Principios de Estadística

Leonardo Collado Torres y María Gutiérrez Arcelus
Licenciatura en Ciencias Genómicas, UNAM
www.lcg.unam.mx/~lcollado/index.php
www.lcg.unam.mx/~mgutierr/index.php

Cuernavaca, México
Febrero - Junio, 2009

Sesión práctica con small RNAs

Intro

- Lo que hoy vamos a ver viene del siguiente artículo:
  - *Sorting of Small RNAs into Arabidopsis Argonaute Complexes Is Directed by the 5' Terminal Nucleotide*
- En dicho artículo encuentran que AGO reconoce información de la secuencia en el extremo 5', y vamos a usar datos de ellos.
- En sí, la información suplementaria está disponible aquí o pueden encontrar el artículo desde NCBI.
- Por ahora, pueden ver el PDF en la página de cursos.

Datos para R

- Ya les arreglé las tablas suplementarias 1 y 2 para poder usarlas en `R`.
  - La 1 ahora es `rnas.csv` que está disponible en datos.
  - La 2 ahora es `mirnas.csv` está en el mismo folder.
- En sí, si alguien gusta, le recomiendo que compare los archivos Excel originales con los .csv que les proporciono. Sobre todo por las notas.

A trabajar!

- Bueno, a trabajar. Por favor apóyense en el código disponible sobre esta clase. Pronto entregarán el código, tal que lo podamos correr sin ningún problema.
- Les recuerdo, usen:
  - Emacs; en un buffer tengan su script y en el otro buffer abran `R`.
  - `R` en Windows usando un editor de textos. Por ejemplo, el que ya viene con `R`. Tendrán que usar copy paste seguido.
  - El comando savehistory. Sin embargo, el output de este lo tendrán que depurar.
- Intenten comentar todas las líneas de código nuevas que usen. Pues luego será parte de la tarea.

Creando `rnas`

- Primero vamos a imitar la figura 2.b de la parte "Total". Pero para eso necesitamos datos. Lean el archivo `rnas.csv`
- ¿Alguien sabe que es un csv y como se lee?
- Una vez que tengan el objeto rnas, chequenlo. Usen `head` o `tail` por ejemplo.

Filtrando rnas

- Bueno, este `data.frame` tiene información que por ahora no nos interesa. Pues despliega información por miRNAs, tasiRNAs, etc.
- Vamos a filtrar los datos y crear el objeto rnas2. Hay que quedarnos con solo las categorías generales y no las subcategorías.

```
> rnas2 <- rnas[c(1, 5, 14, 17, 18,
+     21, 26, 27, 28, 29), ]
```

Las.... bueno, pie

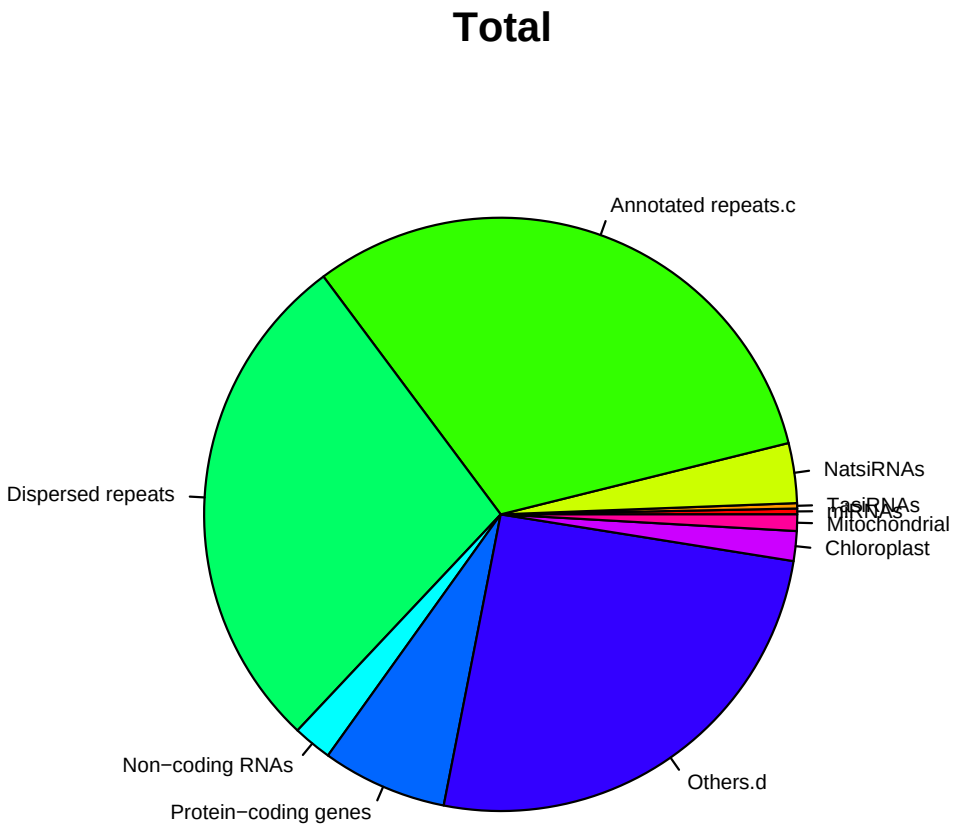
- Son tal vez las gráficas más sencillas de hacer y probablemente las más odiadas.

```
> `?`(pie)
```

- Solo chequen la ayuda de pie. Verán que dice: *Pie charts are a very bad way of displaying information. The eye is good at judging linear measures and bad at judging relative areas. A bar chart or dot chart is a preferable way of displaying this type of data.*
- Bueno, hagamos nuestra gráfica de pie de la sección de "Total".

```
> pie(rnas2[, 1], labels = rownames(rnas2),
+     col = rainbow(10), cex = 0.6,
+     main = "Total")
```

Pie de "Total"


Total
Annotated repeats.c
NatsiRNAs
TasiRNAs
miRNAs
Mitochondrial
Chloroplast
Others.d
Protein−coding genes
Non−coding RNAs
Dispersed repeats

Logramos imitar una :)

- La gráfica es MUY parecida a la del artículo. En él, juntan a los cloroplastos y mitocondrias en una categoría. Además, usan otros colores.
- En la nuestra tenemos problemas para leer los nombres, así que mejor usamos la función legend.

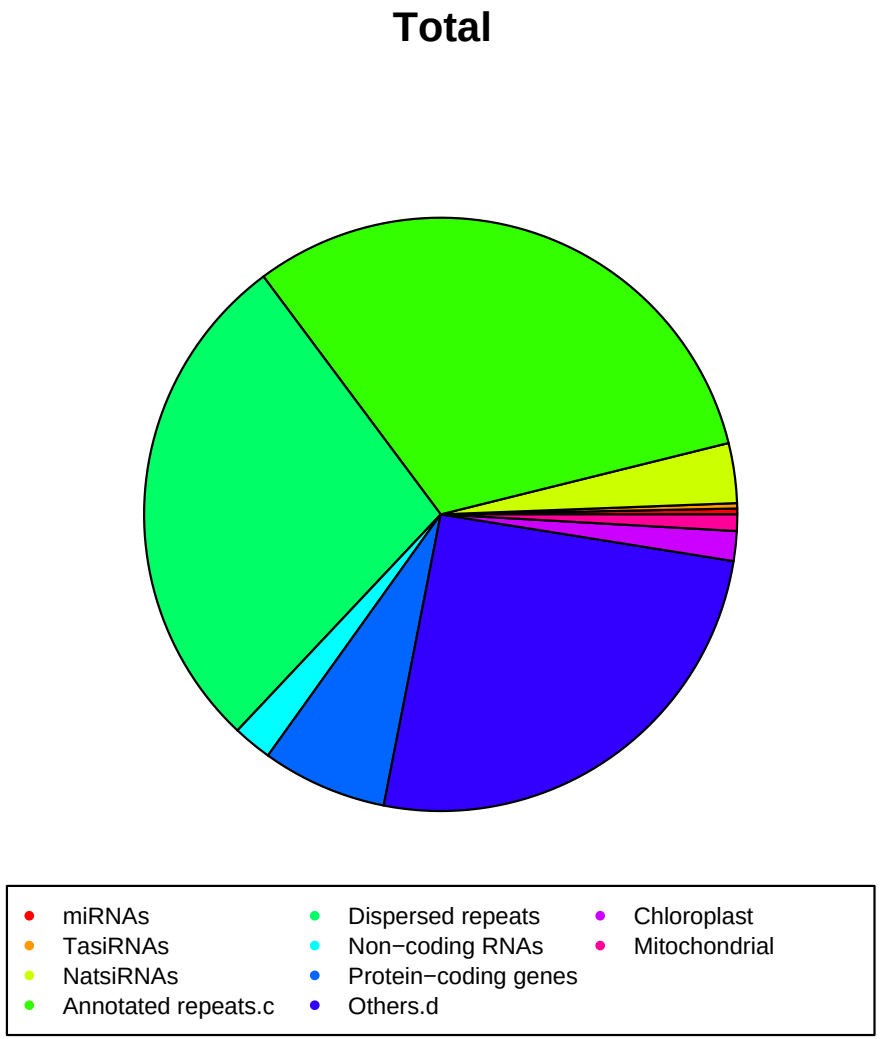
```
> pie(rnas2[, 1], labels = NA, col = rainbow(10),
+     main = "Total")
> legend("bottom", rownames(rnas2),
+     col = rainbow(10), xpd = T,
+     inset = -0.15, pch = 20, cex = 0.7,
+     ncol = 3)
```

- Si se sienten perdidos con los argumentos de la función `legend`, entenderán rápido si checan la ayuda de dicha función.

Logramos imitar una :)

- Ahora ya saben hacer el resto de las gráficas de la figura 2.b de artículo :)

Pie 2 de "Total"



Creando mirnas

- Ahora imitaremos a una gráfica de barras que usan en la figura 3.c Al igual que antes, necesitamos DATA!!!
- Lean el archivo `mirnas.csv` como lo hicimos antes con `rnas.csv`.

```
> mirnas <- read.csv("http://www.lcg.unam.mx/~lcol
+      header = T, row.names = 1)
```

- Acuerdense de explorar el objeto `mirnas`.

EL truco de los factor

- Ahora, quiero que me encuentren para Total, AGO1, 2, 4 y 5 que porcentaje de las secuencias que encuentran empiezan con U, A, C y G.
- No es taaaaaaaaaaan complicado wuahahaha :P. Primero les voy a enseñar un truco con datos de clase `factor`.

```
> head(unclass(mirnas[, 1]))
```

```
[1] 4 3 4 3 4 3
```

- Corran el anterior sin el `head`. Ahora vean que pasa si lo acoplo a un `which`. Igual, vuelvan a hacerlo sin `head`.

```
> head(which(unclass(mirnas[, 1]) ==
+     1))
```

```
[1]  49  80  82 120 122 123
```

Obtiendo nuestra matriz resultado

- Ahora deberían entender esto:

```
> res <- NULL
> for (j in 2:6) {
+     temp <- NULL
+     for (i in 1:4) {
+         temp <- c(temp, sum(mirnas[which(unclass(mirnas[,
+             1]) == i), j]))
+     }
+     res <- cbind(res, temp/sum(temp))
+ }
```

Agregandole nombres

- Bueno, ahora simplemente le agrego nombres a nuestra matriz `res` usando listas.

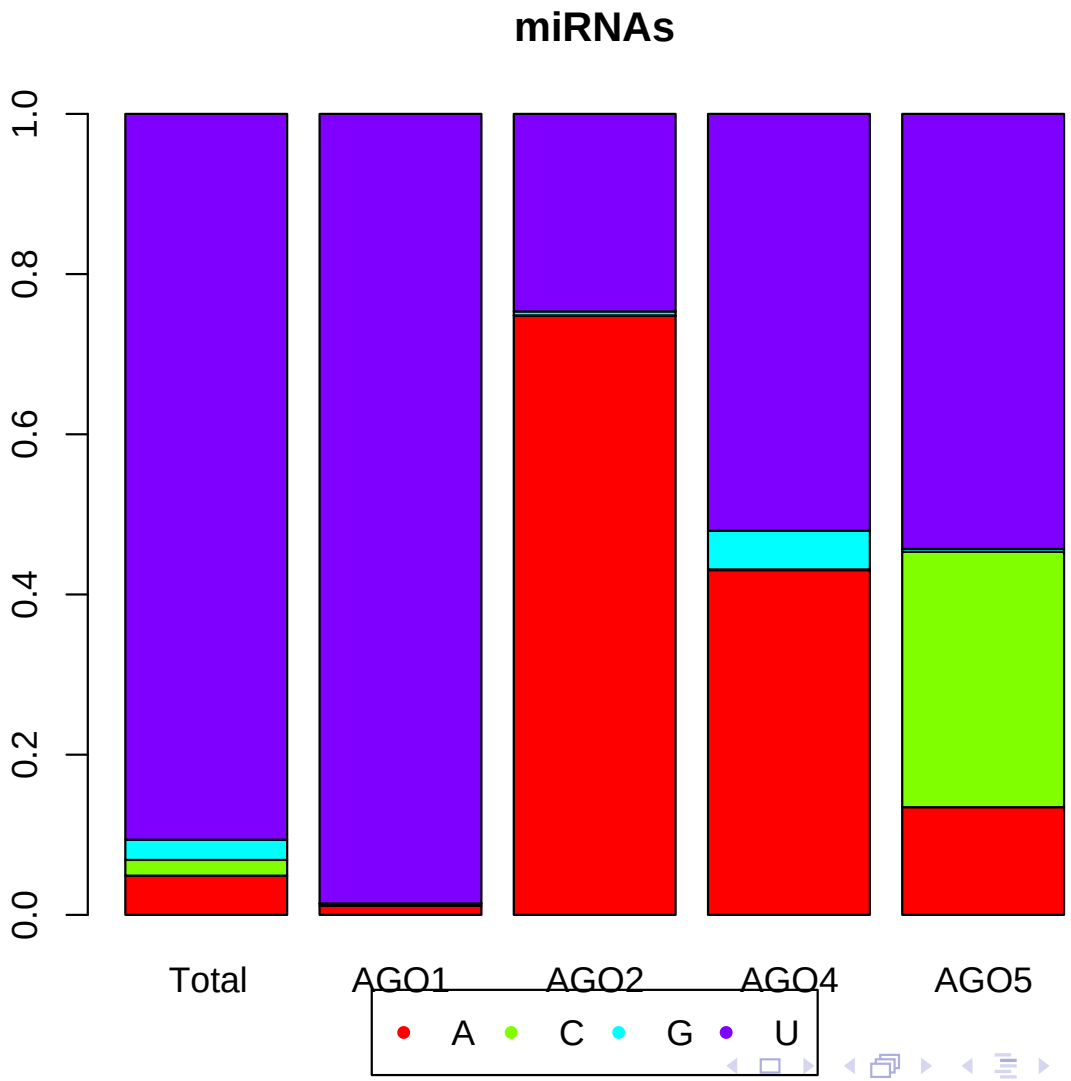
```
> dimnames(res)[[1]] <- c("A", "C",
+     "G", "U")
> dimnames(res)[[2]] <- colnames(mirnas)[2:6]
```

Barplot

- Bien, ahora simplemente usaremos barplot para hacer nuestra gráfica de barras.

```
> barplot(res, ylim = c(0, 1), col = rainbow(4),
+     main = "miRNAs")
> legend("bottom", dimnames(res)[[1]],
+     pch = 20, col = rainbow(4),
+     inset = -0.2, ncol = 4, xpd = T)
```

Barras para miRNAs



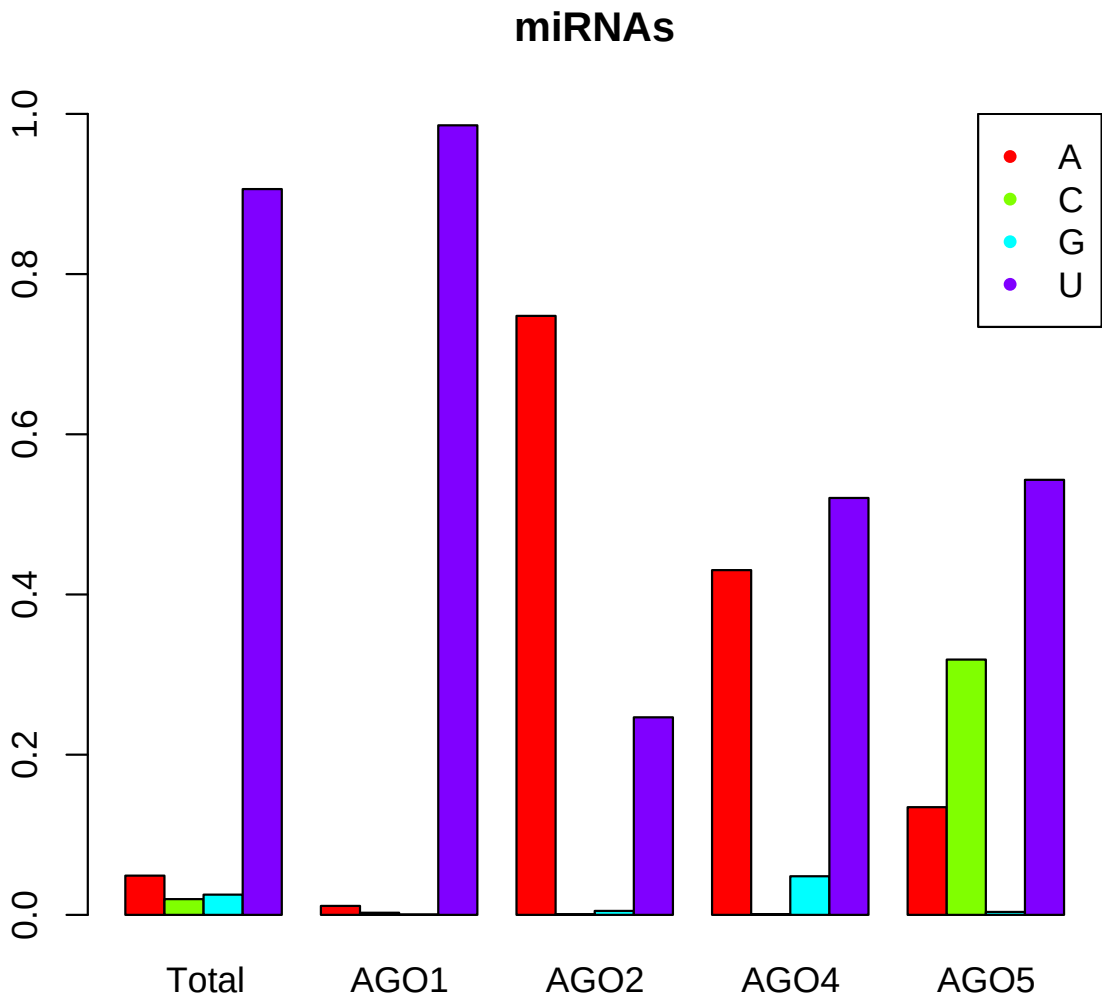
w00t

- ¿Qué tal eh? Se parece bastante a la gráfica 3.c parte de miRNAs, aunque los datos difieren un poco por lo cual no son idénticas.
- No es la mejor gráfica... pero es mejor que la siguiente.

```
> barplot(res, beside = T, ylim = c(0,
+     1), col = rainbow(4), main = "miRNAs")
> legend("topright", dimnames(res)[[1]],
+     pch = 20, col = rainbow(4))
```

- ¿Cúal es la diferencia primordial entre las dos gráficas a nivel de código?

Barras 2 para miRNAs



Level up!

- En nuestro taller de RNAs, María y yo estuvimos de acuerdo en que ese no es el mejor tipo de gráfica.
- Es tiempo de que puedan usar el poderío de `R` tal como se los mostramos en la primera clase :)

  ```
  > `?`(mosaicplot)
  ```

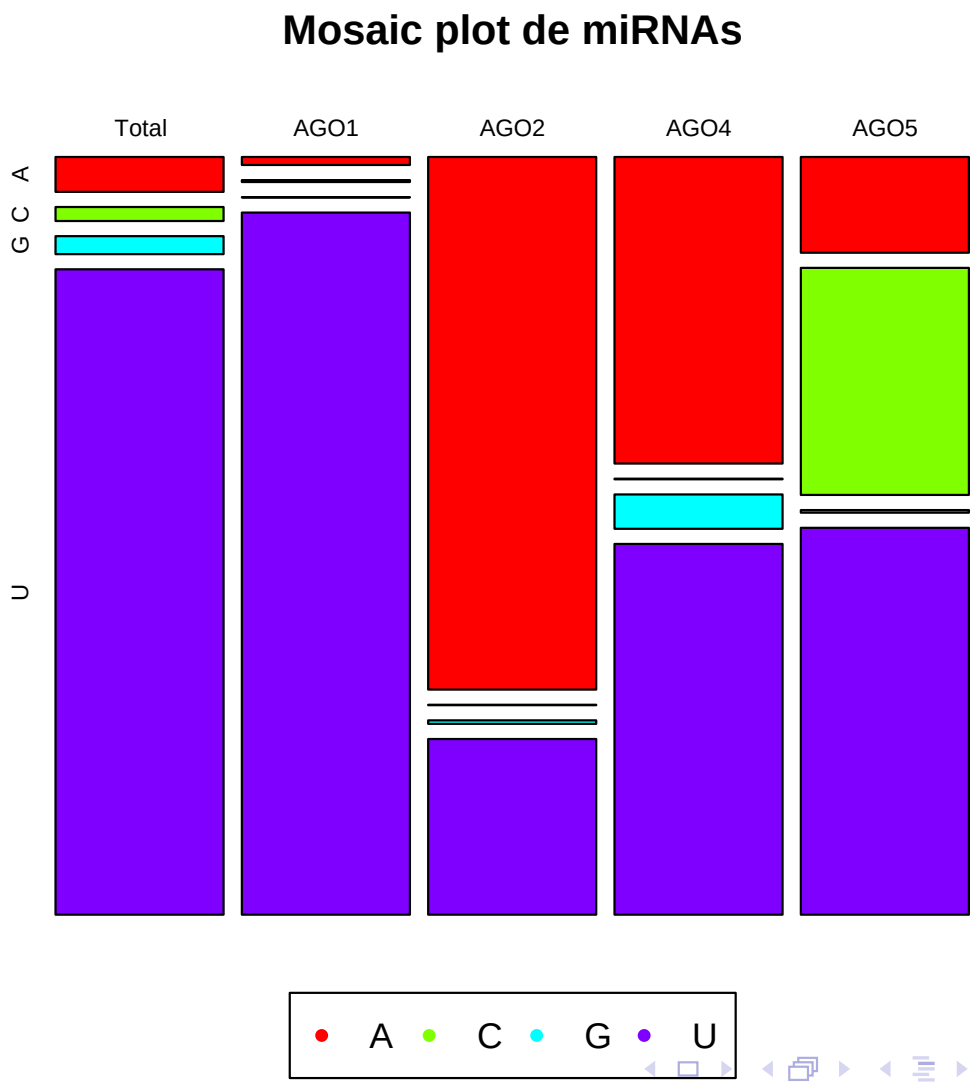
- ¿Complicado, verdad? Bueno, les enseñaré un ejemplo que pueden repetir felizmente.

Empezando..

- Usemos nuestra matriz `res` que creamos antes para la gráfica de barra. Para que funcione como queremos, tenemos que usar la transversa de nuestra matriz, por eso el t().

```
> mosaicplot(t(res), main = "Mosaic plot de miRNA
+      col = rainbow(4))
> legend("bottom", dimnames(res)[[1]],
+      pch = 20, col = rainbow(4),
+      inset = -0.2, ncol = 4, xpd = T)
```

Mosaico de miRNAs



Creando res2

- En realidad, es nuestra misma gráfica de barras pero con espacios extras. Lo que pasa es que nuestros datos en `res` están ya en porcentajes (valores de 0 a 1) así que no nos sirven.

```
> colSums(res)
```

```
Total  AG01  AG02  AG04  AG05
    1     1     1     1     1
```

- Repitamos la creación de la matriz `res` pero manteniendo valores absolutos.

Creando res2


```
> res2 <- NULL
> for (j in 3:6) {
+     temp <- NULL
+     for (i in 1:4) {
+         temp <- c(temp, sum(mirnas[which(unclass(mirnas[,
+             1]) == i), j]))
+     }
+     res2 <- cbind(res2, temp)
+ }
> dimnames(res2)[[1]] <- c("A", "C",
+     "G", "U")
> dimnames(res2)[[2]] <- colnames(mirnas)[3:6]
> colSums(res2)
```

```
   AGO1    AGO2    AGO4    AGO5
1299561  128686   20186   30660
```

Y Waldo?

- ¿Alguien nota lo que cambie?

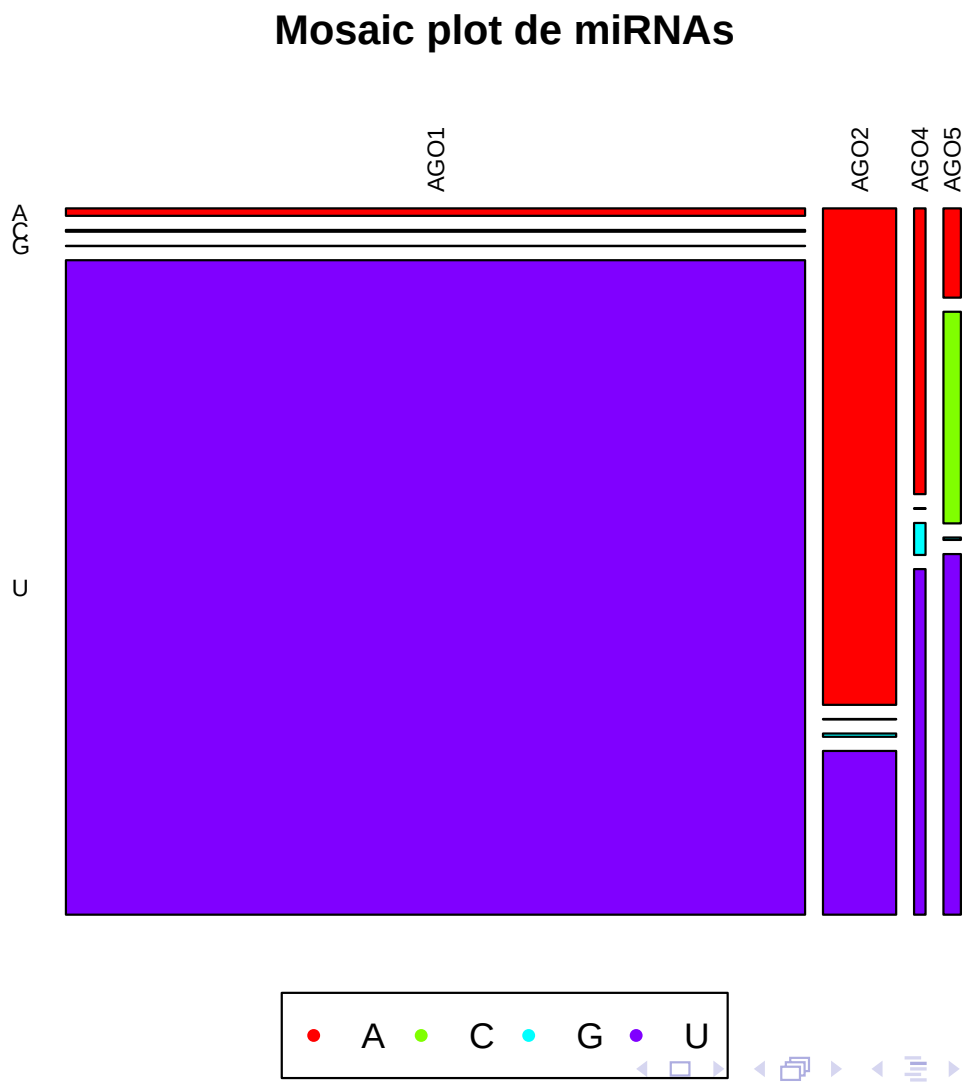
```
> ncol(res) == ncol(res2)

[1] FALSE
```

Nuestro nuevo mosaicplot

- *Indeed*, ya no me interesan los datos de "Total".
- Ahora si repitamos nuestra gráfica de mosaico. Solo le voy a cambiar un poco como se imprime el texto usando el argumento `las`.

```
> mosaicplot(t(res2), main = "Mosaic plot de miRN
+     col = rainbow(4), las = 2)
> legend("bottom", dimnames(res)[[1]],
+     pch = 20, col = rainbow(4),
+     inset = -0.2, ncol = 4, xpd = T)
```

Mosaico 2 de miRNAs



Concluyendo

- Esa gráfica dice mucho más que la que usaron en el artículo. Como ven, la mayoría de los miRNAs están asociados con AGO1, y de estos, la mayoría tienen una U en el extremo 5'.

Tarea :P

- Ahora la parte fuera de clase....
- Averiguen como usar la función pdf y sobre todo con el argumento `onefile`.
- Tienen hasta la mañana del miércoles para subir a Cursos su script de `R` que genere solo un archivo PDF como output. Este archivo PDF debe contener 7 gráficas.
  - Las 5 gráficas de pie de la figura 2.b
  - Las 2 importantes que hicimos en clase para la figura 3.c (la de barras y la de mosaico).
- Como ven, su tarea está prácticamente hecha por nosotros :) Claro, faltan los comentarios y que le cambien el nombre del archivo al homólogo de `lcollado.R` ;)