



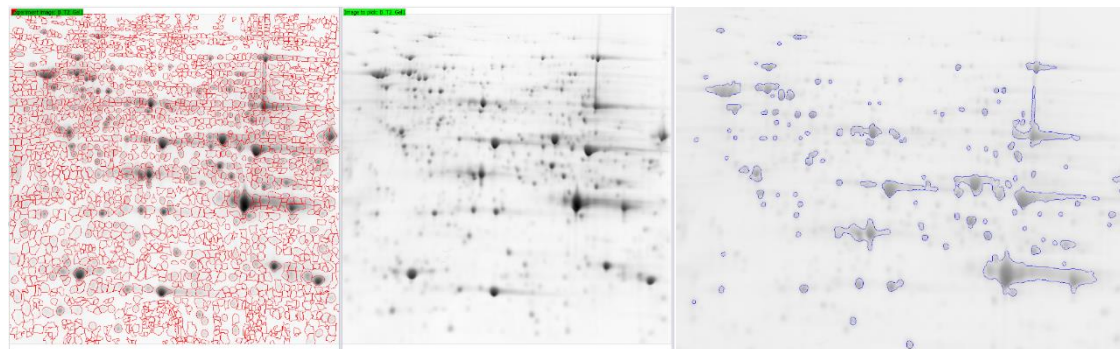
Máster Universitario en
**Ingeniería Biomédica
y Salud Digital**

Técnicas Avanzadas de Análisis y Reconocimiento de Imágenes Biomédicas

PEDRO REAL

Estudio de la viabilidad del algoritmo WaterShed para análisis de imágenes médicas

PLÁCIDO FERNÁNDEZ CUEVAS



ÍNDICE

1. INTRODUCCIÓN
2. IMPLEMENTACIÓN
3. DISEÑO DEL PROGRAMA
4. SOFTWARE MELANIE
5. EXPERIMENTACIÓN
6. CONCLUSIONES
7. BIBLIOGRAFÍA

INTRODUCCIÓN

PRIMERA TOMA DE CONTACTO

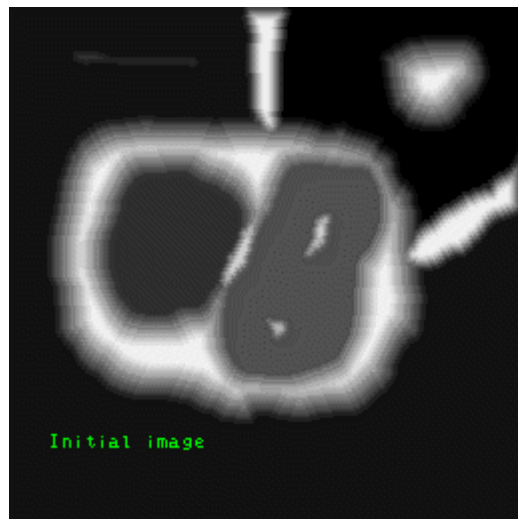
INTRODUCCIÓN – SEGMENTACIÓN DE IMÁGENES

La segmentación de imágenes consiste en dividir la imagen en regiones conexas donde cada región posee propiedades que las diferencian de las demás



INTRODUCCIÓN – WATERSHED

- Algoritmo de segmentación basado en crecimiento de semillas
- Introducida por C. Latuejeoul como “Transformación de WaterShed” y más adelante fue mejorada por S. Beucher y denominada “WaterShed” [1].
- La técnica considera una imagen en escala de grises como una imagen topográfica, donde a cada pixel se le asocia una altura [1].



INTRODUCCIÓN – OBJETIVOS

El objetivo de este trabajo es estudiar la viabilidad del algoritmo WaterShed para tratamiento de imágenes médicas comparando la segmentación producida por un programa propio en comparación con la segmentación de un programa profesional llamado Melanie.

IMPLEMENTACIÓN

ALGORITMO DE WATERSHED

IMPLEMENTACIÓN

Para obtener los resultados del WaterShed, es necesario realizar una implementación del mismo, la cual debe seguir los siguientes pasos:

1. Ordenación de píxeles según el nivel de gris mediante pila FIFO
2. Etiqueta a cada mínimo local, la cual se propaga a los píxeles adyacentes del nivel h
3. Se analiza la imagen desde el mínimo valor de gris hasta el más alto

Para cada iteración de análisis:

1. Componentes conexas entre la binarización h y $h+1$
2. Inundación:
 1. Crecimiento de una vasija ya existente
 2. Aparición de una nueva vasija
 3. Determinación de zonas de influencia

IMPLEMENTACIÓN – PSEUDOCÓDIGO

```
# define MASK -2 /* valor inicial del nivel del Umbral */

# define WSHED 0 /* valor de los pixeles que pertenecen a los watersheds */

# define INIT -1 /* valor inicial de  $[[im]]_0$  */

-input:  $[[im]]_i$ , decimal image;

-output:  $[[im]]_0$ , image of the labeled watersheds;

* Inicializaciones:

-Value INIT que es asignado a cada pixel de  $[[im]]_0$ :

 $\forall p \in D_{im0}, [[im]]_0(p) = INIT;$ 

    -etiqueta_actual: 0;

    -actual_dist: integer variable

    - $[[im]]_0$ : imagen temporal (trabajo) de distancias inicializado a 0;

* Ordenar los píxeles de  $[[im]]_i$  de acuerdo al incremento de sus valores de gris

h_min y h_max son designados como los valores de niveles de gris mas bajo y mas alto
respectivamente

For h in range(h_min hasta h_max){
```

```
/*geodesic SKIZ de nivel h-1 dentro del nivel h*/

Foreach pixel p tal que  $[[im]]_i(p) = h\{$ 

     $[[im]]_0(p) = MASK$ 

    If existe  $p' \in NG(p)$  tal que  $[[im]]_0(p') > 0$  or

         $[[im]]_0(p') == WSHED \{$ 

             $[[im]]_d(p) = 1;$ 

            Agregar(p);

        }

    }

Actual_dist = 1;

Agregar(ficticio_pixel);

While(1) {

    P = Primero();

    If p == ficticio_pixel {

        If Vacio() == true {break}

        Else {Agregar(ficticio_pixel)}

        Actual_dist++

        P = Primero()

    }

}
```

IMPLEMENTACIÓN – PSEUDOCÓDIGO

```

Foreach pixel  $p' \in NG(p)$  {
  If ( $\llbracket im \rrbracket\_d(p') < actual\_dist$  && ( $\llbracket im \rrbracket\_0(p') > 0$ ) || ( $\llbracket im \rrbracket\_0(p') == WSHED$ )){
    /* i.e,  $p'$  ya pertenece a una vasija etiquetado o a el watershed*/
    If ( $\llbracket im \rrbracket\_0(p') > 0$ ){
      If ( $\llbracket im \rrbracket\_0(p) == MASK$  ||  $\llbracket im \rrbracket\_0(p) == WSHED$ ) {
         $\llbracket im \rrbracket\_0(p) = \llbracket im \rrbracket\_0(p')$ 
      } else {
         $\llbracket im \rrbracket\_0(p) = WSHED$ 
      }
    }else if ( $\llbracket im \rrbracket\_0(p') == MASK$  &&  $\llbracket im \rrbracket\_d(p') == 0$  {
       $\llbracket im \rrbracket\_d(p') = Actual\_dist + 1$ 
      Agregar( $p'$ )
    }
  }
}

```

```

/* Chequear si nuevo minimo ha sido descubierto*/
Foreach pixel p tal que  $\llbracket im \rrbracket\_i(p) = h$  {
   $\llbracket im \rrbracket\_d(p) = 0$ 
  If ( $\llbracket im \rrbracket\_0(p) == MASK$  {
    Etiqueta_actual++
    Agregar(p)
     $\llbracket im \rrbracket\_0(p) = etiqueta\_actual$ 

    While Vacio() == false {
       $P' = Primero()$ 
      Foreach pixel  $p'' \in NG(p')$  {
        If( $\llbracket im \rrbracket\_0(p'') == MASK$ ) {
          Agregar( $p''$ )
           $\llbracket im \rrbracket\_0(p'') = etiqueta\_actual$ 
        }
      }
    }
  }
}

```

DISEÑO DEL PROGRAMA

SOFTWARE PROPIO

DISEÑO DEL PROGRAMA

Durante la realización de la experimentación, se ha hecho uso de una librería de imágenes muy conocida, con buenas prestaciones y de código abierto conocida como OpenCV

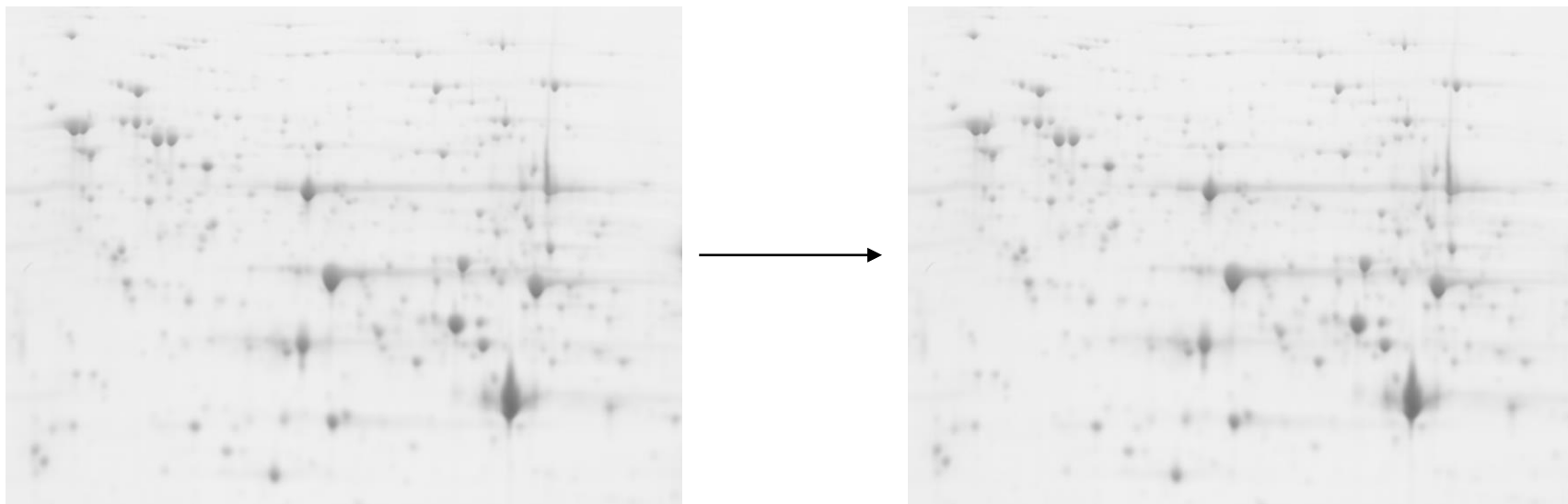


Esta librería nos proporciona una implementación del algoritmo WaterShed, la cual podemos conocer con certeza que funciona correctamente.

Pero el análisis de imágenes no consiste únicamente en aplicar el algoritmo WaterShed, también debemos realizar un pre-procesamiento previo de la imagen.

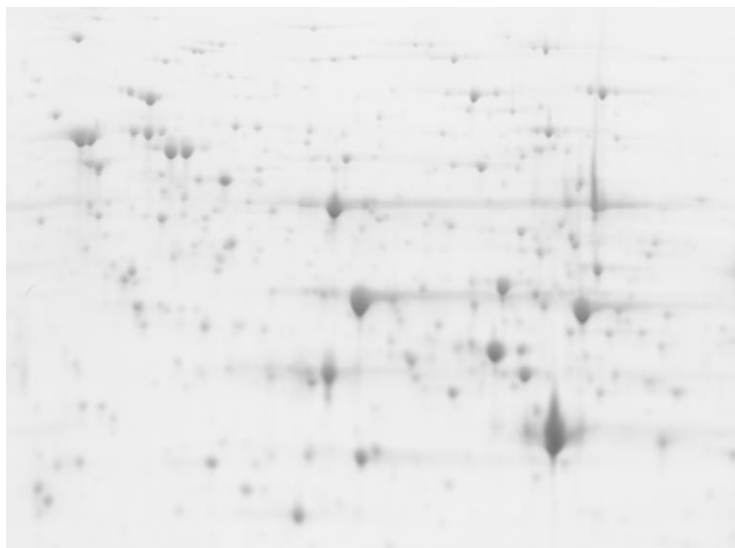
DISEÑO DEL PROGRAMA – PRE-PROCESAMIENTO

1. Conversión de la imagen a escala de grises:



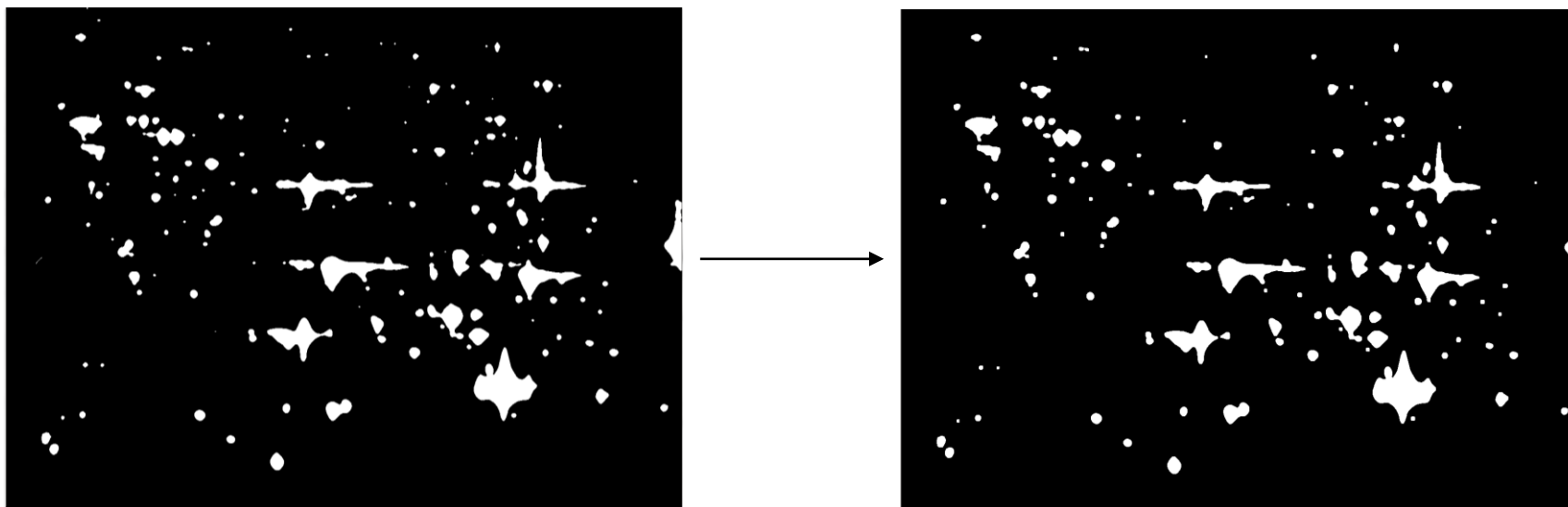
DISEÑO DEL PROGRAMA – PRE-PROCESAMIENTO

2. Aplicación de un filtro umbral (threshold) para diferenciar el fondo de la imagen:



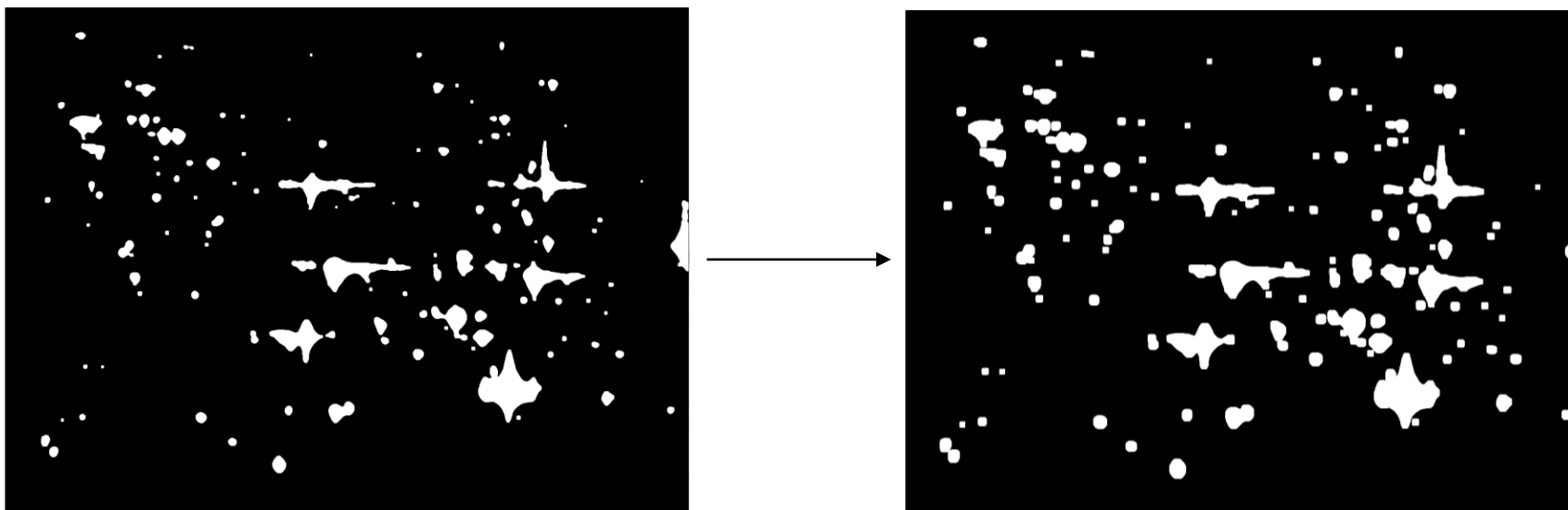
DISEÑO DEL PROGRAMA – PRE-PROCESAMIENTO

3. Aplicación de apertura para eliminar el ruido de fondo:



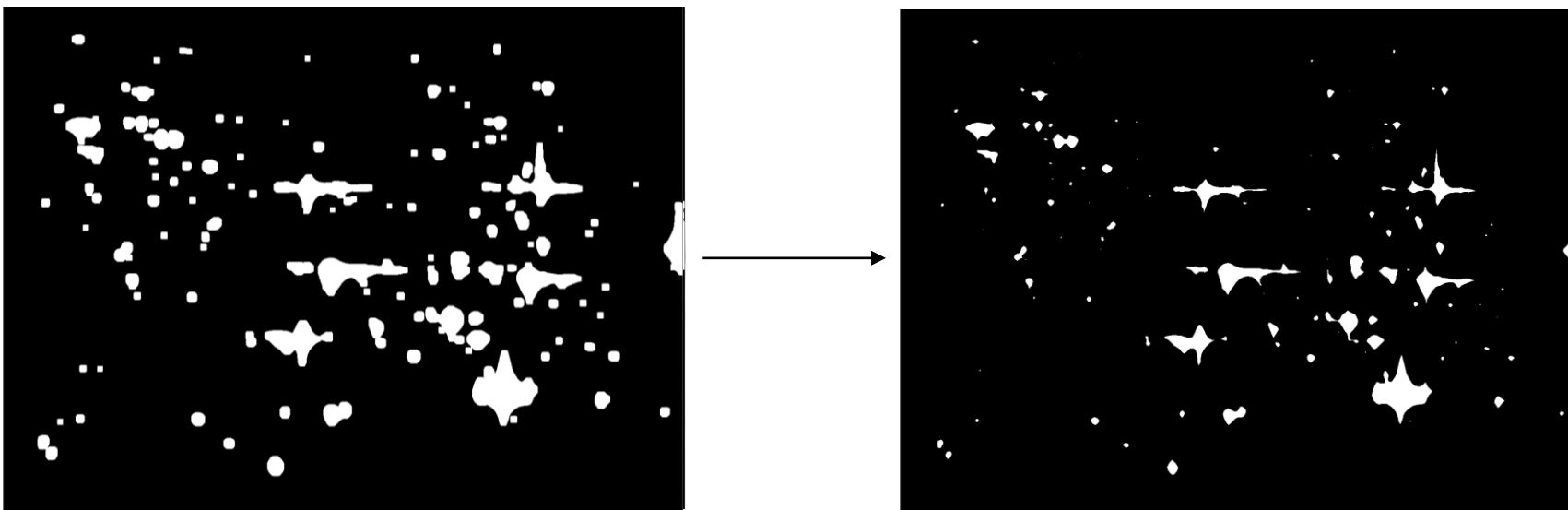
DISEÑO DEL PROGRAMA – PRE-PROCESAMIENTO

4. Extraemos el fondo seguro de la imagen:



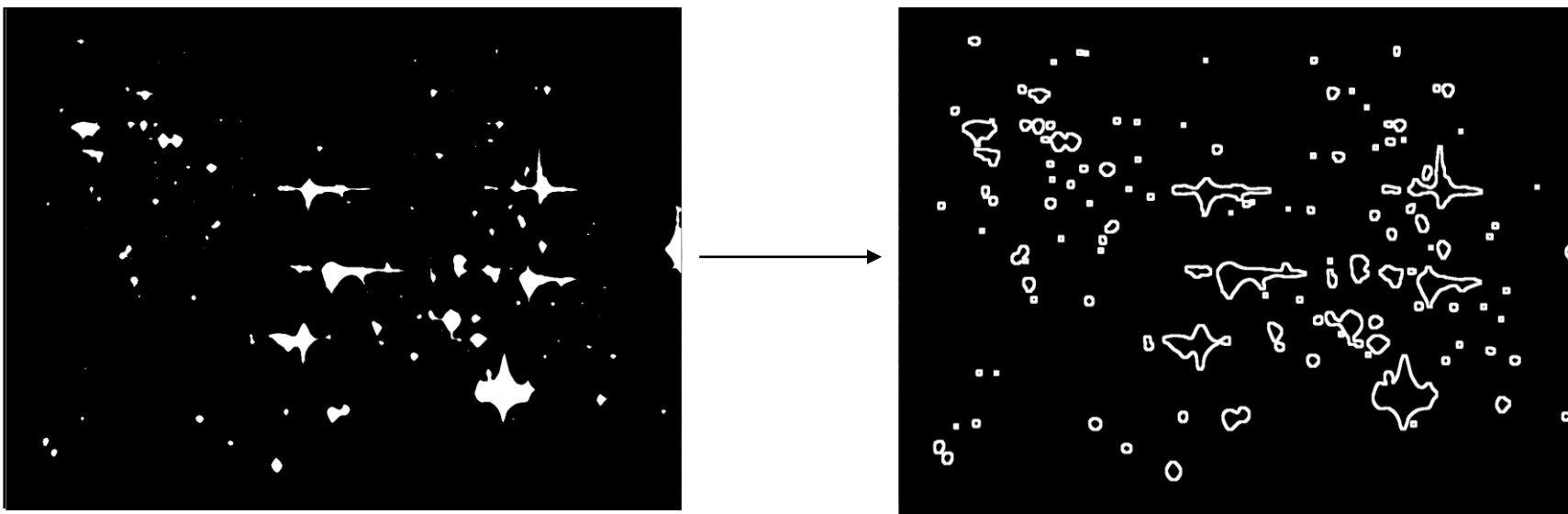
DISEÑO DEL PROGRAMA – PRE-PROCESAMIENTO

5. Extraemos la zona que seguro que no corresponde al fondo:



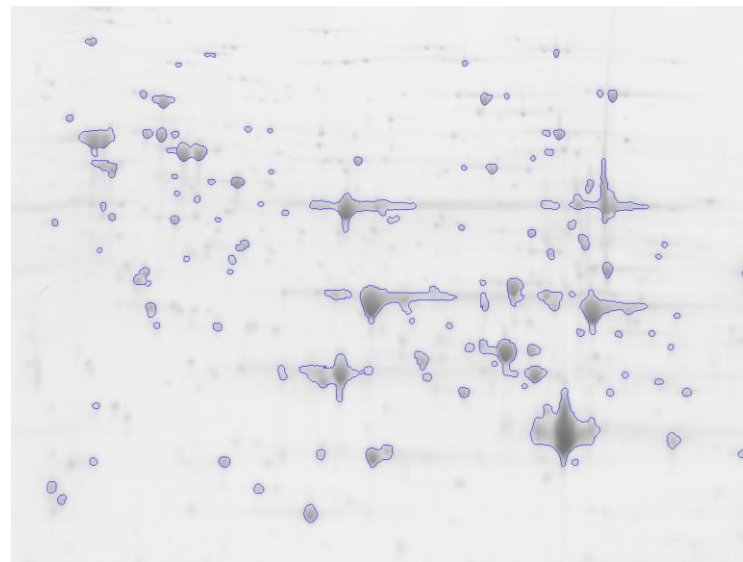
DISEÑO DEL PROGRAMA – PRE-PROCESAMIENTO

6. Extraemos las regiones desconocidas:



DISEÑO DEL PROGRAMA – PRE-PROCESAMIENTO

7. Extraemos los marcadores y aplicamos WaterShed:



SOFTWARE MELANIE

SOFTWARE PARA ANÁLISIS DE IMÁGENES

SOFTWARE MELANIE

Para la comparación de los resultados se ha utilizado un programa de segmentación profesional llamado Melanie.



Dicho programa cuenta con un gran conjunto de funciones enfocadas al análisis de imágenes, tales como:

- Control de calidad de las imágenes
- Diseño experimental
- Alineación
- Detección
- Revisión
- Resultados
- Recolección

EXPERIMENTACIÓN

PRUEBAS REALIZADAS

EXPERIMENTACIÓN

La fase de experimentación consiste en la utilización del software profesional Melanie y nuestro software propio para realizar una segmentación de la misma imagen.

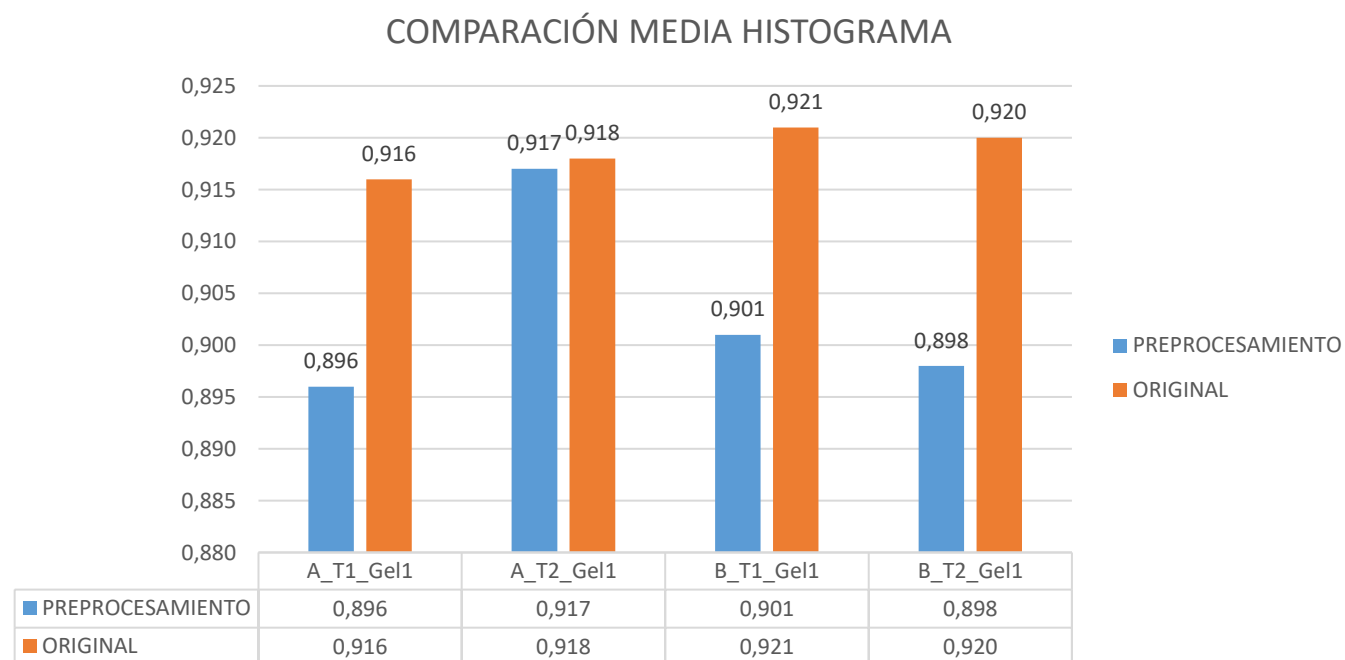
Nos centraremos únicamente en la “calidad” de la segmentación debido a que el trabajo se centra en estudiar la viabilidad del algoritmo WaterShed, de forma que sea viable utilizarlo para segmentar imágenes de forma similar a un nivel profesional.

Por otra parte, el software Melanie es capaz de extraer datos para realizar informes desde la segmentación, operación que nuestro algoritmo no es capaz de realizar.

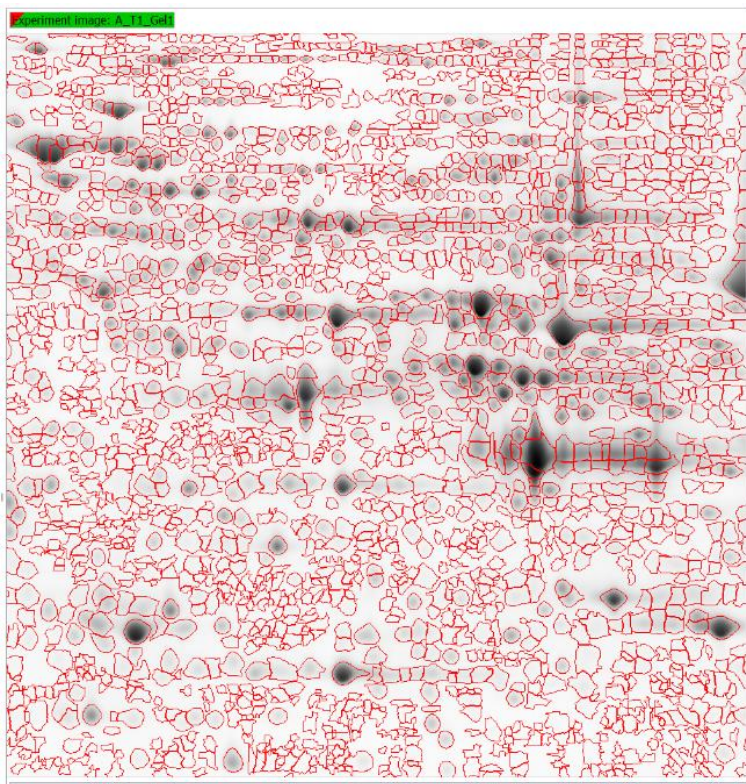
Las imágenes utilizadas están extraídas de la carpeta de ejemplos que nos proporciona el software Melanie, las cuales vienen preparadas para su uso.

EXPERIMENTACIÓN

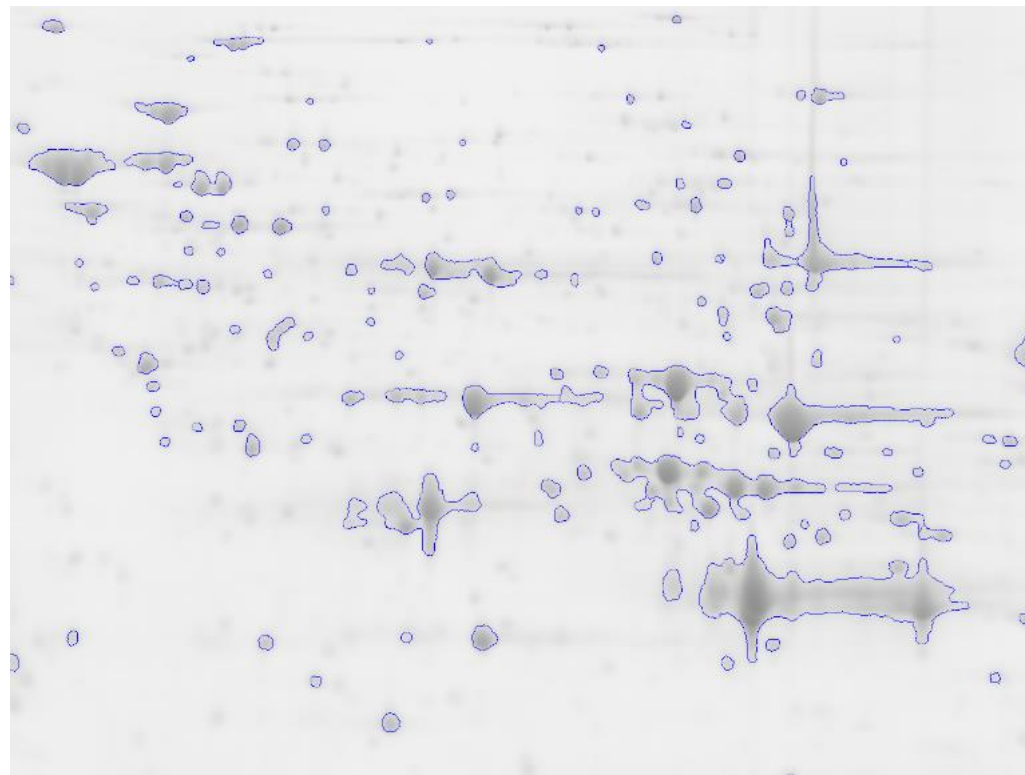
El software Melanie hace un preprocesamiento de las imágenes antes de comenzar con la segmentación. Si comparamos los histogramas de las imágenes originales y las imágenes preprocesadas por Melanie encontramos lo siguiente:



EXPERIMENTACIÓN – A_T1_Gel1

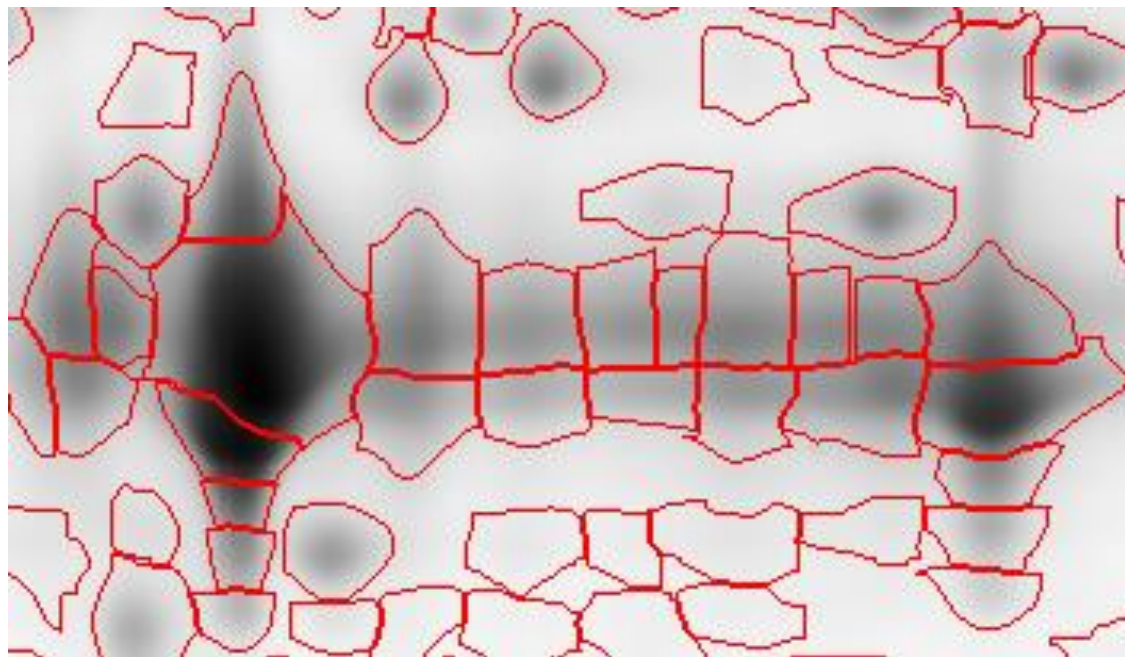


Melanie: 2012 Regiones

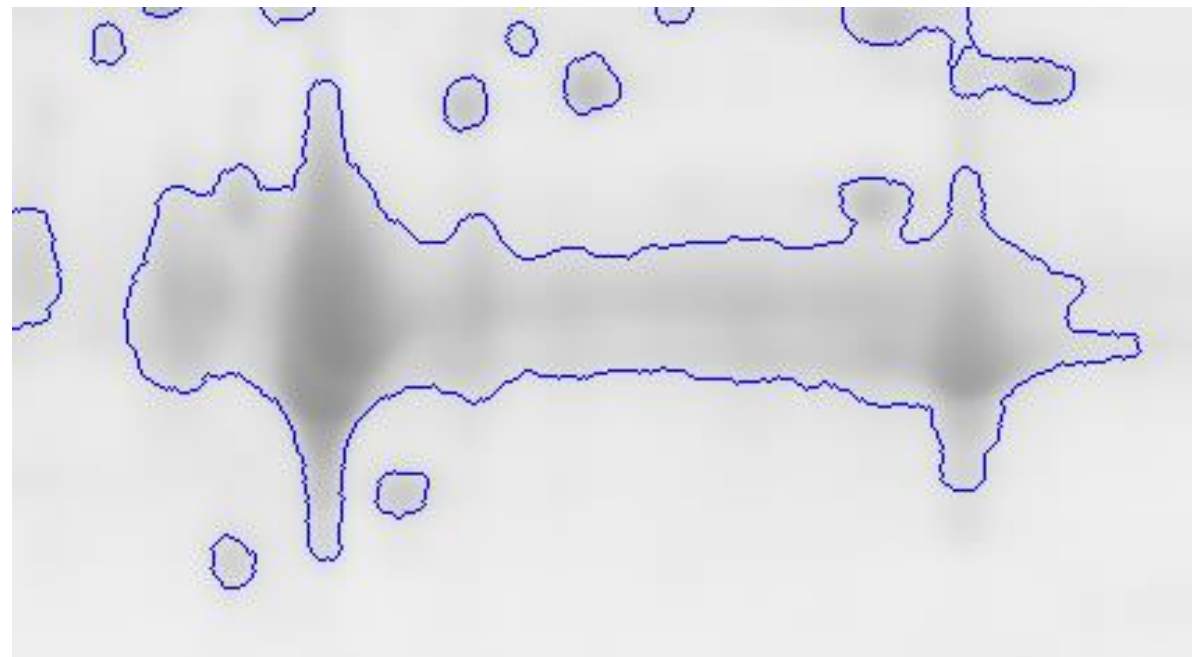


Software propio: 144 Regiones

EXPERIMENTACIÓN – A_T1_Gel1

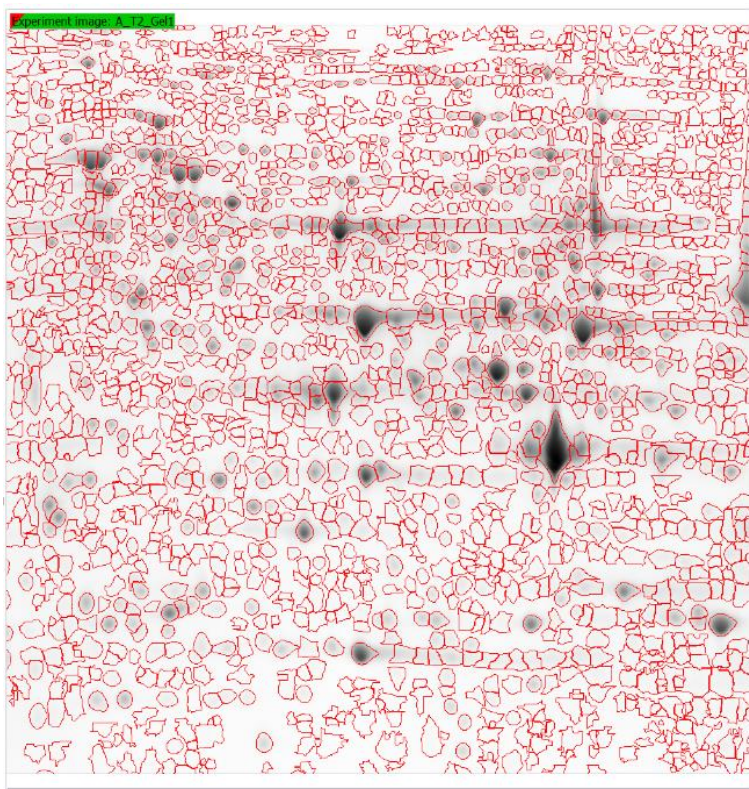


Melanie

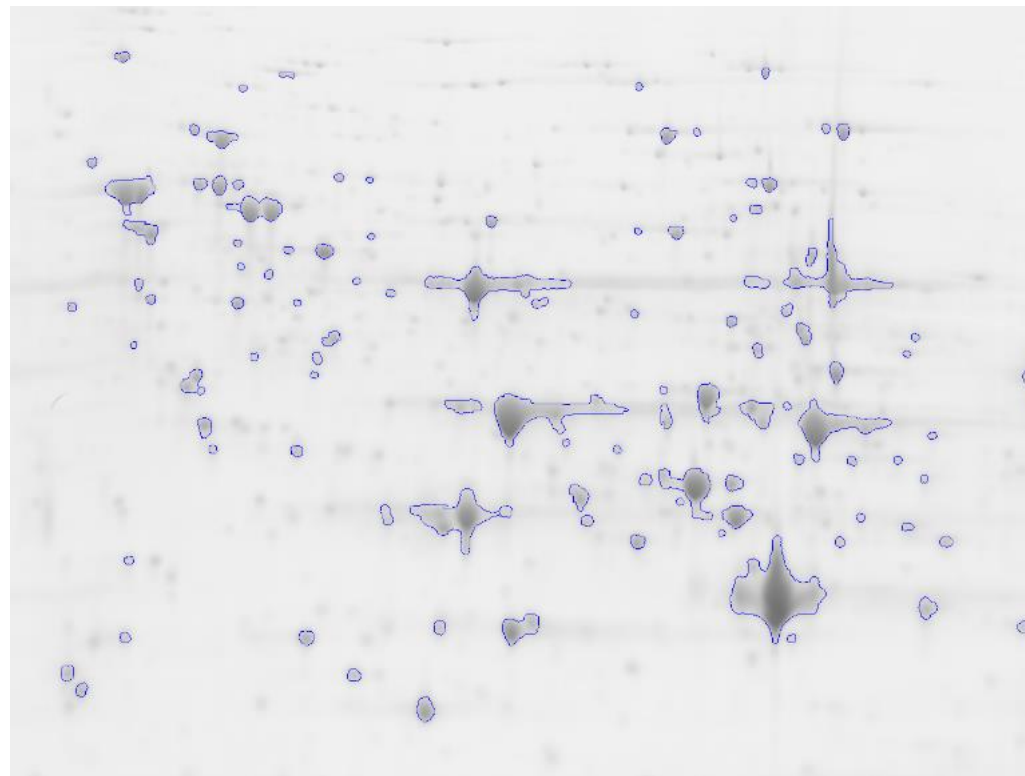


Software propio

EXPERIMENTACIÓN – A_T2_Gel1

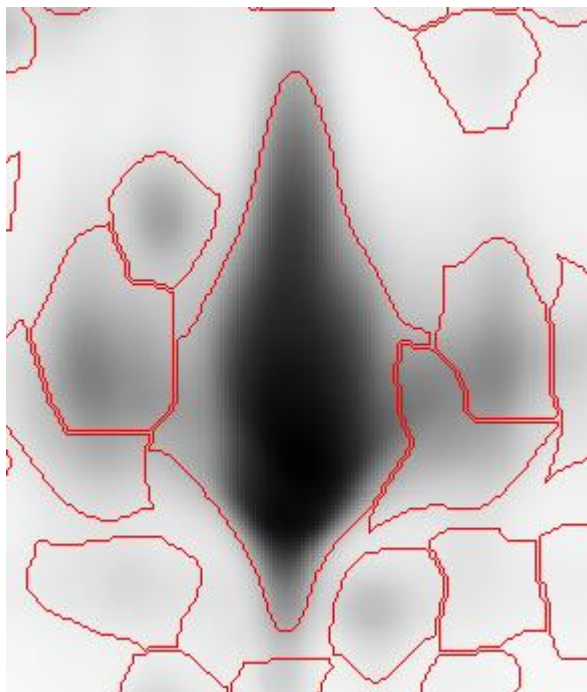


Melanie: 1974 Regiones

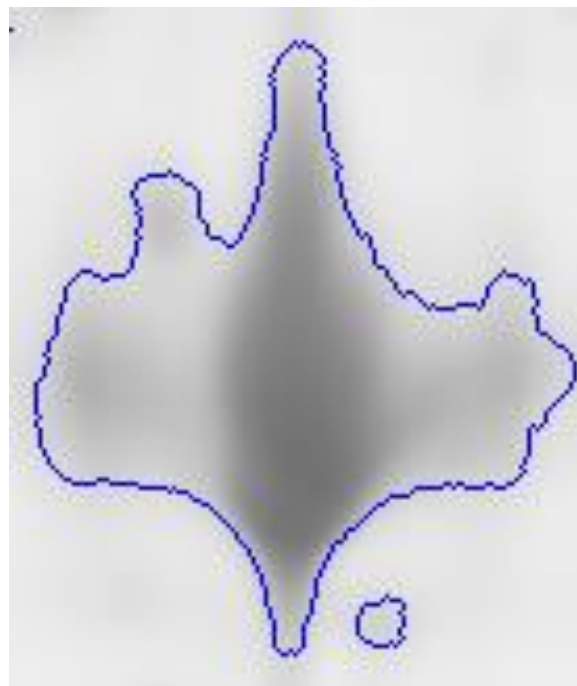


Software propio: 121 Regiones

EXPERIMENTACIÓN – A_T2_Gel1

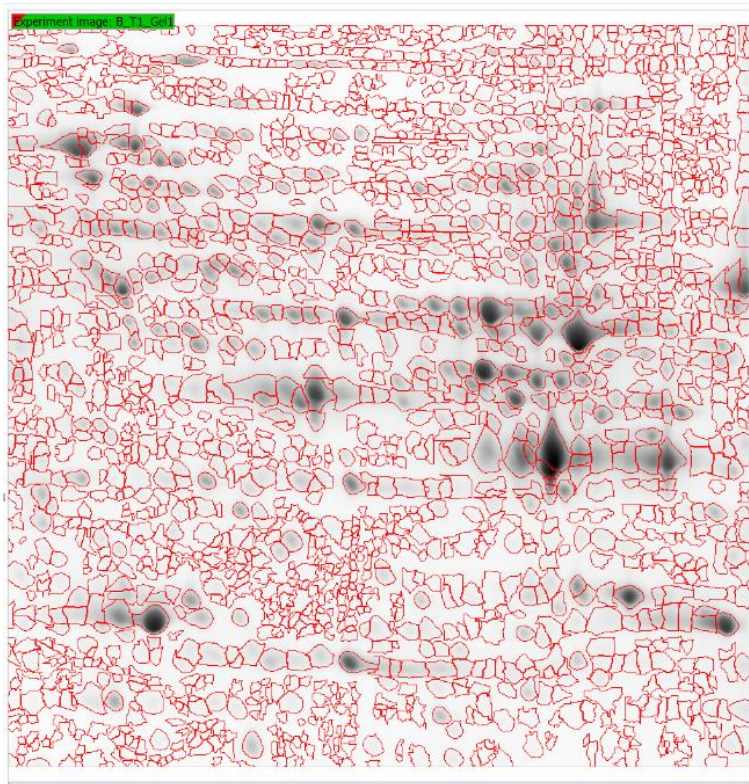


Melanie

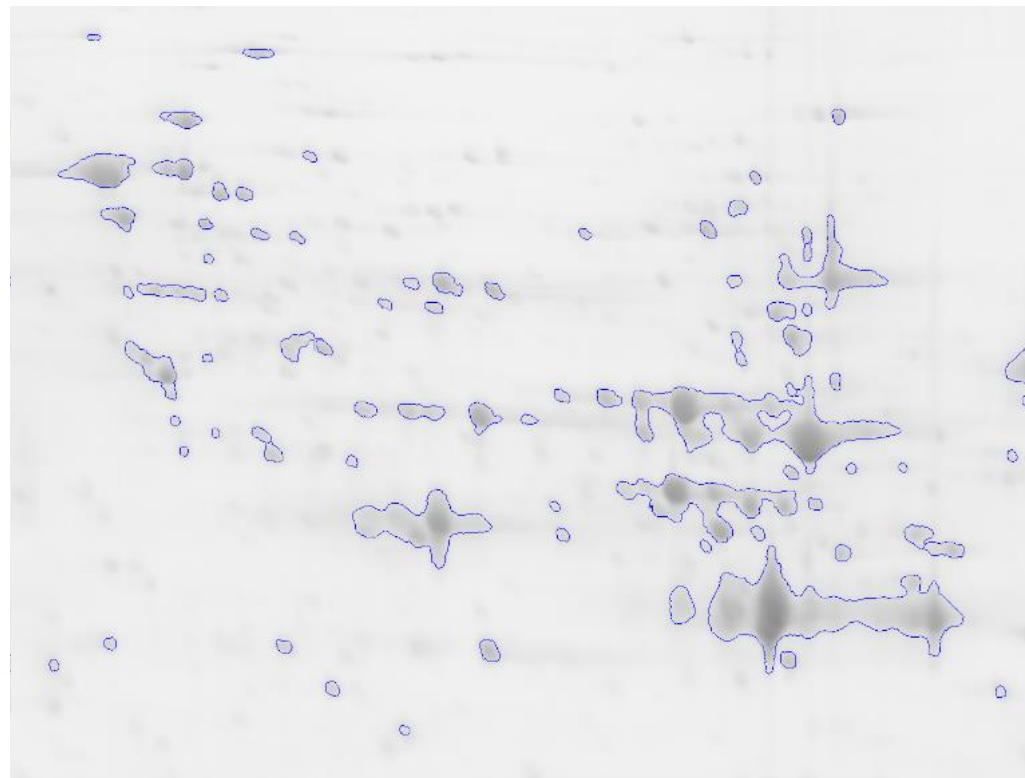


Software propio

EXPERIMENTACIÓN – B_T1_Gel1

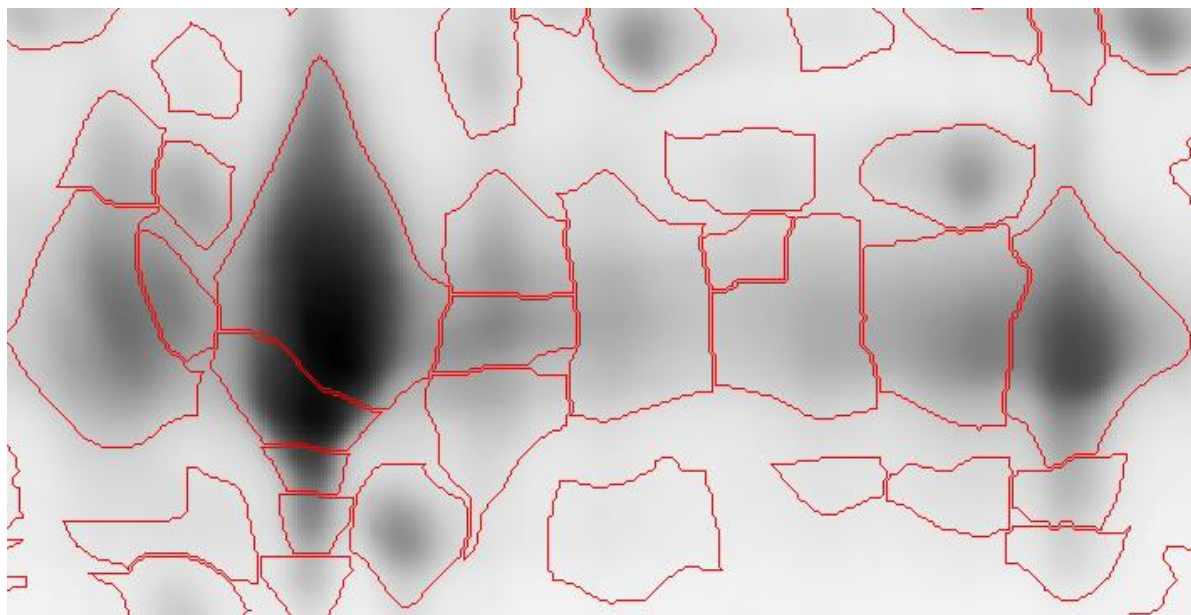


Melanie: 1836 Regiones

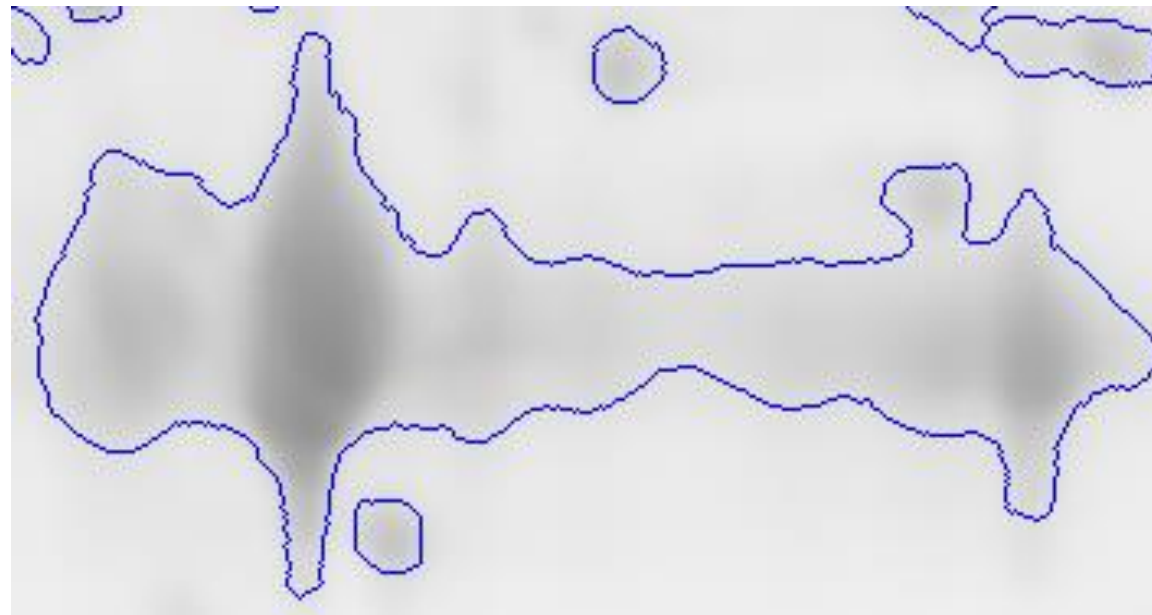


Software propio: 95 Regiones

EXPERIMENTACIÓN – B_T1_Gel1

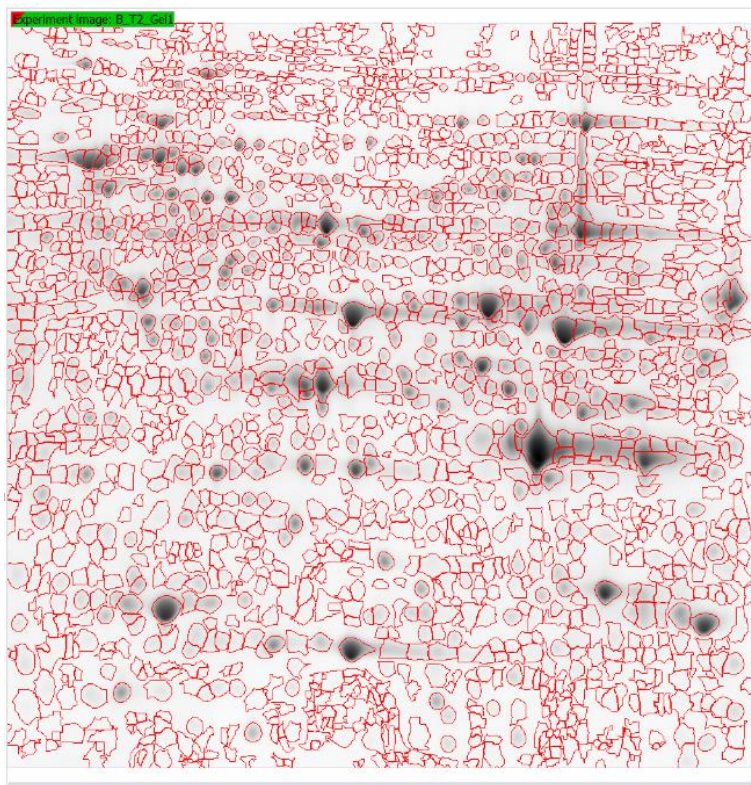


Melanie

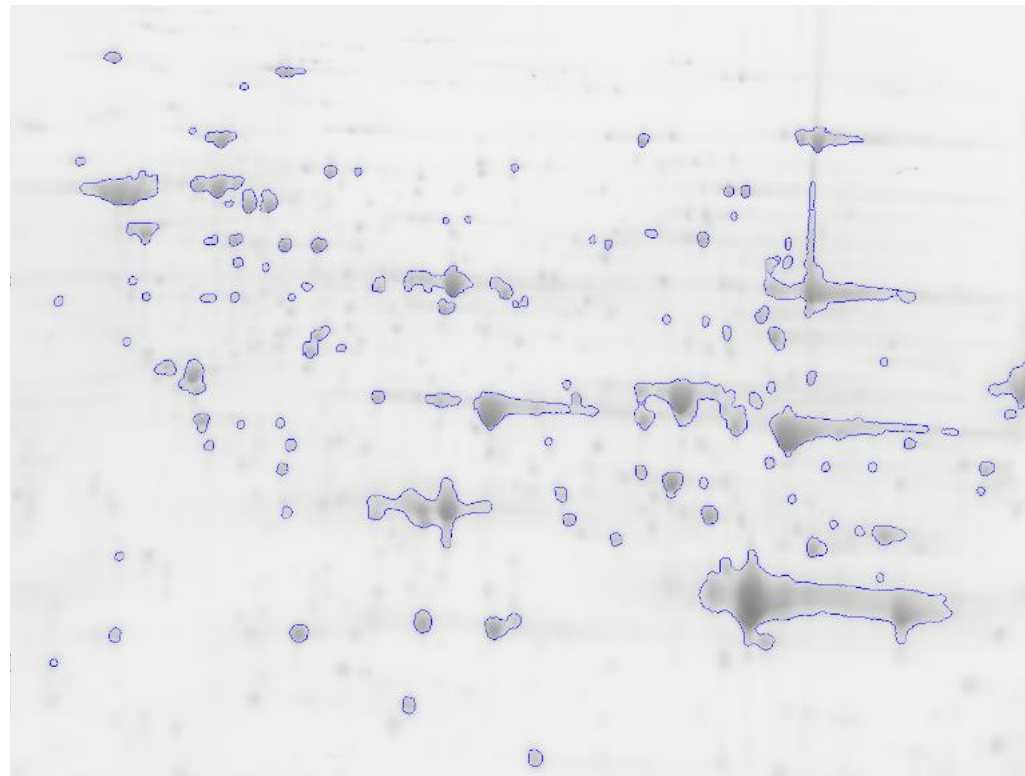


Software propio

EXPERIMENTACIÓN – B_T2_Gel1

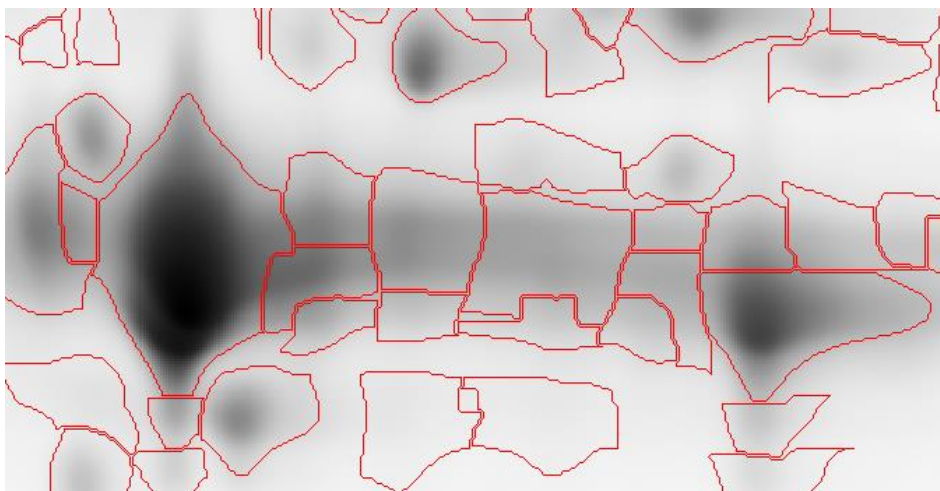


Melanie: 1858 Regiones

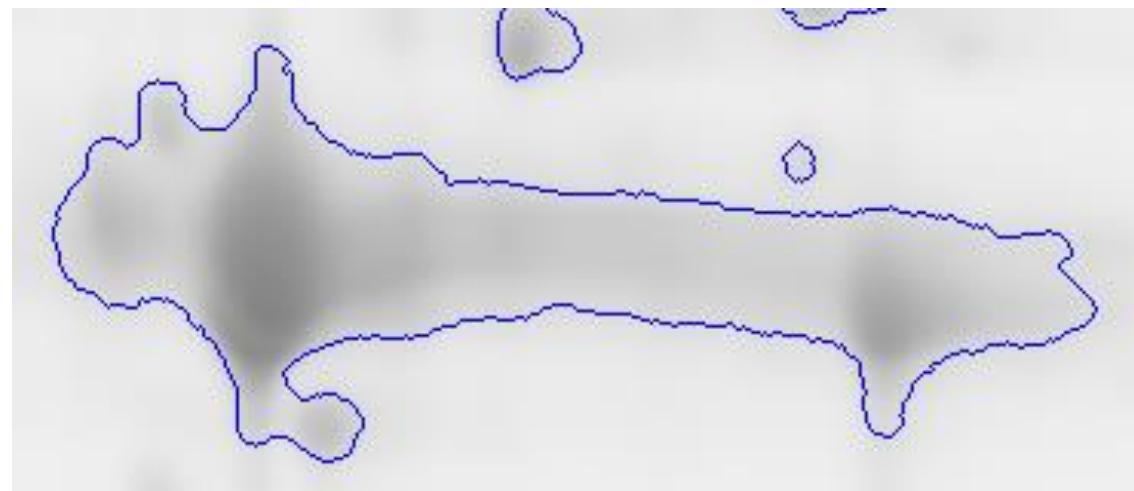


Software propio: 128 Regiones

EXPERIMENTACIÓN – B_T2_Gel1



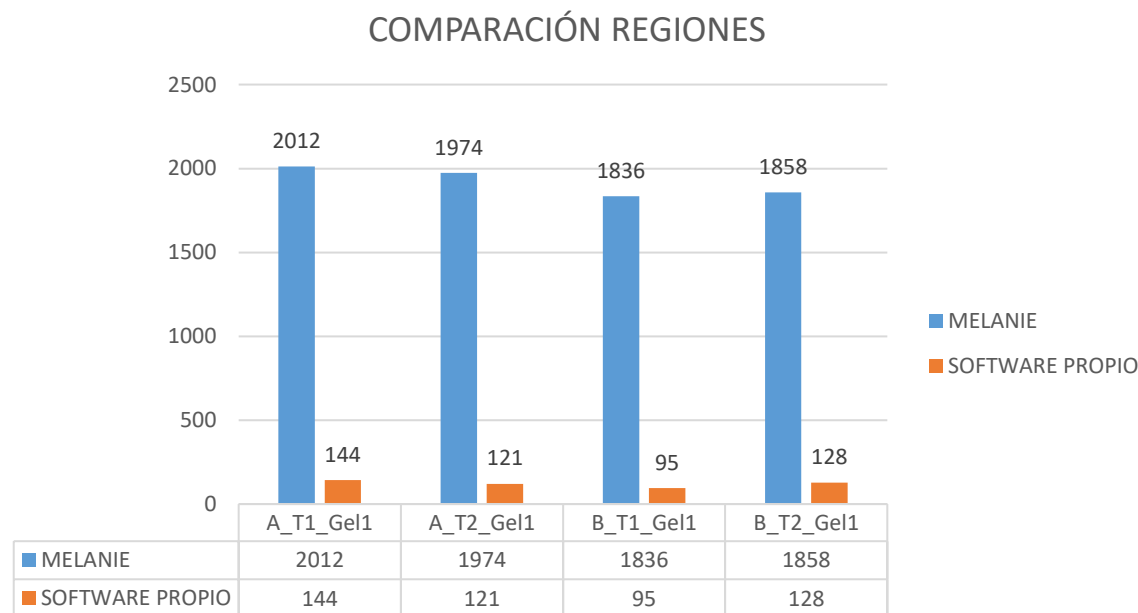
Melanie



Software propio

EXPERIMENTACIÓN

Si comparamos la segmentación realizada por cada programa, obtenemos los siguientes datos:



Se puede ver claramente, que el software Melanie produce un mayor número de segmentaciones que nuestro software propio

CONCLUSIONES

UN BREVE REPASO

CONCLUSIONES

WaterShed es un algoritmo muy versátil y eficiente, el cual posee una muy buena implementación dentro de la librería de OpenCV.

Los resultados obtenidos no son buenos o malos, simplemente diferentes debido al tipo de público enfocado.

Por otra parte, pueden ampliarse las posibilidades del software propio incluyendo a futuro la posibilidad de realizar informes en base a la segmentación obtenida.

BIBLIOGRAFÍA

REFERENCIAS CONSULTADAS

BIBLIOGRAFÍA

Chaki, Nabendu, Shaikh, Soharab Hossain, & Saeed, Khalid. (2014). *Exploring Image Binarization Techniques*. New Delhi: Springer.

Obtenido de https://fama.us.es/permalink/34CBUA_US/3enc2g/alma991013036744504987

Doxygen. (s.f.). *OpenCV - Image Segmentation with Watershed Algorithm*. Obtenido de

https://docs.opencv.org/4.x/d3/db4/tutorial_py_watershed.html

Edgar Molina, Julia Diaz, Hugo Hidalgo-Silva, & Edgar Chávez. (2018). Algoritmos de Binarización Robusta de Imágenes con Iluminación no Uniforme. *Revista Iberoamericana de Automática e Informática Industrial*, 10.

Fernando. (27 de mayo de 2014). *cortesfernando.blogspot*. Obtenido de <http://cortesfernando.blogspot.com/2014/05/binarizacion-imagen.html>

Implementación del algoritmo WaterShed para el análisis de imágenes médicas. (2011). *Revista de Investigación de Sistemas e Informática*, 8.

Obtenido de <https://core.ac.uk/download/pdf/304898407.pdf>

K. Somasundaram, & P. Kalavathi. (2011). Medical Image Binarization Using Square Wave Representation. *Springer, Berlin, Heidelberg*, 7.

Obtenido de https://link.springer.com/chapter/10.1007/978-3-642-19263-0_19

K. Somasundaram, & T. Genish. (2013). Binarization of MRI with Intensity Inhomogeneity using K-Means Clustering for Segmenting Hippocampus.

The International Journal of Multimedia & Its Applications, 9.

Obtenido de [https://www.researchgate.net/profile/Somasundaram-Karuppanagounder/publication/](https://www.researchgate.net/profile/Somasundaram-Karuppanagounder/publication/269678070_Binarization_of_MRI_with_Intensity_Inhomogeneity_Using_K-Means_Clustering_for_Segmenting_Hippocampus/links/56b8770408ae5ad3605f3363/Binarization-of-MRI-with-Intensity)

[269678070_Binarization_of_MRI_with_Intensity_Inhomogeneity_Using_K-Means_Clustering_for_Segmenting_Hippocampus/links/](https://www.researchgate.net/profile/Somasundaram-Karuppanagounder/publication/269678070_Binarization_of_MRI_with_Intensity_Inhomogeneity_Using_K-Means_Clustering_for_Segmenting_Hippocampus/links/56b8770408ae5ad3605f3363/Binarization-of-MRI-with-Intensity)

[56b8770408ae5ad3605f3363/Binarization-of-MRI-with-Intensity](https://www.researchgate.net/profile/Somasundaram-Karuppanagounder/publication/269678070_Binarization_of_MRI_with_Intensity_Inhomogeneity_Using_K-Means_Clustering_for_Segmenting_Hippocampus/links/56b8770408ae5ad3605f3363/Binarization-of-MRI-with-Intensity)

Magro, R. (2013). Binarización de imágenes digitales y su algoritmia como herramienta aplicada a la ilustración entomológica. *Boletín de la Sociedad Entomológica Aragonesa*, 22.

Mengual, P. C. (Mayo de 2014). Segmentación Automática del Cerebro Mediante Técnicas de Tratamiento Digital de Imagen. Valencia, Valencia, España.

Obtenido de https://riunet.upv.es/bitstream/handle/10251/37996/cabanillesmengual_paula.pdf?sequence=1

Oscar Daniel Chuk, Carlos Gustavo, Rodriguez Medina, & Adriana Luna. (2018). Aplicación del algoritmo de Watershed para análisis de calidad de arenas para fracking. *ResearchGate*, 8.

Obtenido de https://www.researchgate.net/publication/338556101_Aplicacion_del_algoritmo_de_Watershed_para_analisis_de_calidad_de_arenas_para_fracking

programación, T. d. (22 de Abril de 2017). *OpenCV operaciones morfológicas*. Obtenido de <http://acodigo.blogspot.com/2017/04/opencv-operaciones-morfologicas.html>

Rubiales, A. (24 de Agosto de 2021). *StackOverflow - Función WaterShed en Python*. Obtenido de <https://es.stackoverflow.com/questions/478895/funcion-watershed-en-python>

Swiss Institute of Bioinformatics. (s.f.). *Software Melanie*. Obtenido de <https://2d-gel-analysis.com/>

Wikipedia. (s.f.). *Wikipedia*. Obtenido de [https://es.wikipedia.org/wiki/Segmentaci%C3%B3n_\(procesamiento_de_im%C3%A1genes\)](https://es.wikipedia.org/wiki/Segmentaci%C3%B3n_(procesamiento_de_im%C3%A1genes))

Páginas web

[1] [OpenCV: OpenCV documentation](#) Página web donde se encuentra la documentación de OpenCV. Vista por última vez el 21/02/2022

[2] [Melanie 2D gel analysis software for protein expression profiling \(2d-gel-analysis.com\)](#) Página web del software Melanie. Vista por última vez el 22/02/2022

[3] [Anaconda | The World's Most Popular Data Science Platform](#) Página web del software utilizado para la creación del entorno de Python con OpenCV. Vista por última vez el 18/02/2022

¡GRACIAS POR SU ATENCIÓN!