un objeto de clase “`maRaw`” y se elige “`none`” como método de normalización se obtienen los mismos datos transformados en un objeto “`maNorm`”:

```
> slotNames(beta7)
[1] "maRf"      "maGf"      "maRb"      "maGb"      "maW"
"maLayout"
[7] "maGnames"  "maTargets" "maNotes"

> beta7MA <- as(beta7, "marrayNorm")
> slotNames(beta7MA)
[1] "maA"      "maM"      "maMloc"    "maMscale"  "maW"
[6] "maLayout"  "maGnames"  "maTargets" "maNotes"
"maNormCall"
> beta7MA@maMscale
<0 x 0 matrix>
> beta7MA@maMloc
<0 x 0 matrix>
```

Recordemos que los log-ratios normalizados M_{norm} están dados por

$$M_{\text{norm}} = (M - 1) / s$$

En el slot `beta7MA@maMscale` se guarda s y en `beta7MA@maMloc` se guarda l . Como no se realizó ninguna Normalización en `beta7MA` esos slots no tienen datos.

Realicemos ahora la normalización por defecto

```
> beta7Norm <- maNorm(beta7)
```

Veamos qué hace

```
> beta7MA@maM[1:2,1:3]                                     # sin Normalizar
      6Hs.195.1.gpr 6Hs.168.gpr 6Hs.166.gpr
[1,]   -0.44651573   0.5737352        NA
[2,]    0.06926266   0.5555187  -0.6453351
> beta7Norm@maMloc[1:2,1:3]                                  # ahora sí @maMloc
tiene datos
      [,1]      [,2]      [,3]
[1,] 0.005572376 0.4561687        NA
[2,] 0.183875526 0.5264531 -0.1057688

> beta7Norm@maM[1:2,1:3]                                     # Normalizados
      6Hs.195.1.gpr 6Hs.168.gpr 6Hs.166.gpr
[1,]   -0.4520881   0.11756657        NA
[2,]   -0.1146129   0.02906567  -0.5395663
*****

> beta7MA@maM[1:2,1:3] - beta7Norm@maMloc[1:2,1:3]
```

```

      6Hs.195.1.gpr 6Hs.168.gpr 6Hs.166.gpr
[1,]      -0.4520881  0.11756657      NA
[2,]      -0.1146129  0.02906567  -0.5395663
> beta7Norm@maM[1:2,1:3]
      6Hs.195.1.gpr 6Hs.168.gpr 6Hs.166.gpr
[1,]      -0.4520881  0.11756657      NA
[2,]      -0.1146129  0.02906567  -0.5395663

```

Los valores M normalizados resultan de restar $(M - 1)$. La normalización anterior no incluye una corrección por escala:

```

> beta7Norm@maMscale
<0 x 0 matrix>

```

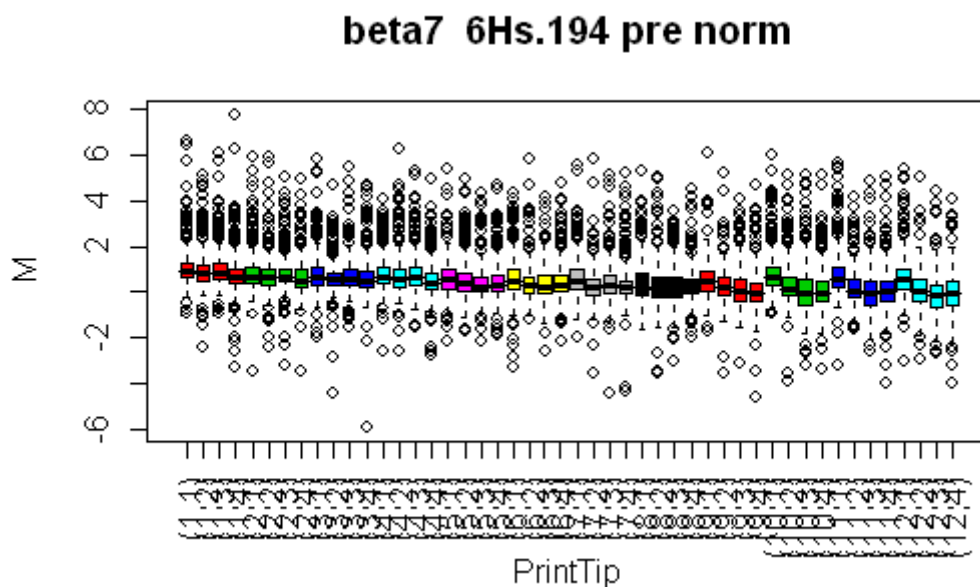
8.2 Visualización de los resultados de la normalización

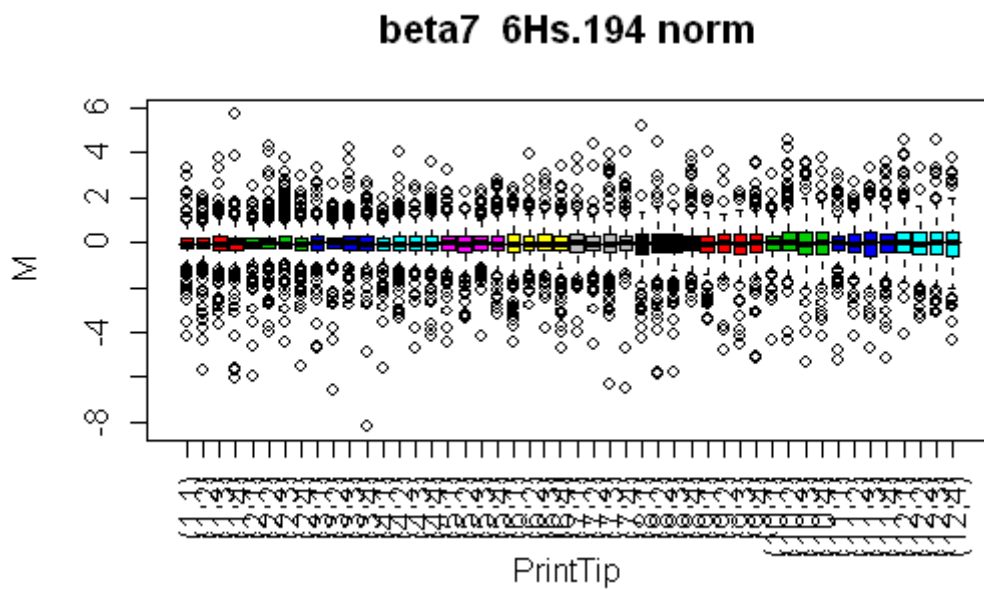
Comparamos los datos pre y pos normalización

```

> boxplot(beta7[, 5], xvar = "maPrintTip", yvar =
"maM",main="beta7 6Hs.194 pre norm")

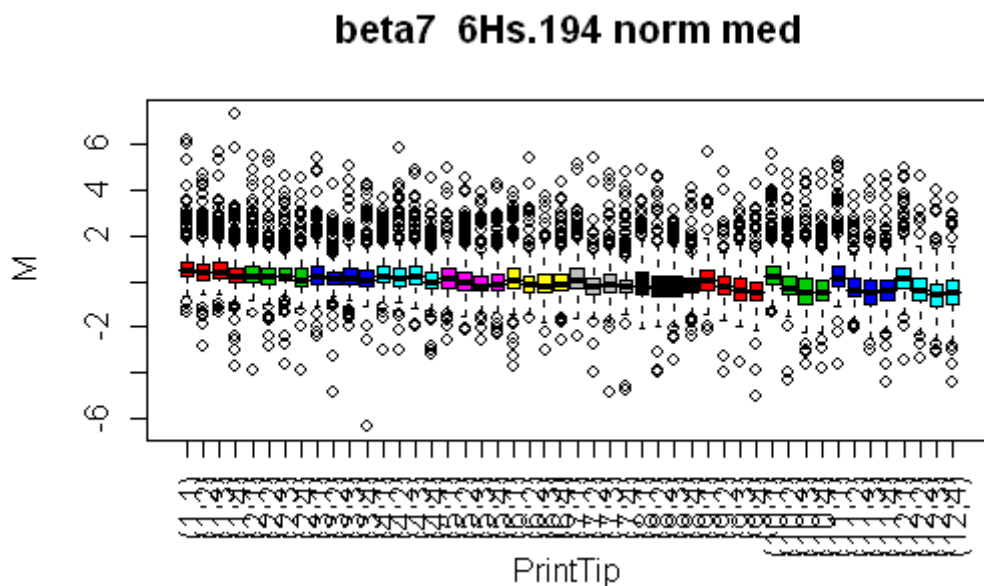
```





```
> boxplot(beta7.norm[, 5], xvar = "maPrintTip", yvar =  
"maM",main="beta7 6Hs.194 norm")
```

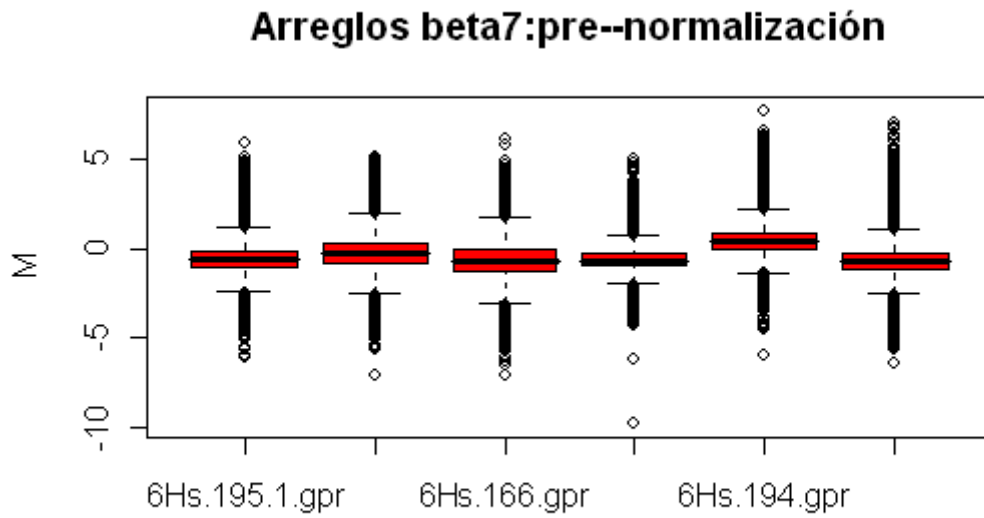
La normalización ha eliminado el sesgo por aguja.



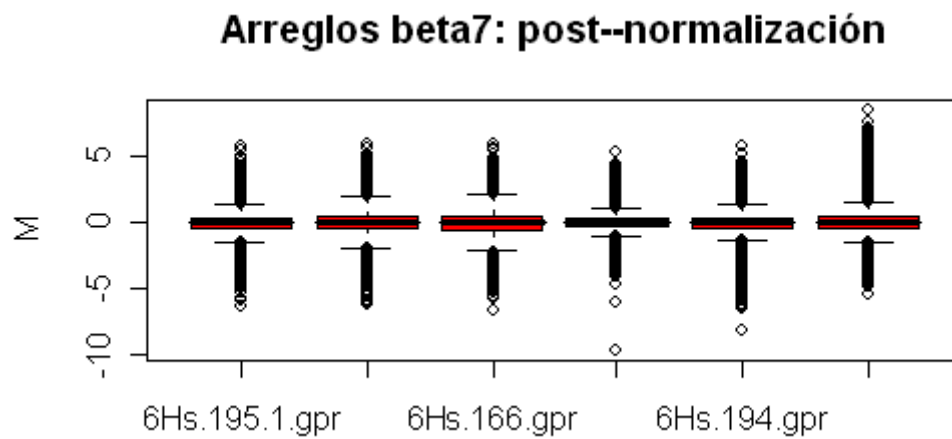
La normalización global por mediana no ha eliminado el sesgo por aguja.

Veamos como se modifican las intensidades entre arreglos

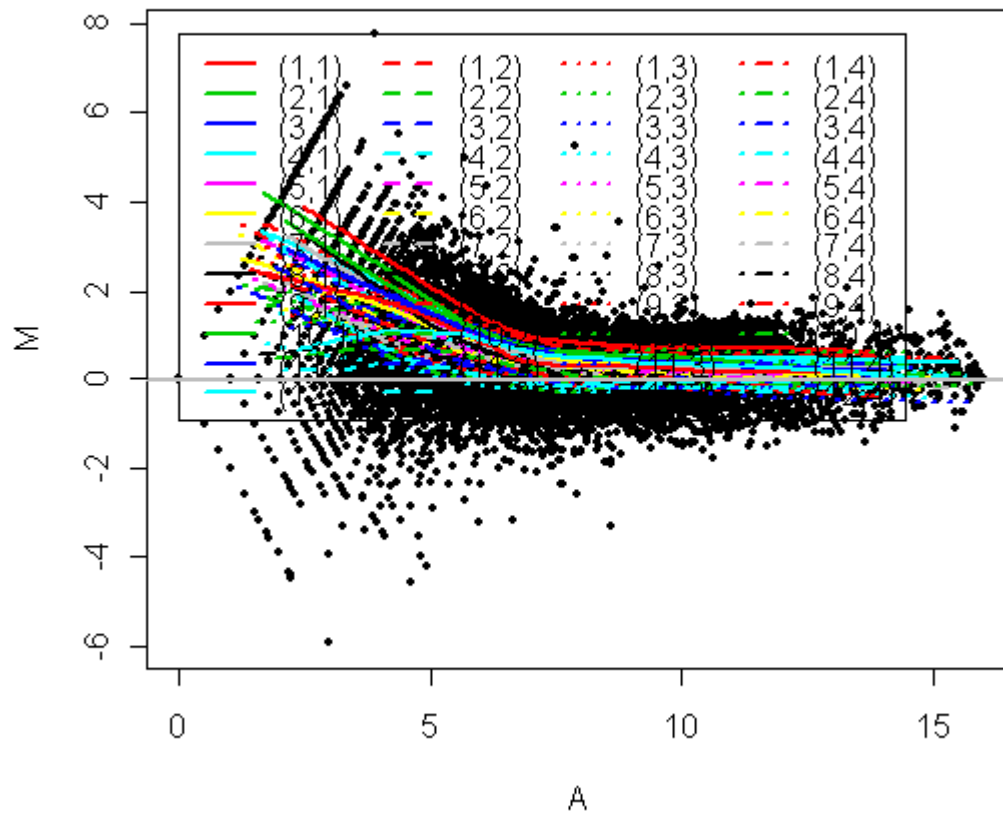
```
> boxplot(beta7, yvar = "maM", main = "Arreglos beta7:pre--normalización")
```



```
> boxplot(beta7.norm, yvar = "maM", main = "Arreglos beta7: post--normalización")
```

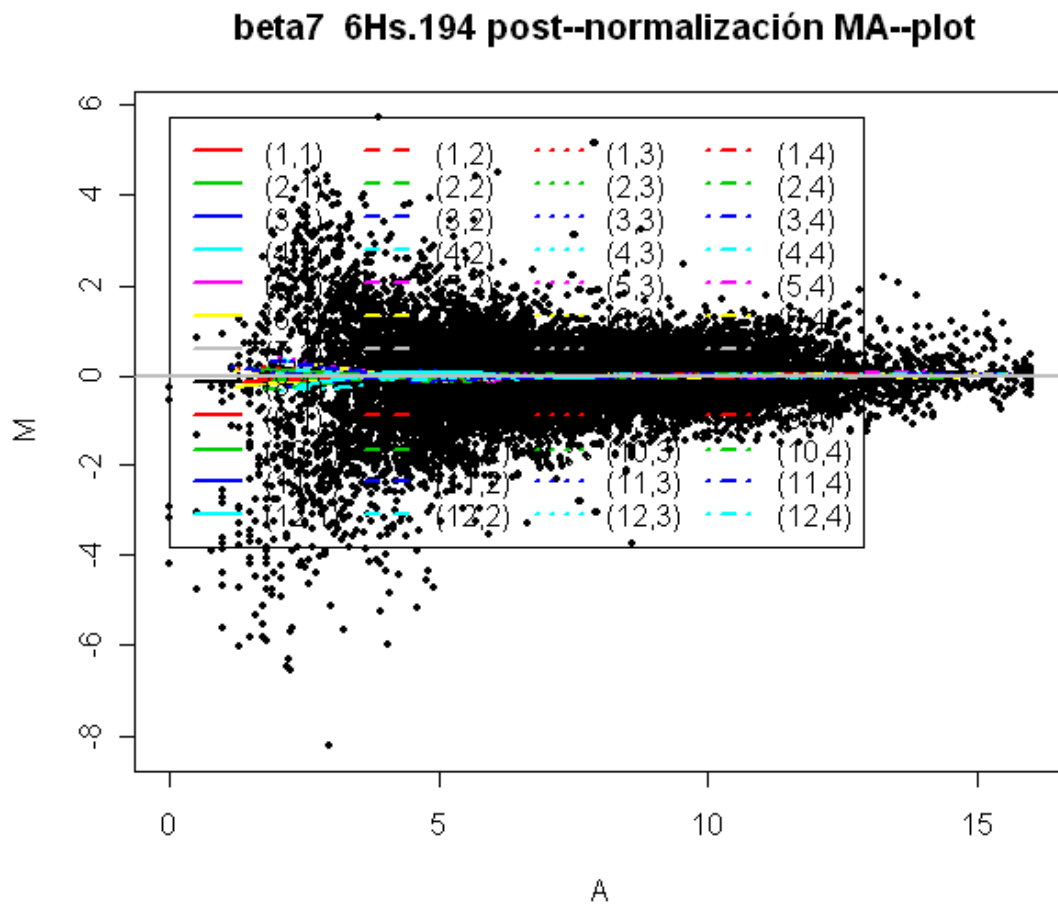


beta7 6Hs.194 pre-normalización MA--plot



```
> plot(beta7[, 5], main = "beta7 6Hs.194 pre--  
normalización MA--plot")
```

Se observa la tendencia de M en función de A, global y para cada print-tip. Restar el background aumenta muchísimo la variabilidad en las intensidades menores.

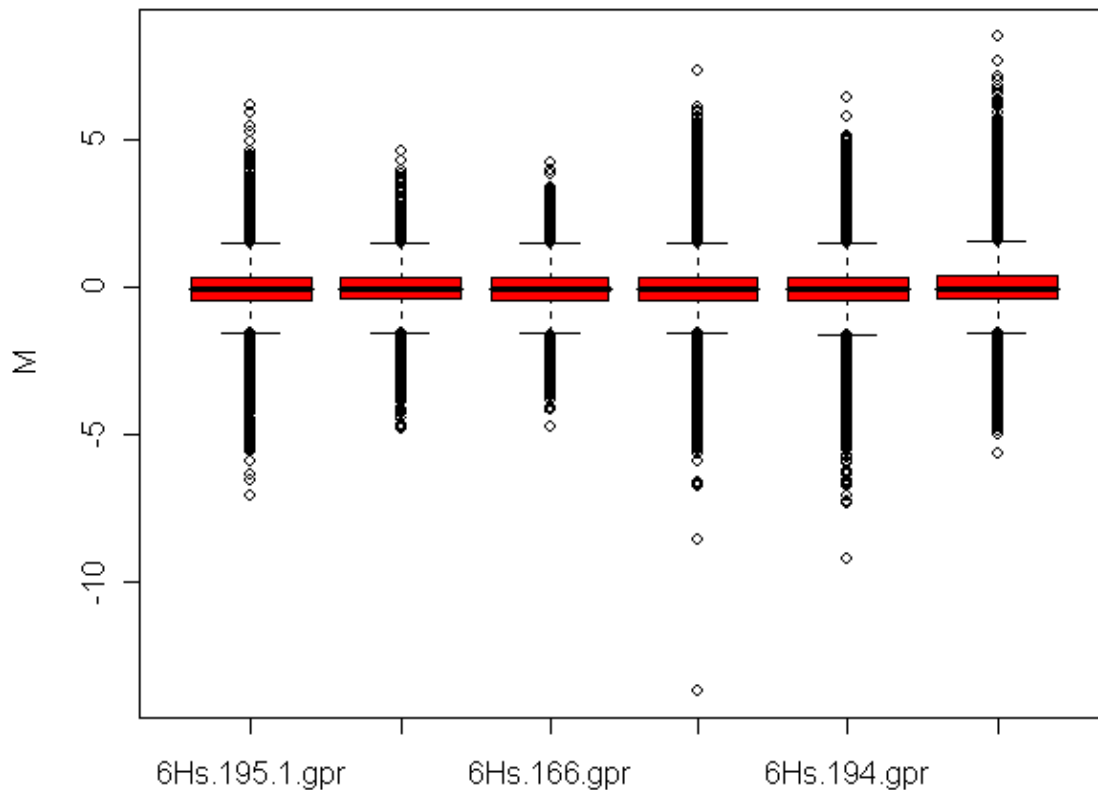


Se ha eliminado la tendencia de M en función de A, tanto globalmente como para cada print-tip pero la variabilidad en los valores menores de A sigue siendo muy alta.

Veamos el efecto de la normalización por escala

```
> boxplot(beta7norm.escala, yvar = "maM", main = "Arreglos
beta7: post--normalización con escala")
```

Arreglos beta7: post--normalización con escala



Referencias

Cleveland, W.S. (1979) "Robust Locally Weighted Regression and Smoothing Scatterplots," *Journal of the American Statistical Association*, Vol. 74, pp. 829-836.

Cleveland, W.S. and Devlin, S.J. (1988) "Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting," *Journal of the American Statistical Association*, Vol. 83, pp. 596-610.

Cleveland, W. and Loader, C. (1995) Smoothing by Local Regression: Principles and Methods (with discussion)[Cleveland, W. and Loader, C. \(1995\): Computational Statistics](#)

Maronna, R.A. , Martin, R.D. and Yohai, V.J. (2006). *Robust Statistics: Theory and Methods*, Wiley, West Sussex.

Yuanyuan Xiao, Mark R. Segal, and Yee Hwa Yang, "Stepwise Normalization of Two-Channel Spotted Microarrays" (November 5, 2004). *Center for Bioinformatics & Molecular Biostatistics*. Paper stepnorm5.
<http://repositories.cdlib.org/cbmb/stepnorm5>

Viñetas de Normalización del marray

8.3 Normalización - Continuación

Hemos visto que la normalización es un proceso para identificar y eliminar errores sistemáticos que no se deban a diferencias reales entre el RNA de los distintos grupos de tejidos a comparar.

- Las fuentes de estos errores sistemáticos se pueden deber a una o más de las siguientes razones
 - o Diferente eficiencia en la incorporación de tintes
 - o Diferencias en la cantidad de mRNA
 - o Diferencias en los parámetros de escaneo
 - o Diferencias entre los grupos de cada aguja
 - o Efectos espaciales
 - o Efectos debido al plato
- Estas variaciones sistemáticas se reflejan en la dependencia de los cocientes en:
 - o La intensidad total de fluorescencia (A)
 - o La heterogeneidad espacial
 - o La aguja
 - o El plato

Existen diferentes propuestas respecto de qué genes utilizar en las normalizaciones.

- Todos los genes del arreglo.
- Genes cuya expresión es constante.
- Controles agregados -spiked in-. Por ejemplo genes de plantas.
- Serie de titulaciones de DNA genómico (Genomic DNA titration series).
- Un conjunto con rango invariante (Rank invariant set).

¿Cuáles utilizar? Depende del tipo de experimento. ¿Qué tipo de propuesta utilizaría para los siguientes experimentos?

- Se comparan muestras en las cuales la mayoría de los genes muestran distinto grado de diferenciación (por ejemplo, distintas etapas de crecimiento de la célula)
- Se comparan muestras en las cuales sólo una fracción pequeña de los genes está alterada (por ejemplo, pérdida de la actividad de un gen en un camino metabólico alterado)

8.3.1 Tipos de normalizaciones.

8.3.1.1 Notación

Cuando la normalización está basada en los

- log-ratios ($\log_2 R/G$) nos referimos a
 - o *normalización de 2 canales*

Cuando la normalización está basada en cada una de

- $\log_2 R$, $\log_2 G$ nos referimos a
 - o *normalización de un canal* (canales por separado)

8.3.1.2 Objetivos

Los tipos de métodos de normalización a utilizar dependen del tipo de preguntas que se desea responder.

- 1) Determinar los **niveles de expresión relativos** de cada gen entre dos tipos de muestras (tejido sano vs. tejido patológico)
- 2) Determinar **cuáles son los genes que se han expresado** en cada una de las muestras o en diferentes tiempos .

El objetivo principal de la *normalización de dos canales* es el ajuste de sesgos *dentro* de cada arreglo. Es necesario para responder al primer problema.

Para responder al segundo problema es necesario realizar una normalización de los dos canales por separado. Además las normalizaciones de un canal son necesarias para que el análisis basado en microarreglos de un canal sea una opción válida para los investigadores.

El siguiente ejemplo muestra la necesidad de realizar un análisis de los canales por separado y por lo tanto también su normalización, en microarrays de dos canales.

Consideremos un ejemplo en el que se comparan animales jóvenes y viejos para los mutantes y wild-type

Nombre del Archivo	Cy3	Cy5
Archivo1	wt.joven	wt.viejo
Archivo2	wt.viejo	wt.joven
Archivo3	mu.joven	mu.viejo
Archivo4	mu.viejo	mu.joven

Cada arreglo de este experimento realiza una comparación directa entre el RNA de joven-viejo. Pero no hay arreglos que comparen mutantes con wild-type. Este es un

ejemplo de diseño no conectado en el cual no hay arreglos ligando mutantes con wild-type.

No es posible realizar una comparación entre mutantes y wild-type utilizando log-ratios que han sido normalizados, únicamente. Es necesario analizar las intensidades de los canales rojo y verde por separado. Esto es analizar $\log(R)$ y $\log(G)$ en vez de log-ratios.

8.3.2 Métodos de normalización

Dos canales juntos - sobre un M-A plot

Ya hemos visto cómo realizar las siguientes normalizaciones de dos canales utilizando el paquete `marray`

- Global: Mediana
- Local, dependientes A: Loess -Print-tip

Un canal

- ANOVA (Kerr et al 2000, Wolfinger et al 2001)
- Quantile normalization (Bolstad et al 2003)
- VSN (Huber et al 2002, Durbin et al 2002, 2003)

8.4 Normalización de los canales por separado

El paquete `marray` no realiza este tipo de normalizaciones, utilizaremos el paquete **limma**.

Recordemos que **beta7norm** contiene las intensidades normalizadas por Print-tip-loess

```
> beta7norm <- maNorm(beta7, norm= "p")
```

La función `normalizeBetweenArrays` está incluida en el paquete `limma`, y toma **cada color como un microarreglo separado**

```
> normalizeBetweenArrays(beta7)
Error en normalizeBetweenArrays(beta7) : object doesn't
appear to be RGList or MAList object
```

Para seguir con los procedimientos de **limma** tendremos que transformar los datos de **marray** crudos o normalizados mediante la función `as(, )`. Necesitamos bajar el paquete “convert”:

<http://bioconductor.org/packages/release/bioc/html/convert.html>

en R

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("convert")
Using R version 2.10.1, biocinstall version 2.5.10.
Installing Bioconductor version 2.5 packages:
[1] "convert"
Please wait...
```

```
probando la URL
'http://www.bioconductor.org/packages/2.5/bioc/bin/windows/
contrib/2.10/convert_1.22.0.zip'
Content type 'application/zip' length 59933 bytes (58 Kb)
URL abierta
downloaded 58 Kb
```

```
package 'convert' successfully unpacked and MD5 sums
checked
```

```
The downloaded packages are in
      C:\Documents and Settings\x\Configuración
local\Temp\RtmpPXhCVj\downloaded_packages
```

```
> library(convert)#para usar la función as(,)
```

```
#datos sin normalizar
> beta7.l <- as(beta7,"RGList")
```

```
> summary(beta7.l)
      Length Class      Mode
R      139104 -none-    numeric
G      139104 -none-    numeric
Rb     139104 -none-    numeric
Gb     139104 -none-    numeric
weights 139104 -none-    numeric
printer      4 -none-    list
genes         6 data.frame list
targets        7 data.frame list
notes         1 -none-    character
```

```
> beta7.l$R[1:2,]
      6Hs.195.1.gpr 6Hs.168.gpr 6Hs.166.gpr 6Hs.187.1.gpr 6Hs.194.gpr 6Hs.243.1.gpr
[1,]          227          370          139          69          187          241
[2,]          208          273          181          80          213          241
```

```
#datos normalizados
> beta7.p <- as(beta7norm,"MAList")
```

```
> summary(beta7.p)
```

	Length	Class	Mode
A	139104	-none-	numeric
M	139104	-none-	numeric
weights	139104	-none-	numeric
printer	4	-none-	list
genes	6	data.frame	list
targets	7	data.frame	list
notes	1	-none-	character
maNormCall	6	-none-	call

```
> beta7.p$R[1:2,]
NULL
```

```
> beta7.p$M[1:2,]
      6Hs.195.1.gpr 6Hs.168.gpr 6Hs.166.gpr 6Hs.187.1.gpr 6Hs.194.gpr 6Hs.243.1.gpr
[1,] -0.4520881  0.11756657      NA      NA      NA      5.294000
[2,] -0.1146129  0.02906567 -0.5395663 -0.9326132      NA      1.286584
```

Observación: Recordemos que por defecto los M y A's se calculan restando el Background. Vemos varios NA's. La función **backgroundCorrect()** del paquete **limma** implementa otras opciones.

8.4.1 Normalización por cuantiles-Quantile Normalization

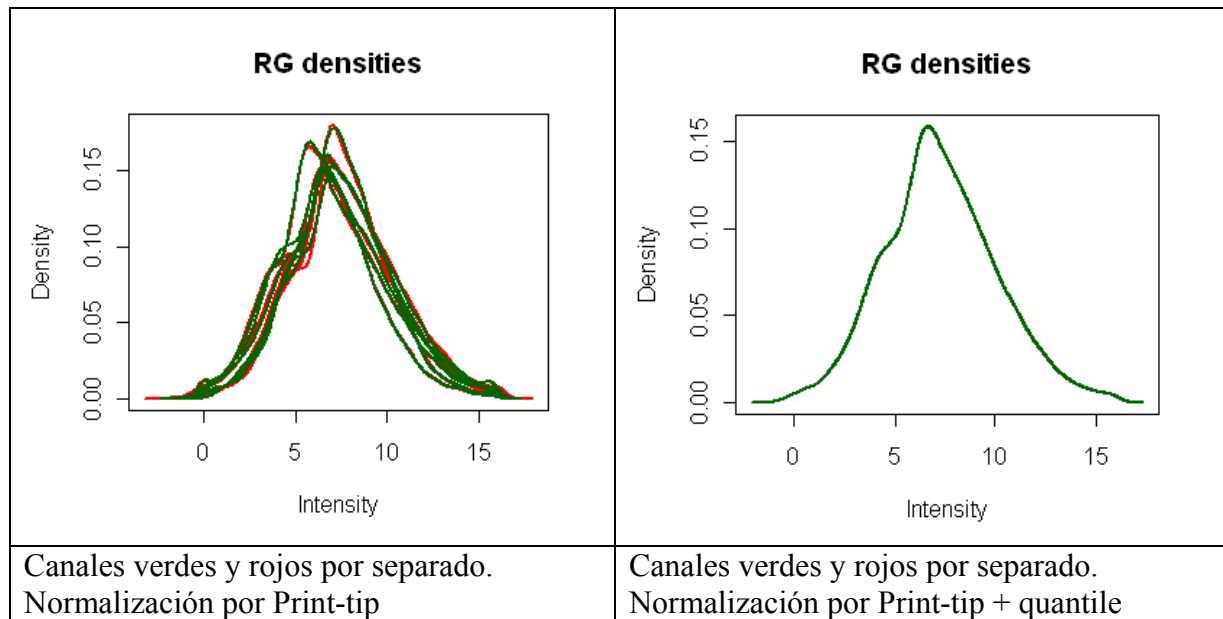
Un enfoque para realizar normalizaciones para cada canal está basado en las funciones **normalizeWithinArrays** y **normalizeBetweenArrays** del paquete **limma**. La primera de las funciones provee una normalización *dentro de cada arreglo* similar a la del marray y la otra *entre arreglos* permite que cada canal tenga una distribución similar entre los arreglos.

```
> beta7.pq <-
  normalizeBetweenArrays(beta7.p, method="quantile")
```

Mediante el método de normalización por cuantiles, las distribuciones de las intensidades de los canales rojo y verde de todos los arreglos quedan idénticas. Mediante

```
> plotDensities(beta7.p)
> plotDensities(beta7.pq)
```

obtenemos las densidades de las intensidades de los canales rojo y verde antes y después de la normalización por cuantiles.



¿En que consiste la normalización por cuantiles?

En transformar los valores de las intensidades en cada canal y en cada muestra, sin cambiarles el orden, de manera que resulten con idéntica distribución.

Se ordenan las intensidades cada uno de los microarreglos-color en orden creciente. Se toma el promedio o mediana por fila. Se reemplaza la media ó la mediana en todas las columnas y se vuelve al orden original. De esta manera el valor máximo es el mismo en todos los arreglos-color,, el valor mínimo es el mismo.

Ejemplo:

Intensidades dentro de cada arreglo-color

Gen	1R	Gen	1G	Gen	2R	Gen	2G
1	8	1	12	1	3	1	1
2	0,5	2	4	2	5	2	2
3	1,5	3	2	3	4	3	0,5
4	14	4	16	4	18	4	20
5	30	5	40	5	17	5	122
6	6	6	5	6	2	6	0
7	5	7	3	7	1	7	18

Intensidades con orden creciente dentro de cada arreglo-color

Gen	1R	Gen	1G	Gen	2R	Gen	2G	Medianas por fila Microarreglo Falso
2	0,5	3	2	7	1	6	0	0,75
3	1,5	7	3	6	2	3	0,5	1,75
7	5	2	4	1	3	1	1	3,5
6	6	6	5	3	4	2	2	4,5
1	8	1	12	2	5	7	18	10
4	14	4	16	5	17	4	20	16,5
5	30	5	40	4	18	5	122	35

Valores Normalizados

Gen	1R	Gen	1G	Gen	2R	Gen	2G	Falso Microarreglo
2	0,75	3	0,75	7	0,75	6	0,75	0,75
3	1,75	7	1,75	6	1,75	3	1,75	1,75
7	3,5	2	3,5	1	3,5	1	3,5	3,5
6	4,5	6	4,5	3	4,5	2	4,5	4,5
1	10	1	10	2	10	7	10	10
4	16,5	4	16,5	5	16,5	4	16,5	16,5
5	35	5	35	4	35	5	35	35

Valores Normalizados con su orden original

Gen	1R	Gen	1G	Gen	2R	Gen	2G
1	10	1	10	1	3,5	1	3,5
2	0,75	2	3,5	2	10	2	4,5
3	1,75	3	0,75	3	4,5	3	1,75
4	16,5	4	16,5	4	35	4	16,5
5	35	5	35	5	16,5	5	35
6	4,5	6	4,5	6	1,75	6	0,75
7	3,5	7	1,75	7	0,75	7	10

Las distribuciones de las intensidades de los microarreglos-color normalizados por cuantiles son idénticas.

8.4.2 Transformaciones estabilizadoras de la varianza

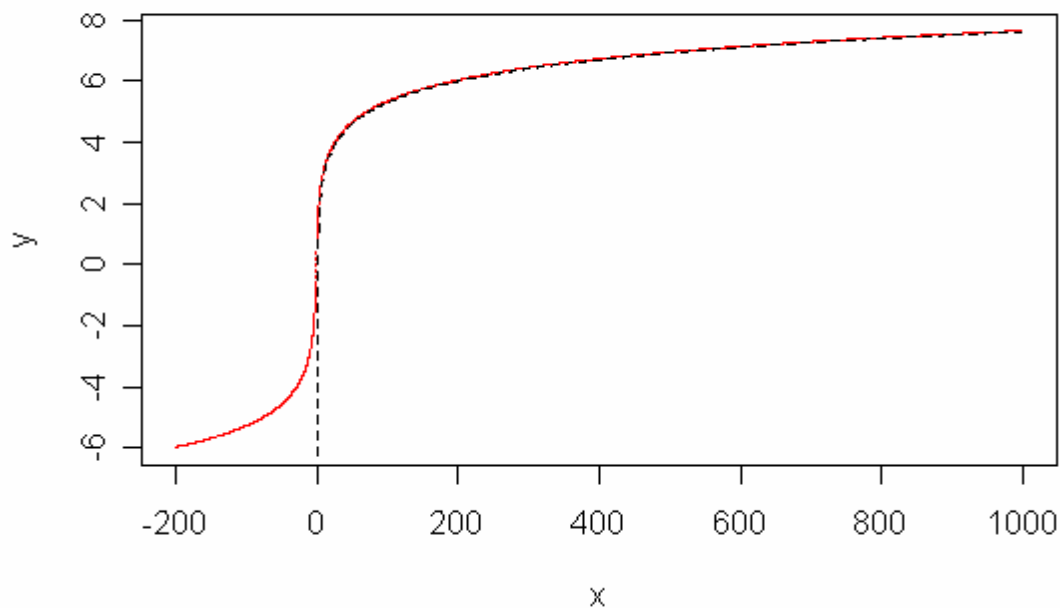
Otro enfoque de normalización de canales por separado es el “*variance stabilizing transformation*” **vs****sn**. Es una opción de la función **normalizeBetweenArrays**. Consiste en ajustar a los datos dados en la matriz x_{ki} (k filas, i columnas -arrays-) la siguiente transformación normalizadora

$$x_{ki} \rightarrow h(x_{ki}) = g \log\left(\frac{x_{ki} - a_i}{b_i}\right)$$

donde $g \log(x) = \log(x + \sqrt{x^2 + 1})$ es llamado logaritmo generalizado: es una función similar al logaritmo natural para valores grandes (comparados con el ruido del background) pero es menos empinada para valores pequeños (ver figura) .

$$g \log(x) \approx \log(x) + \log(2)$$

```
> x <- seq(-200,1000,0.1)
> y <- log(x+ sqrt(x*x+1))
> plot(x,y, pch='.',col="red")
> z <- seq(0.001,1000)
> w <- log(z)+log(2)
> lines(z,w, lty=2)
```



Las diferencias entre los valores transformados son los log-ratios generalizados. Son estimadores comprimidos (shrinkage estimators) del logaritmo natural del cociente de las intensidades (fold change).

Los **log-ratios generalizados** pueden interpretarse como estimadores shrinkage por que **siempre son menores que los log-ratios puros** (naive), la igualdad se alcanza en forma asintótica a medida que aumentan las intensidades. Tienen la ventaja de no sufrir una divergencia de la varianza como le ocurre a los log-ratios naive en intensidades bajas y se mantienen bien definidos y con significado estadístico cuando los datos se acercan a cero e incluso son negativos.

La transformación **vsn** también es una opción de la función **normalizeBetweenArrays**, se aplica a los datos crudos:

```
> beta7.vsn<-
normalizeBetweenArrays(as(beta7,"RGList"),method="vsn")
Loading required package: vsn
vsn2: 23184 x 12 matrix (1 stratum). Please use
'meanSdPlot' to verify the fit.
```

```
> 23184 * 12
[1] 278208
> 23184 * 12/2
[1] 139104
> summary(beta7.vsn)
```

Length	Class	Mode
--------	-------	------

Rb	139104	-none-	numeric
Gb	139104	-none-	numeric
weights	139104	-none-	numeric
printer	4	-none-	list
genes	6	data.frame	list
targets	7	data.frame	list
notes	1	-none-	character
M	139104	-none-	numeric
A	139104	-none-	numeric
preprocessing	3	-none-	list

8.4.3 Justificación de las transformaciones estabilizadoras de la varianza

8.4.3.1 Modelo aditivo multiplicativo

Consideramos el modelo aditivo multiplicativo (Durbin et al 2002) dado por:

$$Y = \alpha + \beta e^{\eta} + \nu$$

donde Y es el nivel de expresión de un gen en un microarray-color, α es la media del background, β es la intensidad verdadera, η y ν son variables aleatorias simétricas independientes con media 0 y varianza σ_{η}^2 σ_{ν}^2 respectivamente.

Para niveles de expresión bajos ($\beta \approx 0$) predomina el tercer término el modelo por encima del segundo, por lo tanto las intensidades siguen aproximadamente el siguiente modelo aditivo:

$$Y \approx \alpha + \nu$$

o sea que las intensidades tienen aproximadamente una distribución simétrica con media α y varianza constante igual a σ_{ν}^2 .

Para intensidades altas domina el segundo término del modelo, por lo tanto siguen el modelo multiplicativo:

$$Y \approx \beta e^{\eta}$$

y la varianza resulta aproximadamente $\beta^2 \text{Var}(e^{\eta})$

Por lo tanto para intensidades altas, se satisface aproximadamente el modelo aditivo en escala logarítmica:

$$\log(Y) = \log(\beta) + \eta$$

Los comportamientos descriptos, para intensidades altas y bajas de la relación media varianza fueron verificados en datos reales (por ejemplo en Durbin et al 2002)

8.4.3.2 Dependencia media varianza en el modelo aditivo multiplicativo

Consideremos nuevamente el modelo aditivo multiplicativo $Y = \alpha + \beta e^\eta + \nu$.

Notación:

$$\begin{aligned} m_\eta &= E(e^\eta); \\ S_\eta^2 &= \text{Var}(e^\eta); \\ S_\nu^2 &= \text{Var}(\nu) \end{aligned}$$

Luego

$$E(Y) = \alpha + \beta m_\eta \quad (1)$$

$$\text{Var}(Y) = S_\eta^2 \beta^2 + S_\nu^2 \quad (2) \quad \text{pues } \eta \text{ y } \nu \text{ son variables aleatorias independientes}$$

Esta expresión pone nuevamente (como vimos en la sección anterior) de manifiesto que para valores pequeños de β la varianza es aproximadamente constante y para valores grandes depende de la intensidad verdadera β en forma cuadrática.

$$\text{De (1)} \quad \beta = \frac{E(Y) - \alpha}{m_\eta}$$

luego en (2) queda explícita la relación media varianza:

$$\text{Var}(Y) = S_\eta^2 \left(\frac{E(Y) - \alpha}{m_\eta} \right)^2 + S_\nu^2$$

8.4.3.3 Estabilización de la relación media varianza.

Supongamos que $h: \mathbb{R} \rightarrow \mathbb{R}$, es una transformación tal que

$\text{Var}(h(Y)) \approx \text{cte}$, i.e. no depende de su media. ¿Cómo es h ?

Mediante un desarrollo de Taylor hasta grado 2 de h alrededor de $\mu = E(Y)$ obtenemos:

$$h(x) = h(\mu) + h'(\mu)(x - \mu) + 1/2 h''(\xi)(x - \mu)^2,$$

$$\text{con } x \leq \xi \leq \mu \quad \text{si } x \leq \mu \quad \text{ó } \mu \leq \xi \leq x \quad \text{si } \mu \leq x$$

Luego

$$\text{Var}(h(Y)) = h'(\mu)^2 \text{Var}(Y) + \text{Var}(1/2 h''(\xi)(Y - \mu)^2) + h'(\mu) h''(\xi) E((Y - \mu)^3)$$

Si los valores de la variable Y no están “demasiado” alejados de μ y h' y h'' son pequeños podemos considerar la siguiente expresión aproximada para la varianza:

$$\text{Var}(h(Y)) \approx h'(\mu)^2 \text{Var}(Y) = \text{cte. Podemos suponer que cte}=1.$$

Vimos que $\text{Var}(Y) = g(\mu)$ por lo tanto $h'(\mu)^2 g(\mu) = 1$ y

$$h(y) = \int \frac{1}{\sqrt{g(\mu)}} d\mu$$

$$g(\mu) = (\mu - \alpha)^2 + c$$

luego una solución de la integral es

$$h(y) = \ln(y - \alpha + \sqrt{(y - \alpha)^2 + c})$$

Referencias:

Archer, K.J., Dumur, C.I. and Ramakrishnan, V. (2004) Graphical technique for identifying a monotonic variance stabilizing transformation for absolute gene intensity signals. *BMC Bioinformatics*, 5:60.

Bengtsson H, Hössjer O. (2006) Methodological study of affine transformations of gene expression data with proposed robust non-parametric multi-dimensional normalization method. *BMC Bioinformatics*. 2006; 7: 100. Published online 2006 March 1. doi: 10.1186/1471-2105-7-100.

Bolstad, B (2001) *Probe Level Quantile Normalization of High Density Oligonucleotide Array Data*. Unpublished manuscript

Bolstad, B. M., Irizarry R. A., Astrand, M, and Speed, T. P. (2003) *A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Bias and Variance*. *Bioinformatics* 19(2) , pp 185-193.

<http://www.stat.berkeley.edu/~bolstad/normalize/normalize.html>

Cui, Xiangqin; Kerr, M. Kathleen; and Churchill, Gary A. (2003) "Transformations for cDNA Microarray Data," *Statistical Applications in Genetics and Molecular Biology*: Vol. 2 : Iss. 1, Article 4.

Available at: <http://www.bepress.com/sagmb/vol2/iss1/art4>

Durbin, B.P., Hardin, J.S., Hawkins, D.M. and Rocke, D.M. (2002) A variance stabilizing transformation for gene expression microarray data. *Bioinformatics*, **18**, S105–S110.

Durbin, B.P. and Rocke, D.M. (2003) Estimation of transformation parameters for microarray data, *Bioinformatics*, **19**, 1360–1367.

Emerson JD: **Mathematical Aspects of Transformation**. *Understanding Robust and Exploratory Data Analysis* . Edited by Hoaglin DC, Mosteller F, Tukey JW. New York: John Wiley & Sons;; 1987:247-282.

Kerr, M. K., Martin, M. & Churchill, G. A. (2000). Analysis of variance for gene expression microarray data. *Journal of Computational Biology*, 7(6):819-37.

Kerr,M.K., Martin,M. and Churchill,G.A. (2001). Statistical design and the analysis of gene expression microarray data. *Genet. Res.*, 77, 123--128.

Kerr MK, Afshari CA, Bennett L, Bushel B, Martinez J, Walker NJ, Churchill GA (2002). Statistical analysis of a gene expression microarray experiment with replication. *Stat. Sinica* 12:203– 217

Holder, D., Raubertas, R.F., Pikounis, V.B., Svetnik, V. and Soper, K. (2001) Statistical analysis of high density oligonucleotide arrays: a SAFER approach. GeneLogic Workshop on low level analysis of Affymetrix GeneChip data, Nov. 19, Bethesda, Maryland. [http://128.32.135.2/users/terry/zarray/Affy/GL Workshop/SAFERv04.pdf](http://128.32.135.2/users/terry/zarray/Affy/GL%20Workshop/SAFERv04.pdf)

Hoyle DC, Rattray M, Jupp R, Brass A (2002). Making sense of microarray data distributions. *Bioinformatics* 18:576–584

Huber, W., Von Heydebreck, A., Sultmann, H., Poustka, A. and Vingron, M. (2002) Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, **18**, S96-S104

Huber, W., Von Heydebreck, A., Sultmann, H., Poustka, A. and Vingron, M. (2003) Parameter estimation for the calibration and variance stabilization of microarray data. *Statist. App. Gen. Mol. Biol.*, **2**, Issue 1, Article 3.
<http://www.bepress.com/sagmb/vol2/iss1/art3>

Munson, P. (2001) A "consistency" test for determining the significance of gene expression changes on replicate samples and two-convenient variance-stabilizing transformations. GeneLogic Workshop on Low Level Analysis of Affymetrix GeneChip Data, Nov. 19, Bethesda, Maryland.

David M. Rocke and Blythe Durbin. A model for measurement error for gene expression analysis. *Journal of Computational Biology*, 8:557–569, 2001.

T. Speed - Editor. "Statistical Analysis of Gene Expression Data". 2003 . Chapman&Hall

Yang YH, Dudoit S, Luu P, Lin DM, Peng V, Ngai J, Speed TP (2002). Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res.* 30:e15

Wolfinger RD, Gibson G, Wolfinger ED, Bennett L, Hamadeh H, Bushel P, Afshari C, Paules RS (2001) Assessing gene significance from cDNA microarray expression data via mixed models. *J. Comp. Biol* 8: 625-637