



# INFORMÁTICA

## Grado en Biotecnología - Curso 2018/2019

### Enseñanzas Prácticas y de Desarrollo

### PRACTICA FINAL DE CURSO

**IMPORTANTE:** La entrega del trabajo se hará a través de la actividad correspondiente en el Aula Virtual (<https://campusvirtual.upo.es/>) antes de las 14:00 horas del LUNES 17 de DICIEMBRE de 2018. Pasada esta hora la aplicación no permitirá la entrega del trabajo, quedando el ejercicio sin calificar para el alumno. La entrega consistirá en subir un único fichero (.c) con el código fuente de la solución al problema.

#### Problema: Aplicación informática para gestión de datos de microarrays

El grupo de investigación genética del Centro Andaluz de Biología del Desarrollo (CABD), necesita desarrollar una aplicación informática en lenguaje C que gestione de manera automática los resultados de sus microarrays y realice determinadas operaciones estadísticas con ellos.

En el campo de la biotecnología, el estudio, procesamiento y análisis de experimentos con microarrays es fundamental para obtener resultados de laboratorio satisfactorios. Un chip microarray es una colección de genes expresados de un determinado tejido bajo unas condiciones experimentales concretas.

Por ejemplo, si estamos estudiando el desarrollo de una enfermedad con ratones como huéspedes de la misma, podemos obtener dos microarrays bajo dos condiciones experimentales:

- Condición 1: con los genes expresados para un ratón sano (Fig. 1).
- Condición 2: con los genes expresados para un ratón enfermo (Fig. 2).

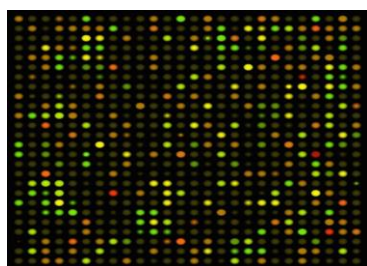


Fig. 1: Microarray ratón sano

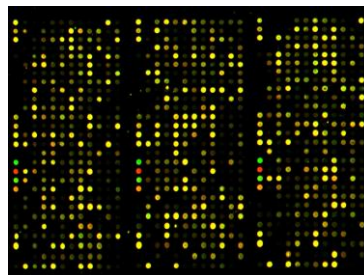


Fig. 2: Microarray ratón enfermo

Para que estos experimentos puedan ser procesados y analizados con técnicas computacionales, se realiza, mediante técnicas de escáner, una transformación de microarray físico a microarray digital. Esta transformación consiste en ordenar todos los experimentos escaneados en una tabla digital de tal forma que las filas de dicha tabla representan los genes y, las columnas, las condiciones experimentales. Siguiendo con el ejemplo anterior, y suponiendo que cada microarray contiene la expresión de 200 genes, nuestra tabla o microarray digital quedará de la siguiente forma:

	Condición 1 (sano)	Condición 2 (enfermo grado 1)	Condición 3 (enfermo grado 2)
Gen 1	Expresión del gen 1 bajo la condición experimental 1	Expresión del gen 1 bajo la condición experimental 2	Expresión del gen 1 bajo la condición experimental 3
Gen 2	Expresión del gen 2 bajo la condición experimental 1	Expresión del gen 2 bajo la condición experimental 2	Expresión del gen 2 bajo la condición experimental 3
...	...	...	...
Gen 200	Expresión del gen 200 bajo la condición experimental 1	Expresión del gen 200 bajo la condición experimental 2	Expresión del gen 200 bajo la condición experimental 3



Teniendo en cuenta la estructura de los microarrays digitales, realice las siguientes tareas de programación relacionadas.

### SE PIDE:

De manera general, el programa mostrará el siguiente menú de opciones de manera ininterrumpida hasta que el usuario introduzca un cero para finalizar el programa:

```
----- MENU -----
1.- Introducir los nombres de los genes
2.- Introducir los datos de la matriz del microarray
3.- Mostrar los datos del microarray
4.- Modificar el valor de un gen para una condición
5.- Media y condición que más se expresa de un gen
6.- Calcular la matriz de distancias
7.- Mostrar la matriz de distancias
8.- Ordenar las distancias de un gen y mostrar
0.- Salir.
-----
Introduzca una opción:
```

1. Si el usuario introduce la opción 1, el programa pedirá al usuario que introduzca el código o nombre de 3 genes. (haga uso de la función **rellenarGenes**)

2. Si el usuario introduce la opción 2, el programa pedirá al usuario que introduzca los resultados del microarray para 4 determinadas condiciones, a las que llamaremos "Cond0", "Cond1", "Cond2" y "Cond3". Dicha opción informará al usuario de a qué gen y a qué condición está solicitando el resultado, como se muestra a continuación (haga uso de la función **rellenarMicroarray**):

```
Introduzca una opcion
2
Introduzca el resultado de exponer el gen Gen0 a la condicion Cond0:
```

3. Si el usuario introduce la opción 3, el programa imprimirá la matriz de datos por pantalla, como se muestra en la figura. Si el usuario no ha introducido aún los datos mostrará el mensaje: "Por favor, introduzca el nombre de los genes primero", volviendo a mostrar el menú principal (haga uso de la función **mostrarMicroarray**).

```
Introduzca una opcion: 3

=====
Mostrando el microarray
=====
      Cond0  Cond1  Cond2  Cond3
Gen0  0.5000  0.6000  0.9000  0.2000
Gen1  1.0000  0.8000  0.9000  0.7000
Gen2  0.8000  0.7000  0.6000  1.0000
```

4. Si el usuario introduce la opción 4, el programa pedirá al usuario el nombre o código de un gen (de los introducidos en la opción 1), una determinada condición (las posibles son "Cond0", "Cond1", "Cond2" o "Cond3") y un valor que sustituya al introducido anteriormente en esa posición de la matriz de resultados. Una vez realizado el cambio, se volverá a mostrar la matriz de resultados para así comprobar que el cambio se efectuado de forma correcta. Si no encontrara el gen o la condición, se informará al usuario de ello (haga uso de las funciones **modificarMicroarray** y **mostrarMicroarray**).

```
Introduzca una opcion: 4

Introduzca gen a buscar:Gen1
Introduzca condicion a buscar:Cond2
Introduce el valor a modificar del gen Gen1 para la condicion Cond2:0.25

=====
Mostrando el microarray
=====
      Cond0  Cond1  Cond2  Cond3
Gen0  0.5000  0.6000  0.9000  0.2000
Gen1  1.0000  0.8000  0.2500  0.7000
Gen2  0.8000  0.7000  0.6000  1.0000
```

5. Si el usuario introduce la opción 5, el programa pedirá al usuario que introduzca el nombre de un determinado gen y mostrará por pantalla la media para todas las condiciones y, por otro lado, el nombre de la condición que más se expresa. En esta opción **no hace falta** que compruebe si el nombre del gen existe o no, se considerará que el usuario introducirá un nombre de gen válido (haga uso de la función **calculosMicroarray**).



```
Introduzca una opcion: 5

Introduzca gen a buscar:Gen2
La media del gen Gen2 es 0.775000
La condicion que mas se expresa es Cond3
```

6. Si el usuario introduce la opción 6, el programa calculará la matriz de distancias de cada gen con respecto a los demás genes. Válgase para ello de la fórmula explicada en el apartado "NOTAS" al final del documento. Esta función realizará el cálculo de la matriz de distancias, sin mostrarla por pantalla (haga uso de la función **calculoDistancias**).

7. Si el usuario introduce la opción 7, el programa imprimirá la matriz de distancias calculada en el punto 6, como se muestra en la figura (haga uso de la función **mostrarDistancias**).

```
Introduzca una opcion: 7

=====
Mostrando la matriz de distancias
=====
      Gen0    Gen1    Gen2
Gen0    0.0000    1.8500    1.5000
Gen1    1.8500    0.0000    0.9500
Gen2    1.5000    0.9500    0.0000
```

8. Si el usuario introduce la opción 8, el programa pedirá al usuario el nombre de un determinado gen e imprimirá sus distancias con otros genes ordenados de manera ascendente (de menor a mayor) como se muestra en la figura. Si no encontrara el gen, se informará al usuario de ello (haga uso de la función **ordenaDistanciaGen**).

```
Introduzca una opcion: 8

Introduzca gen para buscar sus distancias:Gen0
El vector de distancias ordenado del gen Gen0.
distancia 0.000000 con gen Gen0.
distancia 1.500000 con gen Gen2.
distancia 1.850000 con gen Gen1.
```



## FUNCIONES A DESARROLLAR:

1. **rellenarGenes**: función que recibe una matriz de caracteres (con un tamaño a priori fijado por usted) y almacenará en ella el nombre de los genes a estudiar.
2. **rellenarMicroarray**: función que recibe la matriz de caracteres con el nombre de los genes, una matriz de cadena de caracteres con los nombres de las condiciones a analizar y una matriz de tipo decimal vacía. Dicha función solicitará al usuario que rellene los datos del microarray, especificando a qué gen y a qué condición se refiere.
3. **mostrarMicroarray**: función que recibe la matriz de caracteres con el nombre de los genes, la matriz de caracteres con el nombre de las condiciones a analizar y la matriz de datos del microarray e imprimirá los datos por pantalla.
4. **modificarMicroarray**: función que recibe la matriz de datos del microarray, la matriz de caracteres que almacena el nombre de los genes, la matriz de caracteres de condiciones y dos cadenas de caracteres: una correspondiente al nombre de un gen y otra a la de una determinada condición (solicitados previamente en la función principal). Esta función buscará y sustituirá el valor numérico real por el que introduzca el usuario por consola. Si no existe el gen o no existe la condición solicitada, se informará al usuario de ello.
5. **calculosMicroarray**: función que recibe la matriz de datos del microarray, una cadena de caracteres con el nombre de un determinado gen, la matriz de caracteres que almacena el nombre de los genes y un puntero a una variable que contendrá la posición del microarray con el valor que más se expresa. Esta función devolverá, además, un valor de tipo decimal que contendrá la media de los valores expresados de ese determinado gen.
6. **calculoDistancias**: función que recibe la matriz de datos del microarray y una matriz de tipo decimal vacía que guardará las distancias euclídeas entre los datos de los distintos genes. Válgase para ello de la fórmula explicada en el apartado "NOTAS" al final del documento. Nótese que la diagonal principal de la matriz de distancias siempre valdrá 0.0 al compararse la distancia de un gen consigo mismo.
7. **mostrarDistancias**: función que recibe la matriz de distancias y la matriz de caracteres que almacena el nombre de los genes. Esta función imprimirá por pantalla los datos de la matriz de distancias.
8. **ordenaDistanciaGen**: función que recibe una cadena con el nombre de un determinado gen, la matriz de caracteres que contiene el nombre de los distintos genes y la matriz de distancias. Primero, la función realizará una copia de la fila de las distancias correspondientes al gen de entrada en un vector auxiliar y una copia de los nombres de los genes. Seguidamente, se ordenarán los valores de las distancias de dicho vector de menor a mayor y, consecuentemente, la copia del vector de nombres. Finalmente, se imprimirán las distancias ordenadas junto con el nombre del gen al que corresponde dicha distancia.  
Si no existiera el gen, se informará al usuario de ello.

## NOTAS:

- Declare las constantes COND (4) y GEN (3), correspondientes a las cuatro condiciones y tres tipos de genes del problema.
- Declare en la función principal una matriz con los códigos de las condiciones. Estas serán "Cond0", "Cond1", "Cond2" y "Cond3".
- Concepto de Distancia Euclídea entre dos vectores:

Dados:  $\hat{A} = \langle a_1, a_2, a_3, \dots, a_n \rangle$  y  $\hat{B} = \langle b_1, b_2, b_3, \dots, b_n \rangle$ :

$$DISTANCIA\_EUCLÍDEA(\hat{A}, \hat{B}) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$