



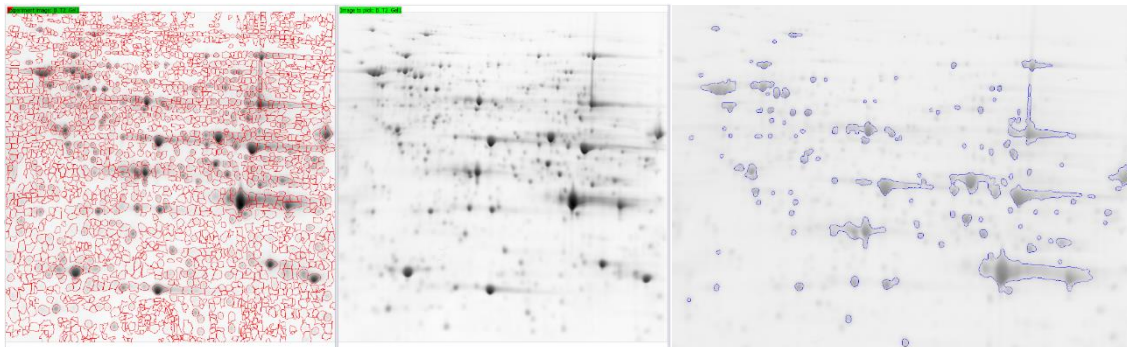
UNIVERSIDAD DE SEVILLA

Escuela Técnica Superior de Ingeniería Informática

Trabajo Dirigido de "Técnicas Avanzadas en Análisis y Reconocimiento de Imágenes Biomédicas"

Curso 2021/22

ESTUDIO DE VIABILIDAD DEL ALGORITMO WATERSHED PARA RECONOCIMIENTO DE REGIONES EN IMÁGENES MÉDICAS



Autor: Plácido Fernández Cuevas

Email de contacto: placidofernandez555@hotmail.com

ÍNDICE

1. INTRODUCCIÓN	5
2. MARCO TEÓRICO DEL PROBLEMA	7
2.1 SEGMENTACIÓN DE IMÁGENES.....	7
2.2 ALGORITMO WATERSHED.....	8
3. IMPLEMENTACIÓN WATERSHED.....	11
3.1 IDEA GENERAL.....	11
3.2 VARIABLES Y FUNCIONES.....	13
3.3 PSEUDOCÓDIGO	13
4. DISEÑO DEL PROGRAMA.....	16
5. SOFTWARE MELANIE	17
5.1 CONTROL DE CALIDAD	18
5.2 DISEÑO EXPERIMENTAL	18
5.3 CONFIGURACIÓN DE ALINEACIÓN	19
5.4 ALINEACIÓN	19
5.5 DETECCIÓN	20
5.6 REVISIÓN	20
5.7 RESULTADOS	21
5.8 RECOLECCIÓN.....	21
6. EXPERIMENTACIÓN	22
6.1 A_T1_Gel1	25
6.2 A_T2_Gel1	29
6.3 B_T1_Gel1	33
6.4 B_T2_Gel1	37
CONCLUSIONES EXPERIMENTACIÓN.....	41
7. CONCLUSIONES.....	43
8. BIBLIOGRAFÍA.....	45
A1. CÓDIGO PROGRAMA PROPIO	47
A2. METODOLOGÍA DE ELABORACIÓN DEL TD Y CARGA DE TRABAJO.....	50

LISTA DE FIGURAS

Ilustración 1 - Ejemplo de segmentación (Implementación del algoritmo WaterShed para el análisis de imágenes médicas, 2011).....	7
Ilustración 2 - Líneas divisoria, mínimos y cuencas (Implementación del algoritmo WaterShed para el análisis de imágenes médicas, 2011).....	8
Ilustración 3 - Sobresegmentación (Implementación del algoritmo WaterShed para el análisis de imágenes médicas, 2011).....	9
Ilustración 4 - Aplicación de marcadores a la técnica (Mengual, 2014).....	9
Ilustración 5 - Control de calidad (Swiss Institute of Bioinformatics, s.f.).....	18
Ilustración 6 - Diseño experimental (Swiss Institute of Bioinformatics, s.f.)	18
Ilustración 7 - Configuración de alineación (Swiss Institute of Bioinformatics, s.f.)	19
Ilustración 8 – Alineación (Swiss Institute of Bioinformatics, s.f.).....	19
Ilustración 9 – Detección (Swiss Institute of Bioinformatics, s.f.)	20
Ilustración 10 – Revisión (Swiss Institute of Bioinformatics, s.f.)	20
Ilustración 11 – Resultados (Swiss Institute of Bioinformatics, s.f.).....	21
Ilustración 12 – Recolección (Swiss Institute of Bioinformatics, s.f.)	21
Ilustración 13 - B_T2_Gel1.....	23
Ilustración 14 - Resultado Melanie Imagen 1.....	25
Ilustración 15 - Resultado implementación propia WaterShed A_T1_Gel1	25
Ilustración 16 - Histograma imagen preprocesada por Melanie.....	26
Ilustración 17 - Histograma imagen sin preprocesar	27
Ilustración 18 - ROI A_T1_Gel1 - Melanie	28
Ilustración 19 - ROI A_T1_Gel1 - Software propio	28
Ilustración 20 - Resultado Melanie A_T2_Gel1	29
Ilustración 21 - Resultado implementación propia WaterShed A_T2_Gel1	29
Ilustración 22 - Histograma imagen preprocesada por Melanie.....	30
Ilustración 23 - Histograma imagen sin preprocesar	31
Ilustración 24 - ROI A_T2_Gel1 - Melanie	32
Ilustración 25 - ROI A_T2_Gel1 - Software propio	32
Ilustración 26 - Resultado Melanie B_T1_Gel1	33
Ilustración 27 - Resultado implementación propia WaterShed B_T1_Gel1	33
Ilustración 28 - Histograma imagen preprocesada por Melanie.....	34
Ilustración 29 - Histograma imagen sin preprocesar	35
Ilustración 30 - ROI B_T1_Gel1 - Melanie	36
Ilustración 31 - ROI B_T1_Gel1 - Software propio	36
Ilustración 32 - Resultado Melanie B_T2_Gel1	37
Ilustración 33 - Resultado implementación propia WaterShed B_T2_Gel1	37
Ilustración 34 - Histograma imagen preprocesada por Melanie.....	38
Ilustración 35 - Histograma imagen sin preprocesar	39
Ilustración 36 - ROI B_T2_Gel1 - Melanie	40
Ilustración 37 - ROI B_T2_Gel1 - Software propio	40

RESUMEN

El objetivo de este trabajo consiste en estudiar la viabilidad de uso e implementación del algoritmo WaterShed, un algoritmo de segmentación por crecimiento de semillas para el estudio de imágenes biomédicas. Se estudia un artículo en el que se muestra una revisión bibliográfica del algoritmo, incluyendo una sencilla implementación en pseudocódigo. En dicho artículo, se prueba analizando manchas de proteínas en imágenes de electroforesis en geles de dos dimensiones.

Palabras clave: WaterShed, Segmentación, Implementación, Electroforesis, Geles de dos dimensiones

ABSTRACT

The goal of this work is to study the feasibility of using and implementing the WaterShed algorithm, a seed growth segmentation algorithm for the study of biomedical images. An article is studied in which a bibliographic review of the algorithm is shown, including a simple implementation in pseudocode. In that article, it is tested by analyzing protein spots in two-dimensional gel electrophoresis images.

Keywords: WaterShed, Segmentation, Implementation, Electrophoresis, Two-dimensional gel

1. INTRODUCCIÓN

El tratamiento de imágenes se ha convertido en una práctica muy habitual en la medicina moderna. Esto ha sido así gracias a la utilización de algoritmos que permiten procesar las imágenes para obtener información relevante de las mismas.

Un tipo de algoritmos muy utilizados son los de segmentación de la imagen a partir de los cuales se pueden extraer partes “útiles” de las imágenes a partir de la imagen base.

En este trabajo, vamos a contemplar una revisión del algoritmo WaterShed basado en la segmentación por crecimiento de semillas. Esta técnica fue introducida por C. Latuejeoul como “Transformación de WaterSehd” y más adelante fue mejorada por S. Beucher y denominada “WaterShed” finalmente (Implementación del algoritmo WaterShed para el análisis de imágenes médicas, 2011).

Este algoritmo se basa en una morfología matemática que permite extraer las fronteras de las regiones que hay en una imagen. Su consideración como un algoritmo de segmentación por crecimiento de semillas se debe a que este algoritmo clasifica los píxeles según su proximidad espacial, el gradiente de sus niveles de gris y la homogeneidad de sus texturas (Implementación del algoritmo WaterShed para el análisis de imágenes médicas, 2011).

Gracias a esto, la imagen puede visualizarse como una representación topográfica de un terreno, donde a cada pixel de la imagen se le asocia una altura en función de su nivel de gris correspondiente. Una vez representada la altura de la imagen, se utiliza un proceso de inundación desde los niveles más bajos (cuencas) hasta que las aguas de las diferentes cuencas se unen (Implementación del algoritmo WaterShed para el análisis de imágenes médicas, 2011).

Al tratarse de un algoritmo flexible y eficiente, es ampliamente utilizado para el tratamiento de imágenes en general, además de ser utilizado para el tratamiento de imágenes médicas destacando en el análisis de manchas de proteínas en imágenes de electroforesis en geles de dos dimensiones.

Basándonos en el artículo (Implementación del algoritmo WaterShed para el análisis de imágenes médicas, 2011) vamos a comparar la implementación del pseudocódigo ofrecido con la implementación existente del algoritmo WaterShed integrado en el módulo de segmentación de Slicer3D.

2. MARCO TEÓRICO DEL PROBLEMA

En esta parte del documento, vamos a desarrollar el marco teórico que envuelve al algoritmo WaterShed, así como un pseudo-código que permitiría realizar la implementación de dicho algoritmo.

2.1 SEGMENTACIÓN DE IMÁGENES

La segmentación de imágenes divide una imagen en sus partes constituyentes u objetos. Normalmente, se suele segmentar las imágenes para separar las partes de interés del resto de la imagen, de forma que el nivel de separación dependerá de la aplicación concreta que se le quiera dar a la imagen.

Por otra parte, la segmentación orientada a regiones consiste en el crecimiento de regiones agrupando píxeles adyacentes que presentan características o propiedades similares entre ellos.

Para poder realizar este proceso, se parte de un conjunto de puntos, los cuales se utilizan como semillas para dichas regiones. Estas semillas pueden ser escogidas manualmente o de forma automática.

La imagen es dividida en regiones conexas, donde cada región posee propiedades que las diferencia de las demás y que permite su procesamiento independiente. Este comportamiento puede observarse en la figura 1.



Ilustración 1 - Ejemplo de segmentación (Implementación del algoritmo WaterShed para el análisis de imágenes médicas, 2011)

Como se ha dicho en la introducción, la técnica WaterShed se considera una técnica de segmentación basada en el crecimiento de semillas, donde se clasifican los píxeles según su proximidad espacial, el gradiente de sus niveles de gris y la homogeneidad de sus texturas. Es por esto que se toma como una técnica de detección de contornos y crecimiento de regiones al mismo tiempo.

2.2 ALGORITMO WATERSHED

En geografía, watershed (cuenca) es la cordillera que divide áreas drenadas por diferentes sistemas fluviales. La transformación watershed aplica estas ideas a procesamiento de imágenes de manera que resuelve problemas de segmentación (Mengual, 2014).

En lo que se refiere a la segmentación, para conseguir un buen funcionamiento del algoritmo es utilizar una imagen que sea fácilmente transformable a la representación necesaria. En este caso, es necesaria una imagen en escala de grises.

La técnica considera una imagen en escala de grises como una imagen topográfica de un relieve, donde a cada pixel se le asocia como valor de altura su nivel de gris. Además, se incorpora un proceso de inundación de valles desde los niveles más bajos hasta que las aguas se unen, tal y como hemos visto en la introducción (Implementación del algoritmo WaterShed para el análisis de imágenes médicas, 2011).

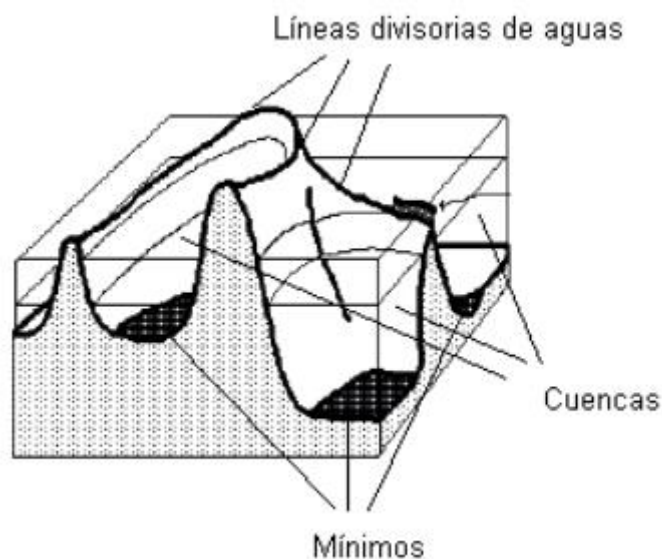


Ilustración 2 - Líneas divisoria, mínimos y cuencas (Implementación del algoritmo WaterShed para el análisis de imágenes médicas, 2011)

Las líneas divisorias de agua forman los contornos de los objetos de la imagen, dando lugar a la segmentación de la imagen y que corresponden además a zonas de elevada intensidad de gris.

El problema es que en imágenes no homogéneas o con ruido se genera una sobresegmentación debido al elevado número de mínimos locales y generando así muchas zonas que realmente no son de interés para el análisis, como se puede apreciar en la figura 3:

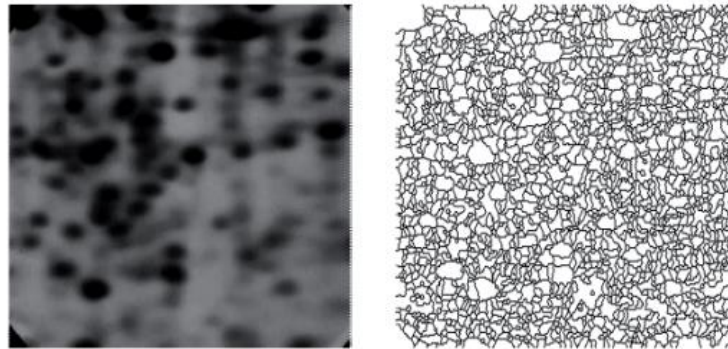


Ilustración 3 - Sobresegmentación (Implementación del algoritmo WaterShed para el análisis de imágenes médicas, 2011)

Este problema de sobresegmentación puede solucionarse utilizando semillas posicionadas en la imagen de una forma inteligente. En concreto, para el problema visto en la figura 3, si aplicamos estos marcadores obtendremos lo siguiente:

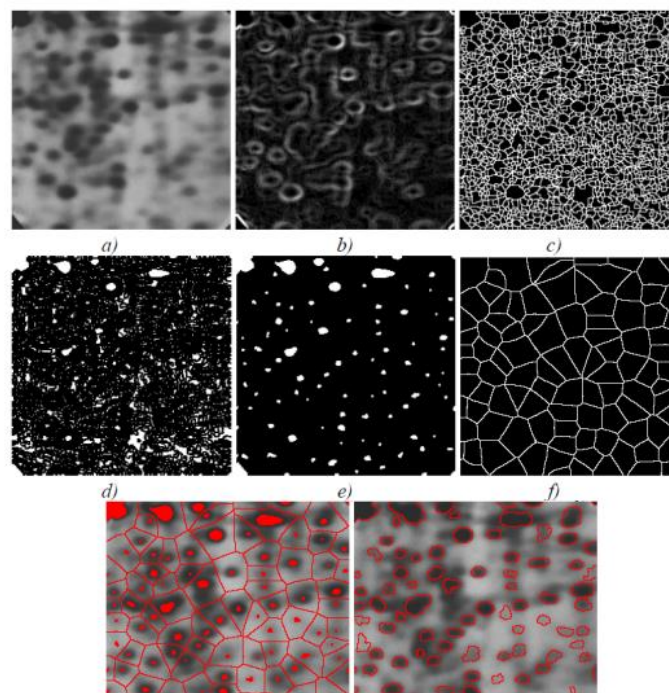


Ilustración 4 - Aplicación de marcadores a la técnica (Mengual, 2014)

Donde la imagen las diferentes imágenes corresponden a:

- **A**: Corresponde a la imagen de entrada para realizar su segmentación
- **B**: Corresponde a la imagen gradiente obtenida
- **C**: Corresponde a la salida obtenida por sobresegmentación
- **D**: Corresponde a los mínimos regionales del gradiente
- **E**: Corresponde a los marcadores internos
- **F**: Corresponde a los marcadores externos
- **G**: Corresponde a la unión entre marcadores externos e internos
- **H**: Corresponde al resultado de la segmentación

Este ejemplo ha sido extraído del artículo (Mengual, 2014) tal y como se indica en el nombre de la figura 4.

Otra posible solución y método de implementación es la segmentación mediante WaterShed estocástico, la cual consiste en establecer un marco probabilístico a la hora de detectar contornos con el fin de que sea un método robusto frente a variaciones en las condiciones de segmentación. Concretamente, consiste en introducir un enfoque estocástico utilizando marcadores aleatorios para construir una función densidad de probabilidad de contornos con la que se definirán las regiones más significativas (Mengual, 2014).

Dependiendo del problema que queramos analizar, se deberá definir una heurística distinta para elegir las semillas.

En cualquier caso, esta técnica está recomendada para imágenes homogéneas y con gradiente de intensidad débiles para obtener una segmentación óptima.

3. IMPLEMENTACIÓN WATERSHED

Existe una gran multitud de implementaciones del algoritmo WaterShed, entre ellas son destacables:

- WaterShed con un marcador
- WaterShed Estratificado
- WaterShed Estocástico
- WaterShed Pseudoestocástico

Estas diferentes implementaciones surgen a partir de los dos métodos de segmentación vistos en el apartado anterior: WaterShed por crecimiento de semillas y WaterShed estocástico.

En el caso concreto de este trabajo, vamos a centrarnos en la implementación y análisis del WaterShed por crecimiento de semillas. En concreto, se plantea la implementación de la propuesta de algoritmo de Vincent y Soille (1991) debido a la eficiencia del proceso de inundación y su menor coste computacional (Implementación del algoritmo WaterShed para el análisis de imágenes médicas, 2011).

3.1 IDEA GENERAL

El algoritmo comienza con la ordenación de los píxeles según el nivel de gris, para luego acceder a ellos mediante una pila FIFO (First In First Out).

A cada uno de los mínimos locales se le asigna una etiqueta diferente, la cual se propaga a todos los píxeles adyacentes en un nivel h . Se comienza analizando la imagen binarizada con un umbral h igual al mínimo valor de gris de la imagen hasta el máximo valor de gris.

En cada paso se analizan las componentes conectadas entre la binarización h y $h+1$. Al final de la inundación, todos los píxeles exceptuando las líneas WaterShed poseen una etiqueta que indica a que región pertenecen.

El proceso de inundación se realiza comparando la imagen f , binarizada con un umbral i , que denota $Z_i(f)$ y la imagen en un nivel superior $Z_{i+1}(f)$ desde $i=0$ hasta $i=N$, donde N es el máximo nivel de gris con que se representa la imagen.

En cada nivel i de la imagen f , se tienen mínimos regionales, denominados $m_i(f)$.

La vasija de la imagen f en el nivel de i , se denota como $W_i(f)$ que inicialmente corresponden a los mínimos regionales.

De esta forma, $W_{i+1}(f)$ es el resultado de la inundación de la vasija $W_i(f)$.

Durante el proceso de inundación, se pueden llegar a contemplar hasta tres casos diferentes:

1. **Crecimiento de una vasija ya existente en $Z_i(f)$.** En este caso, si el valor del pixel de las imágenes que se comparan, es decir $Z_{i+1}(f)$ es mayor que $Z_i(f)$, se produce el crecimiento de la vasija $W_i(f)$, lo que se traduce a que se genera $W_{i+1}(f)$. En caso de que sean iguales, la vasija permanece con el mismo tamaño.
2. **Aparición de una nueva vasija.** En este caso, se puede producir una nueva vasija dependiendo del valor de los píxeles de las imágenes que se comparan. Este fenómeno ocurre cuando $Z_{i+1}(f)$ es menor que $Z_i(f)$.
3. **Determinación de zonas de influencia.** Si la inundación del nivel $i+1$ une vasijas inundadas en el nivel i , se deben separar las regiones utilizando las zonas de influencia de cada componente conectada. En este caso, la vasija $i+1$ ($W_{i+1}(f)$) corresponde a una zona de influencia de la vasija i ($W_i(f)$).

(Implementación del algoritmo WaterShed para el análisis de imágenes médicas, 2011)

3.2 VARIABLES Y FUNCIONES

Comenzamos definiendo una variable G , la cual representa una malla digital $Z^2 \times Z^2$ que puede ser de algún tipo: cuadrado con conectividad 4 u 8; Hexadecimal con conectividad 6.

$NG(p)$ corresponde al conjunto de los vecinos de un pixel p con respecto a G , de forma que: $NG(p) = \{p' \in Z^2, (p,p') \in G\}$

La estructura de datos FIFO que utiliza el algoritmo y que contiene píxeles ordenados según su nivel de gris, permite realizar las siguientes tres operaciones:

1. **Agregar(p):** Añade el pixel p a la pila
2. **Primero():** Devuelve el pixel que se encuentre en la cima y lo elimina
3. **Vacio():** Devuelve true si la pila se encuentra vacía y false en caso contrario

(Implementación del algoritmo WaterShed para el análisis de imágenes médicas, 2011)

3.3 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 :

$$\forall p \in D_{im_0}, im_0(p) = INIT;$$

-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()
    }
  }
}
```

```

Foreach pixel  $p' \in NG(p)$  {
    If ( $im_d(p') < actual\_dist$  && ( $im_0(p') > 0$ ) ||
    ( $im_0(p') == WSHED$ )){
        /* i.e,  $p'$  ya pertenece a una vasija
        etiquetado o a el watershed*/
        If ( $im_0(p') > 0$ ){
            If ( $im_0(p) == MASK$  ||  $im_0(p) ==$ 
            WSHED) {
                 $im_0(p) = im_0(p')$ 
            } else {
                 $im_0(p) = WSHED$ 
            }
        } else if  $im_0(p') == MASK$  &&  $im_d(p') == 0$  {
             $im_d(p') = Actual\_dist + 1$ 
            Agregar( $p'$ )
        }
    }
}
/* Chequear si nuevo minimo ha sido descubierto*/
Foreach pixel  $p$  tal que  $im_i(p) = h$  {
     $im_d(p) = 0$ 
    If  $im_0(p) == MASK$  {
        Etiqueta_actual++
        Agregar( $p$ )
         $im_0(p) = etiqueta\_actual$ 

        While Vacio() == false {
             $P' = Primero()$ 
            Foreach pixel  $p'' \in NG(p')$  {
                If( $im_0(p'') == MASK$ ) {
                    Agregar( $p''$ )
                     $im_0(p'') = etiqueta\_actual$ 
                }
            }
        }
    }
}
}

```

(Implementación del algoritmo WaterShed para el análisis de imágenes médicas, 2011)

4. DISEÑO DEL PROGRAMA

El análisis de una imagen biomédica no se realiza únicamente aplicando un algoritmo (en este caso WaterShed) directamente sobre la imagen, sino que es necesario realizar un procesamiento previo para poder obtener unos buenos resultados.

En base a esto, el procesamiento previo para las imágenes que queremos segmentar utilizar el algoritmo WaterShed es algo relativamente estándar.

El flujo de diseño del pre-procesamiento para el WaterShed es el siguiente:

1. Se realiza un tratamiento de difuminado de la imagen si se requiere para eliminar datos irrelevantes para la segmentación buscada
2. Se convierte la imagen a escala de grises
3. Se aplica un umbral (threshold) para diferenciar el fondo de la imagen, el cual no es de interés
4. Se almacena el área de la imagen que se está seguro de que corresponde al fondo
5. Se busca el área de la imagen que se está seguro de que NO corresponde al fondo
6. Se extraen las regiones entre el fondo y el no fondo que son desconocidas para su tratamiento
7. Aplicación de algoritmo WaterShed

Como vemos, el algoritmo WaterShed es utilizado en una parte del análisis, no es el análisis en sí. (Rubiales, 2021)

Aunque este proceso es algo muy estandarizado, existen una serie de parámetros que pueden hacer que el resultado de la segmentación sea diferente al que se espera, haciendo necesario el diseño específico del pre-procesamiento para cada tipo de análisis.

5. SOFTWARE MELANIE

Debido a que vamos a utilizar el software Melanie, veo necesaria realizar una pequeña introducción al mismo para describir su funcionamiento.

Según su página web, Melanie se basa en más de 35 años de experiencia en el campo del procesamiento de imágenes. Su historia comienza como un proyecto de doctorado en la universidad de Ginebra, en 1983, en el que se creó la base para el software que es utilizado a día de hoy por investigadores de todo el mundo.

El propósito de Melanie tiene la intención de fomentar descubrimientos confiables para una mejor ciencia, de forma que el software proporciona orientación de análisis de imágenes y herramientas flexibles para permitirle detectar diferencias reales en la expresión de proteínas con alta objetividad, sensibilidad y confianza.

Melanie es una solución software integral para la visualización, emparejamiento, detección, cuantificación y análisis de electroforesis en gel 2D e imágenes de Western blot. El flujo de trabajo de Melanie es el siguiente:

5.1 CONTROL DE CALIDAD

El primer paso consiste en la carga de imágenes que se van a analizar, donde además, se pueden obtener datos sobre la calidad de las mismas de una forma bastante detallada y ofrecida por el propio programa:

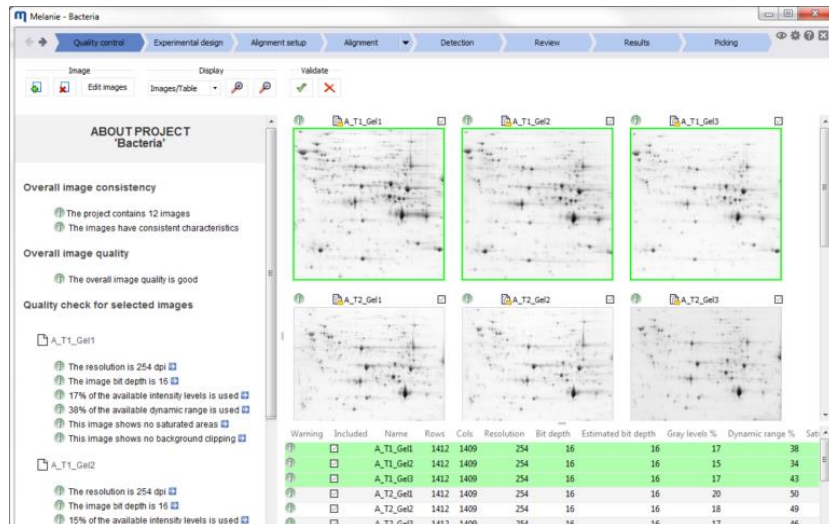


Ilustración 5 - Control de calidad (Swiss Institute of Bioinformatics, s.f.)

5.2 DISEÑO EXPERIMENTAL

El apartado correspondiente al diseño experimental nos permite crear uno de los diseños comunes, definir factores y niveles de factores, y asignar imágenes a los diferentes tratamientos. También podemos especificar variables adicionales para un diseño experimental coherente y equilibrado.

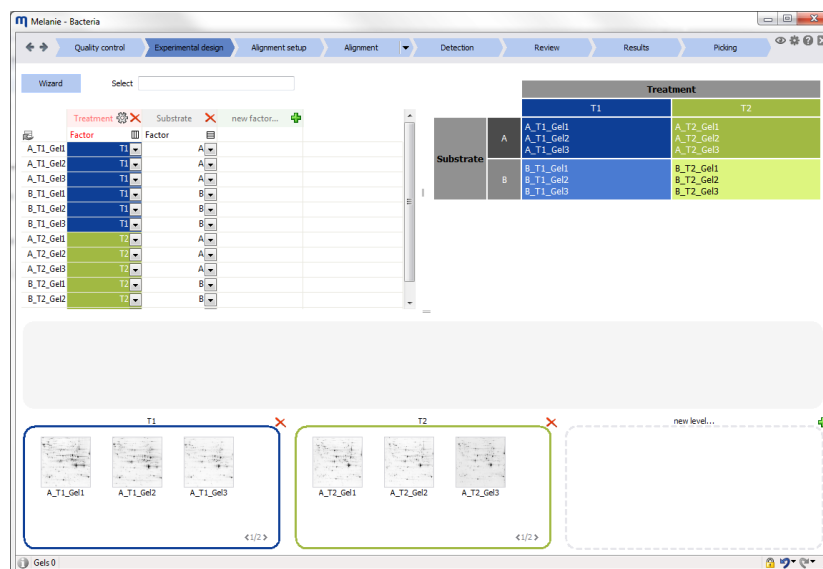


Ilustración 6 - Diseño experimental (Swiss Institute of Bioinformatics, s.f.)

5.3 CONFIGURACIÓN DE ALINEACIÓN

El apartado de configuración de alineación nos permite optimizar y configurar la eficiencia de la alineación dentro de los grupos de imágenes similares. Puede agrupar imágenes en función de factores definidos en el diseño experimental o crear su propia jerarquía de grupos en función del conocimiento sobre las imágenes.

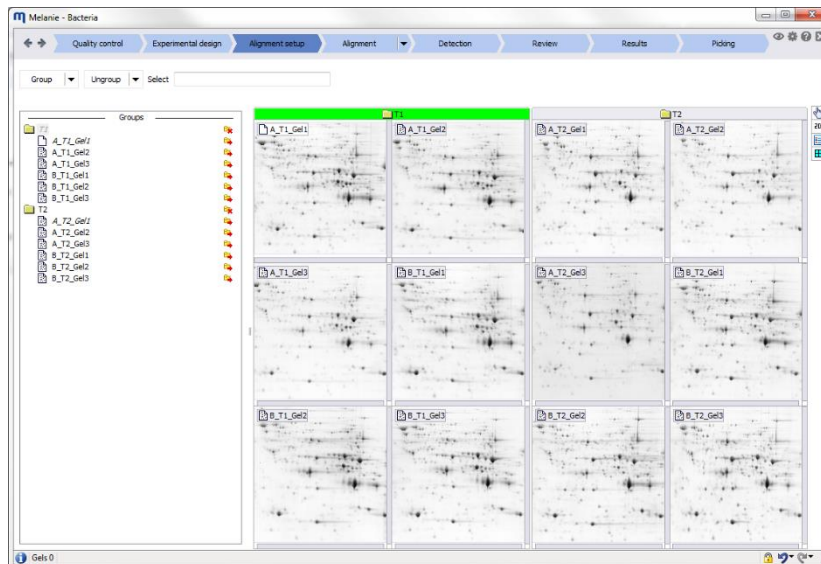


Ilustración 7 - Configuración de alineación (Swiss Institute of Bioinformatics, s.f.)

5.4 ALINEACIÓN

En esta fase podemos utilizar el algoritmo de alineación automática que nos aporta el programa para eliminar la variación posicional entre los diferentes geles utilizados.

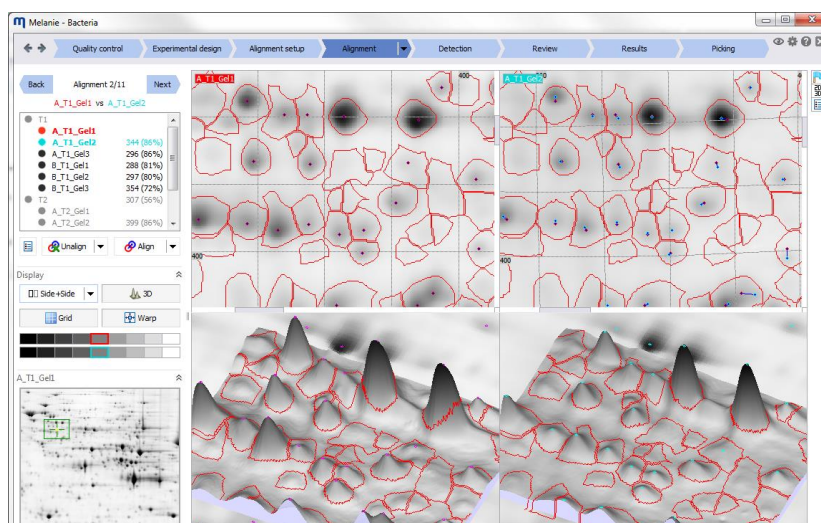


Ilustración 8 – Alineación (Swiss Institute of Bioinformatics, s.f.)

5.5 DETECCIÓN

En este paso se realiza la detección de las diferentes manchas que se encuentran en la imagen que hayamos cargado anteriormente.

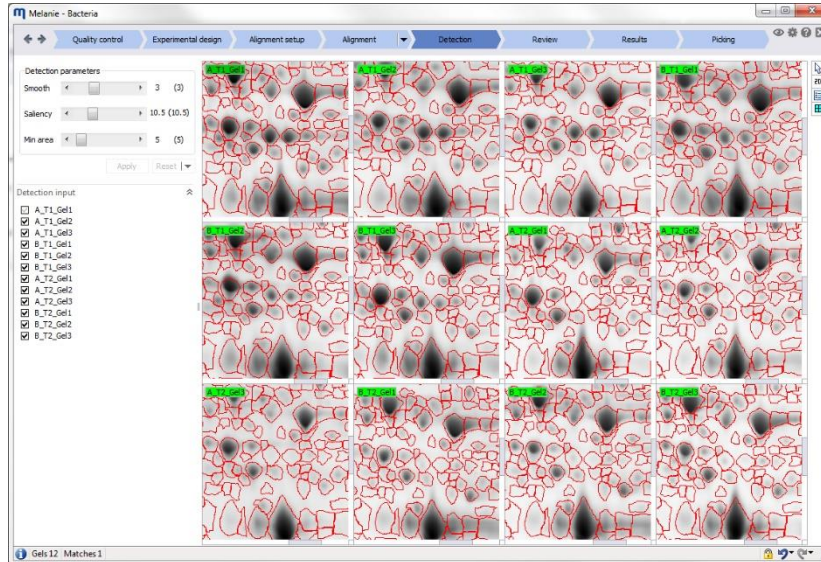


Ilustración 9 – Detección (Swiss Institute of Bioinformatics, s.f.)

5.6 REVISIÓN

En este apartado el software nos permite elegir diferentes manchas y escoger cuál de ellas queremos aislar para analizarlas más detenidamente.

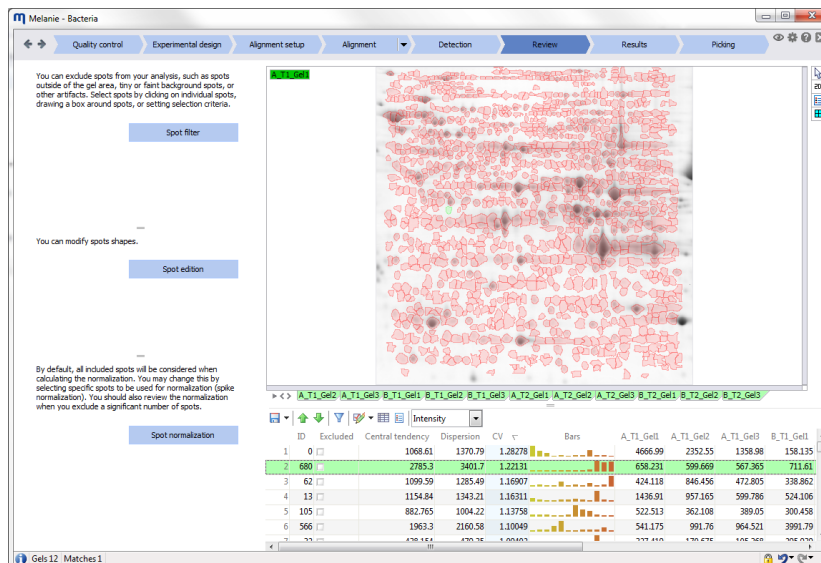


Ilustración 10 – Revisión (Swiss Institute of Bioinformatics, s.f.)

5.7 RESULTADOS

En este apartado se pueden explorar y validar los diferentes resultados obtenidos por la segmentación.



Ilustración 11 – Resultados (Swiss Institute of Bioinformatics, s.f.)

5.8 RECOLECCIÓN

En este apartado se pueden escoger los diferentes puntos de la segmentación de los cuales se quiere extraer la información permitiendo realizar un análisis posterior de los resultados.

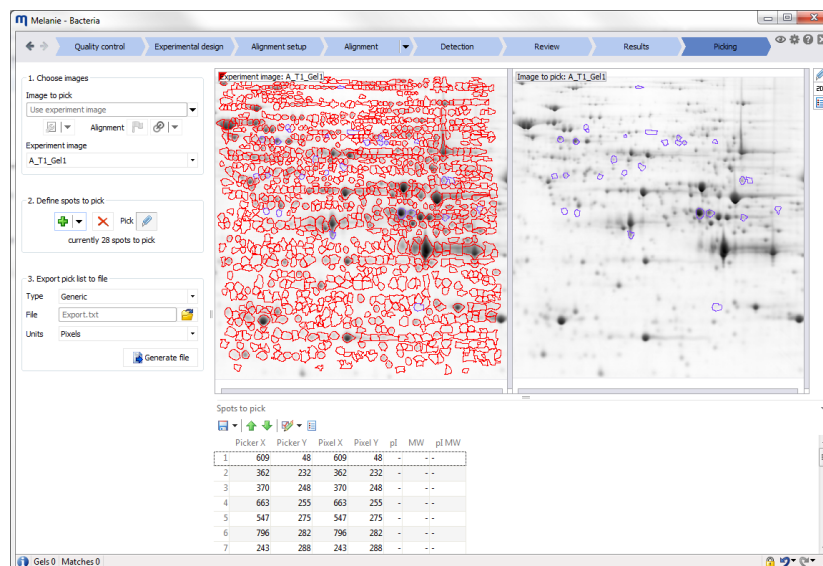


Ilustración 12 – Recolección (Swiss Institute of Bioinformatics, s.f.)

6. EXPERIMENTACIÓN

Para realizar esta parte se ha utilizado el programa especializado en análisis de geles en 2D llamado Melanie, presentado en el apartado anterior, del cual podemos encontrar más información en su página oficial, a la que se puede acceder pinchando en el siguiente [link](#).

Este es un programa de pago, pero el equipo de Melanie nos ha cedido un programa de prueba que contiene algunos ejemplos, aunque no presenta todas las funcionalidades que posee el programa completo. Entre estos ejemplos existen imágenes de geles en 2D los cuales utilizaremos para realizar la comparación entre el resultado obtenido por un software profesional como Melanie y un software propio.

El código utilizado para el software propio ha sido obtenido desde la página oficial de OpenCV donde existe un ejemplo de segmentación de imágenes mediante el algoritmo WaterShed. El código base que hay disponible en esta página no ha sido suficiente para realizar una segmentación que podamos considerar aceptable, debido a que cada segmentación necesita unos parámetros únicos. En este caso, únicamente ha sido necesario editar la distancia de transformación para determinar las zonas desconocidas y acotarlas se forma que esta distancia ha sido un 10% de la distancia máxima desde la zona conocida y parece que con ese cambio se consigue un resultado decente.

La idea de obtener estos resultados es la de compararla con los resultados obtenidos por el software Melanie, los cuales vamos a considerar nuestra verdad de base, ya que al tratarse de un software profesional podemos partir de la premisa de que los resultados serán buenos.

Vamos a realizar el análisis de cuatro imágenes diferentes para comparar los resultados de la forma dicha anteriormente. Las imágenes son:

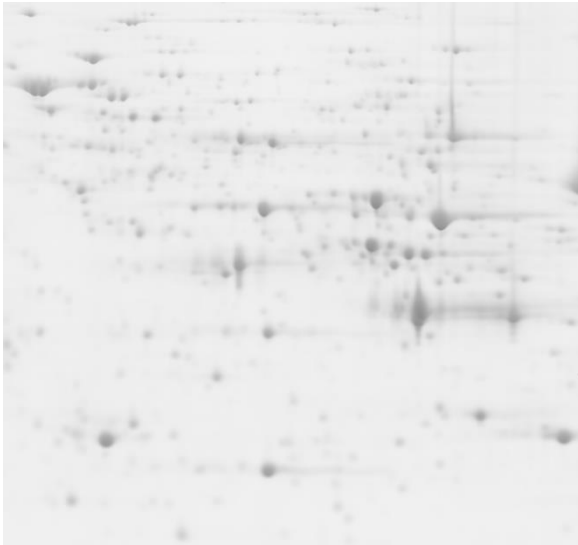


Ilustración 13 - A_T1_Gel1



Ilustración 13 - A_T2_Gel1



Ilustración 13 - B_T1_Gel1



Ilustración 13 - B_T2_Gel1

Como se puede apreciar, parecen imágenes muy similares, pero hay pequeñas diferencias en cada una de ellas que las hacen únicas.

Melanie nos ofrece información acerca de las imágenes cuando las cargamos en su apartado de calidad de imagen. La información obtenida en este caso para las cuatro imágenes es la siguiente:

Nombre	Anchura	Altura	Resolución
A_T1_Gel1	1409	1409	254
A_T2_Gel1	1409	1409	254
B_T1_Gel1	1409	1409	254
B_T2_Gel1	1409	1409	254

Nombre	Prof. Bit	Nivel de gris %	Rango Din %
A_T1_Gel1	16	17	38
A_T2_Gel1	16	20	50
B_T1_Gel1	16	17	39
B_T2_Gel1	16	18	43

Nombre	Min Value	Max Value
A_T1_Gel1	3524	28530
A_T2_Gel1	3500	36315
B_T1_Gel1	3308	28738
B_T2_Gel1	3132	31354

Nombre	Min gray	Max gray	Gray levels
A_T1_Gel1	3524	28530	11146
A_T2_Gel1	3500	36315	12933
B_T1_Gel1	3308	28738	10953
B_T2_Gel1	3132	31354	11733

Como vemos, la anchura, la altura, la resolución y la profundidad de bits de las diferentes imágenes es idéntica. Las diferencias aparecen en los colores y sobre todo en los niveles de gris, donde cada imagen presenta unos niveles diferentes.

6.1 A_T1_Gel1

Para la primera imagen, el resultado obtenido por el software Melanie es el siguiente:

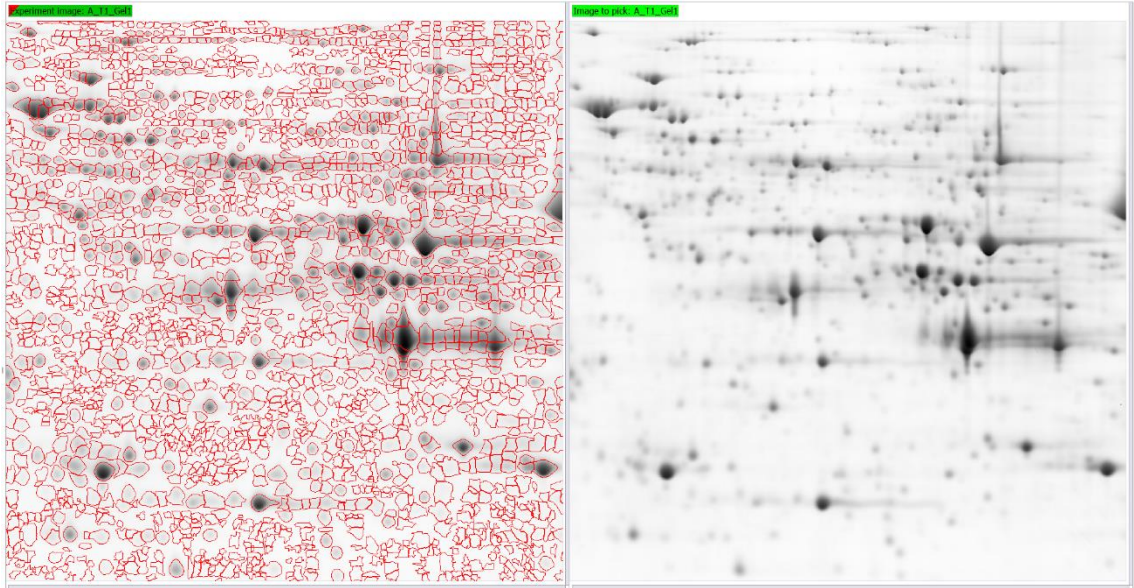


Ilustración 14 - Resultado Melanie Imagen 1

Vemos que consigue muy buena segmentación para la imagen, consiguiendo aislar de forma correcta los puntos de interés. Aunque, por otra parte, hay algunas manchas en las que se produce un poco de sobresegmentación, pero en general se consigue una segmentación muy buena.

El resultado que se obtiene con el programa desarrollado por nosotros utilizando la implementación de WaterShed de openCV es el siguiente:

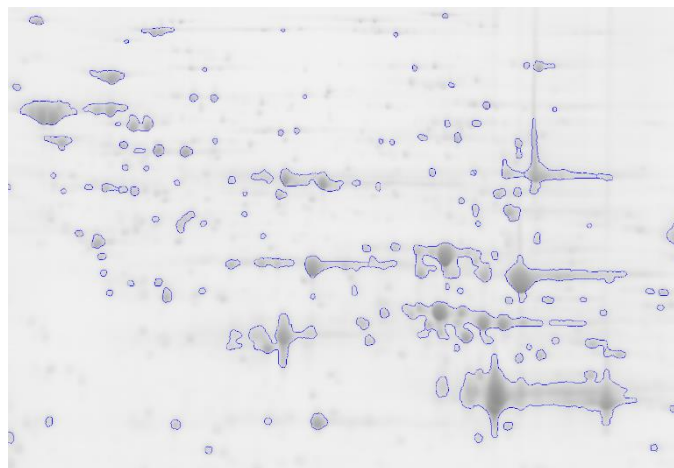


Ilustración 15 - Resultado implementación propia WaterShed A_T1_Gel1

Lo primero que podemos destacar es que vemos que Melanie nos ofrece dos imágenes de salida, una de ellas segmentada y otra sin ningún tipo de segmentación que corresponde a la imagen base que se ha utilizado para la segmentación. A simple vista, se puede observar que la imagen es más oscura en las zonas grises que la imagen original ofrecida como entrada. Si utilizamos GIMP para poder estudiar los histogramas de las imágenes obtenemos lo siguiente:

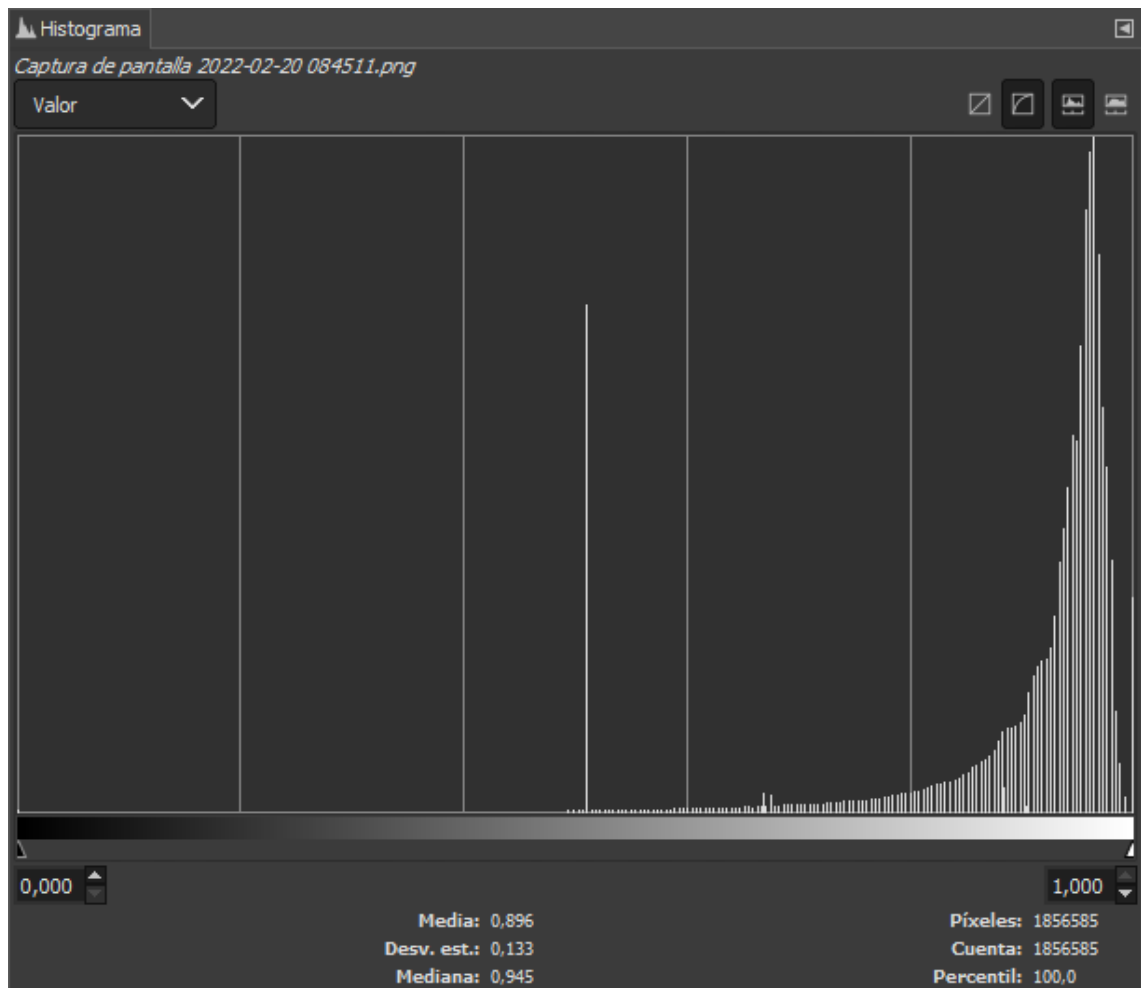


Ilustración 16 - Histograma imagen preprocesada por Melanie

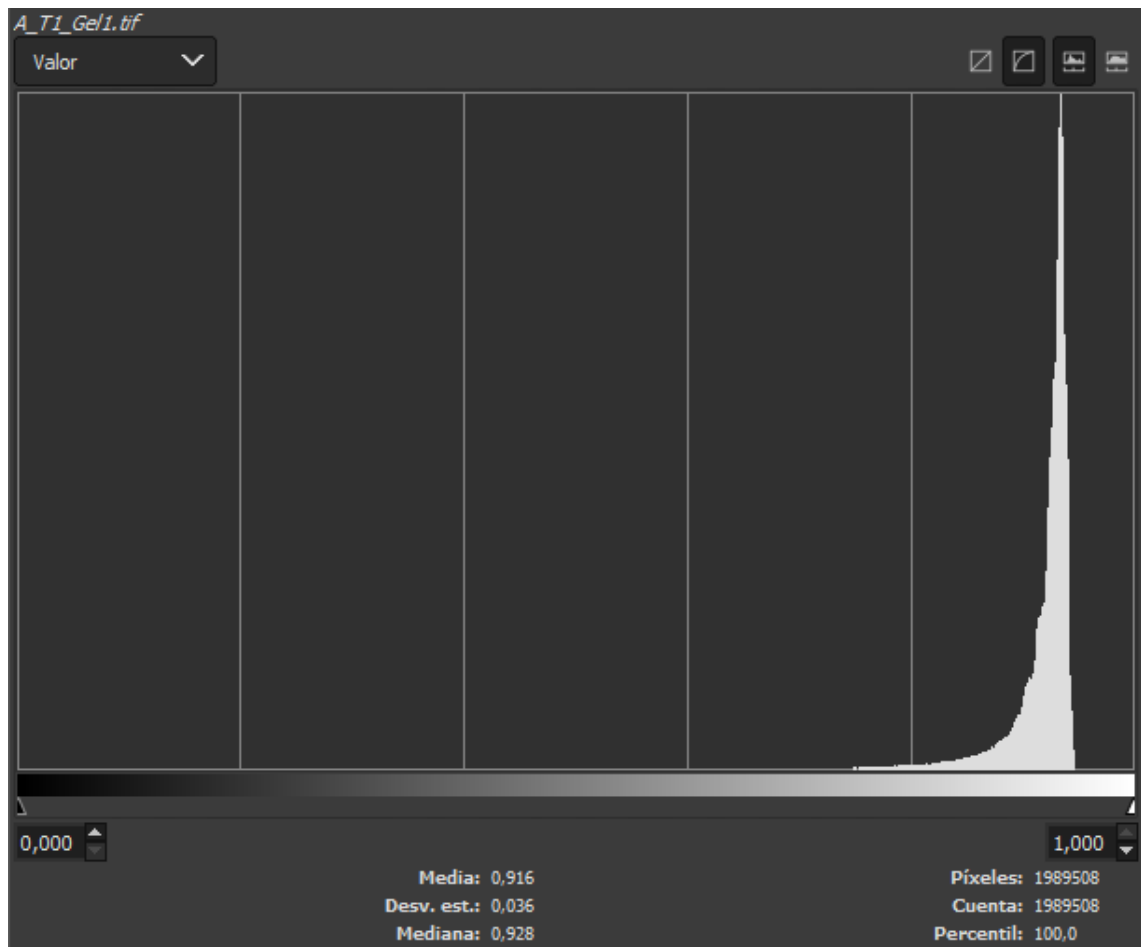


Ilustración 17 - Histograma imagen sin preprocesar

Podemos ver varios parámetros que reflejan la hipótesis anterior del cambio de imagen, pero el más claro de ellos es la media del histograma, donde en la imagen original es de 0.916 y en la imagen que proporciona Melanie como salida corresponde a 0.896.

Por otra parte, si nos detenemos más en la propia segmentación en sí misma, podemos observar que el software propio genera muchas menos regiones en comparación con el software Melanie. En concreto, las regiones generadas por cada uno de los softwares son:

- **Melanie:** 2012 regiones generadas
- **Software propio:** 144 regiones generadas

Si nos centramos en una de las manchas y hacemos un análisis más en profundidad de lo obtenido por los diferentes softwares, vemos lo siguiente:

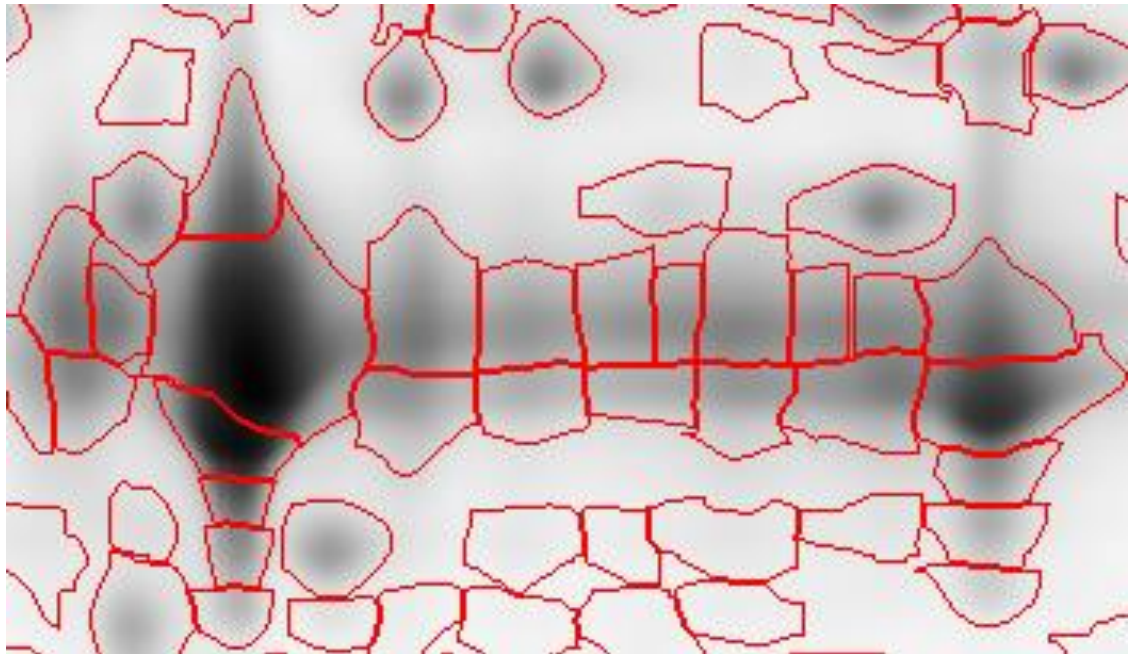


Ilustración 18 - ROI A_T1_Gel1 - Melanie

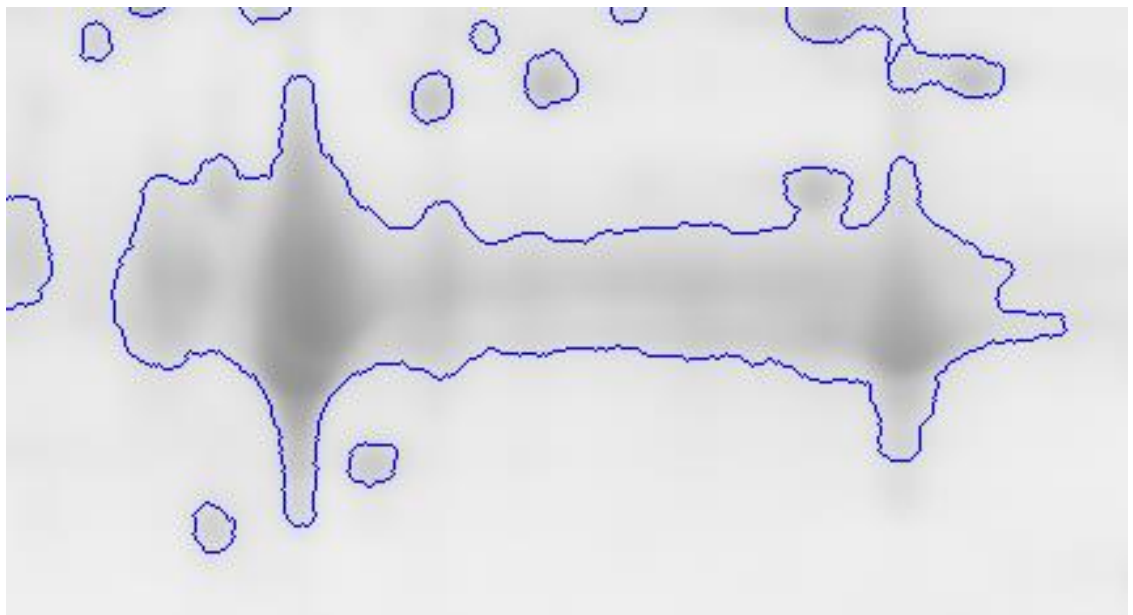


Ilustración 19 - ROI A_T1_Gel1 - Software propio

Como podemos ver, mientras que el software Melanie diferencia varias regiones en esa mancha, nuestro software propio la interpreta como una única región debido a que las dos manchas de proteínas más grandes poseen una conexión a modo de puente.

6.2 A_T2_Gel1

Para la segunda imagen, el resultado obtenido por el software Melanie es el siguiente:

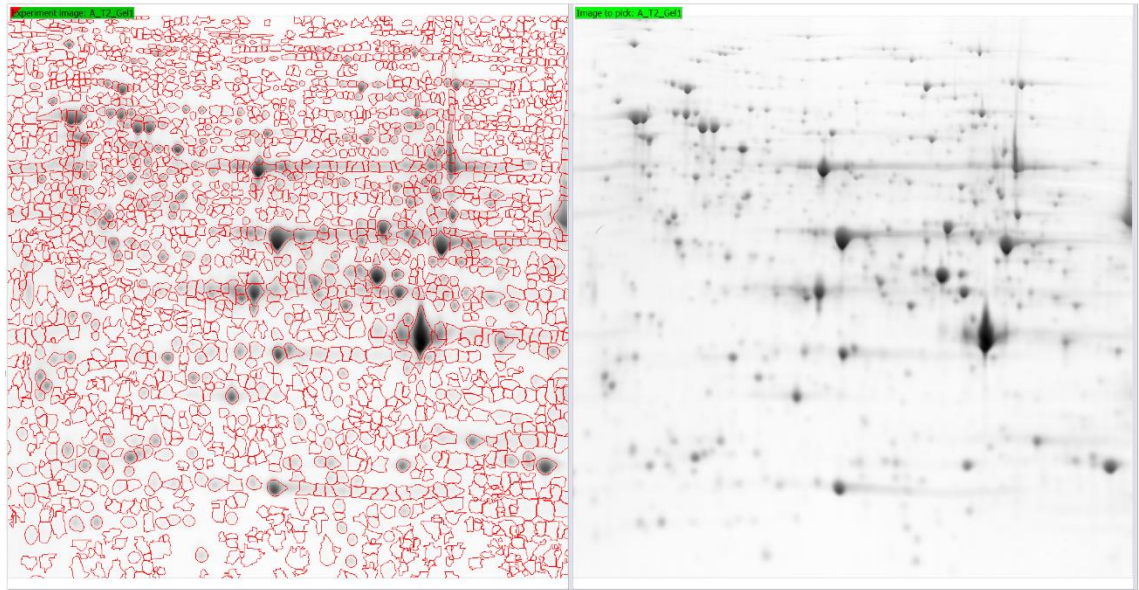


Ilustración 20 - Resultado Melanie A_T2_Gel1

Vemos que al igual que en el caso anterior, la segmentación obtenida para la imagen es muy buena, pero hay algunos casos en los que parece que se produce una sobresegmentación, ya que aparecen regiones que aparentemente no parecen tener nada dentro.

El resultado obtenido por el programa desarrollado por nosotros obtiene la siguiente imagen:

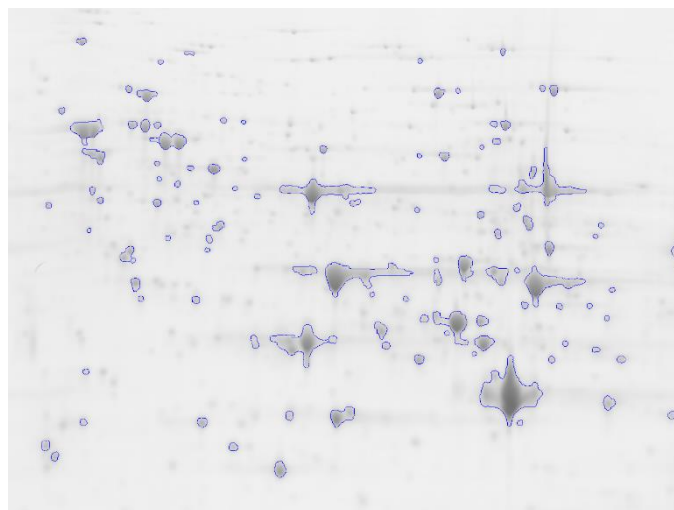


Ilustración 21 - Resultado implementación propia WaterShed A_T2_Gel1

Al igual que ocurre en el caso anterior con la primera imagen que hemos tratado, se puede apreciar que el software Melanie está realizando un preprocesamiento de la imagen añadiéndole oscuridad a las zonas de interés. Esto podemos comprobarlo utilizando el programa GIMP para analizar su histograma como hemos hecho en el caso anterior:

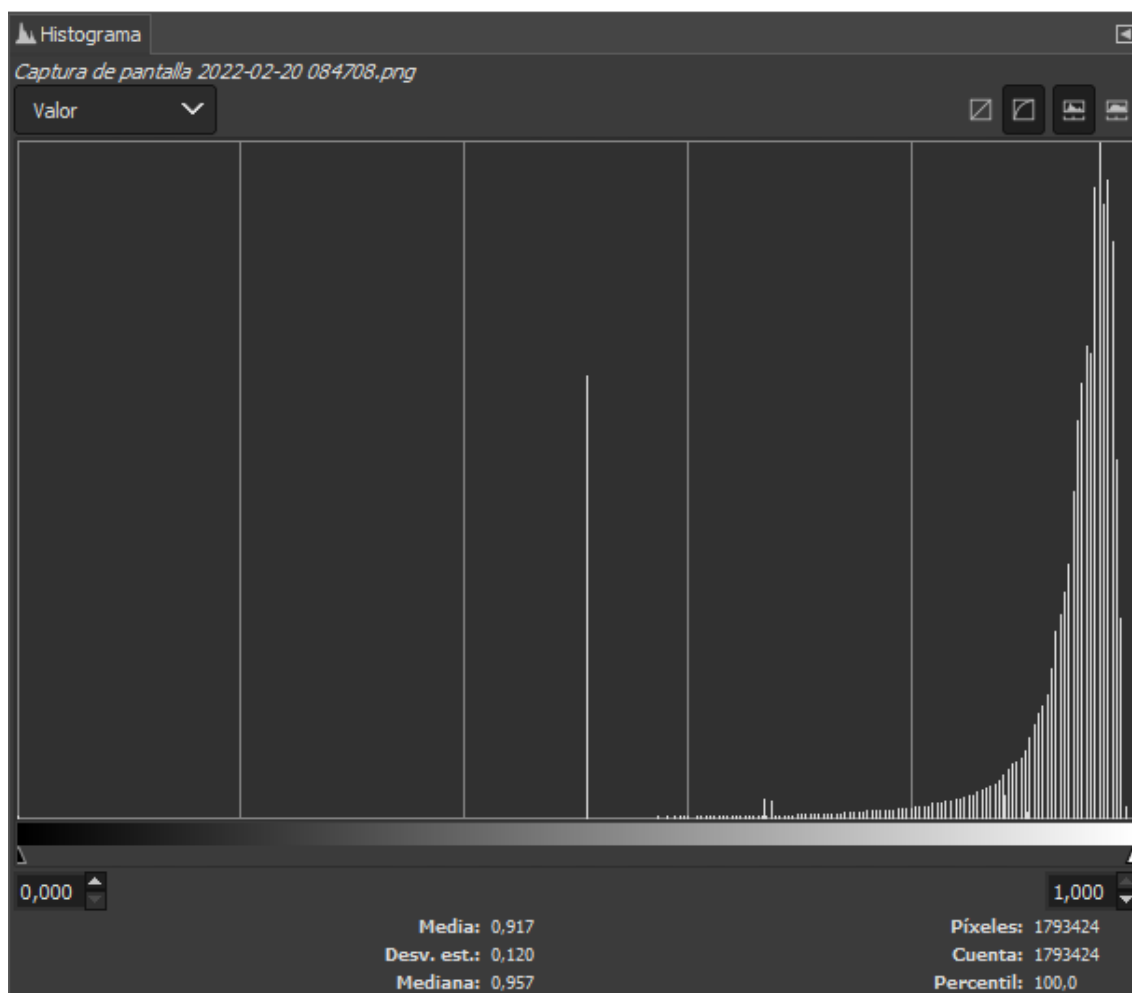


Ilustración 22 - Histograma imagen preprocesada por Melanie

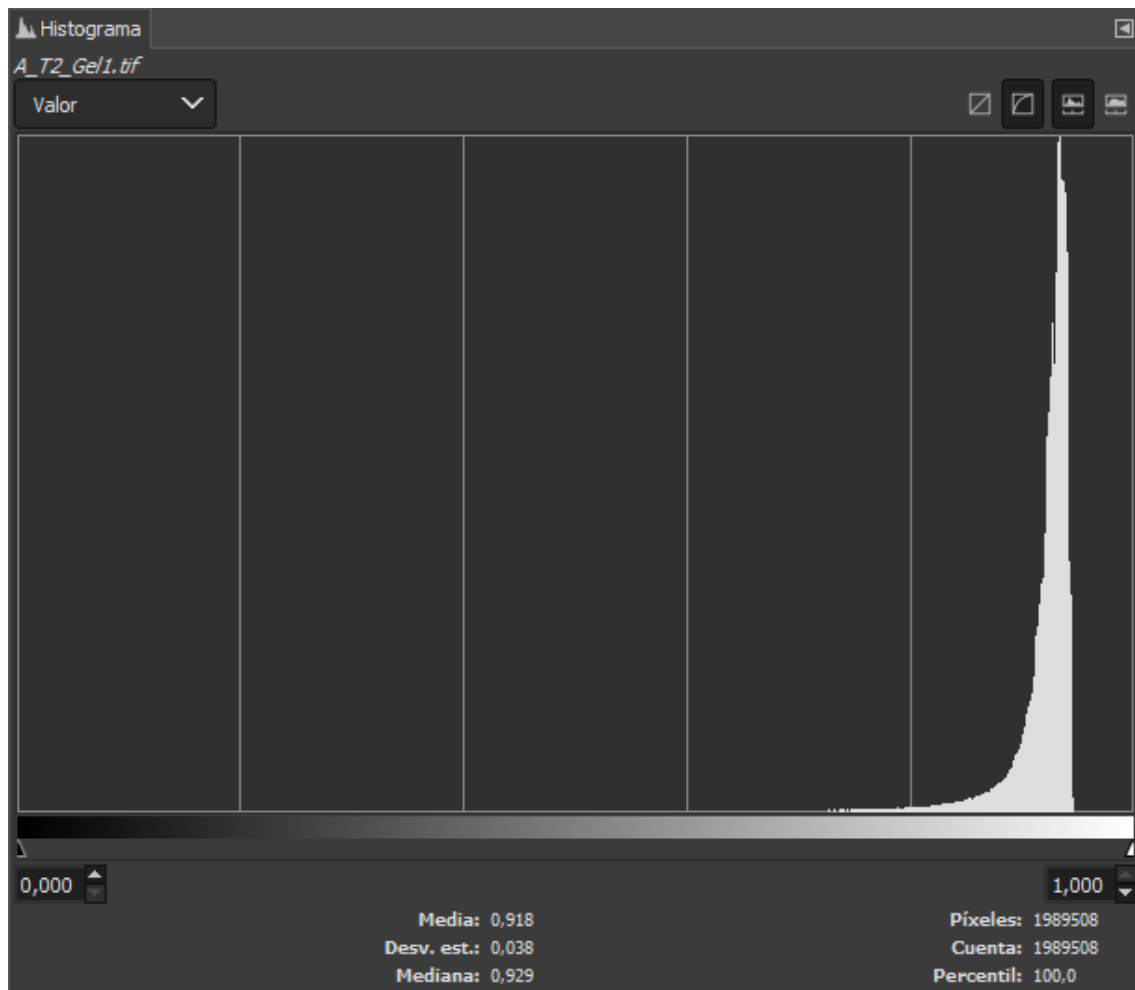


Ilustración 23 - Histograma imagen sin preprocesar

Al igual que en el caso anterior, podemos ver en los parámetros del histograma de las imágenes que la imagen preprocesada por Melanie posee una media de 0.917 contra la media de 0.918 de la imagen sin preprocesar. En este caso, la media parece apuntar a que no existe una gran diferencia entre imágenes, pero esto puede deberse a que existe un menor número de manchas de gran interés en comparación con la imagen anterior.

De igual forma, si miramos la representación del histograma, podemos ver en la imagen preprocesada por Melanie que existe un pico cercano a los colores más negros que no existe en la imagen sin preprocesar.

Si nos detenemos en la segmentación como en el caso anterior, podemos ver que el software propio genera muchas menos regiones que el software Melanie, pero para tener una idea más clara vamos a verlo con datos:

- **Melanie:** 1974 regiones generadas
- **Software propio:** 121 regiones generadas

Podemos ver claramente que el software propio genera muchas menos regiones que el software Melanie. Si nos centramos en alguna mancha igual que en el caso anterior, podemos ver lo siguiente:

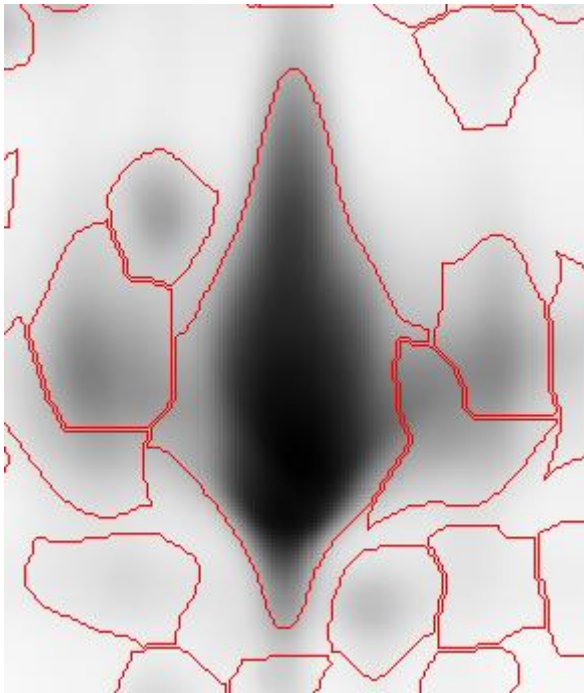


Ilustración 24 - ROI A_T2_Gel1 - Melanie

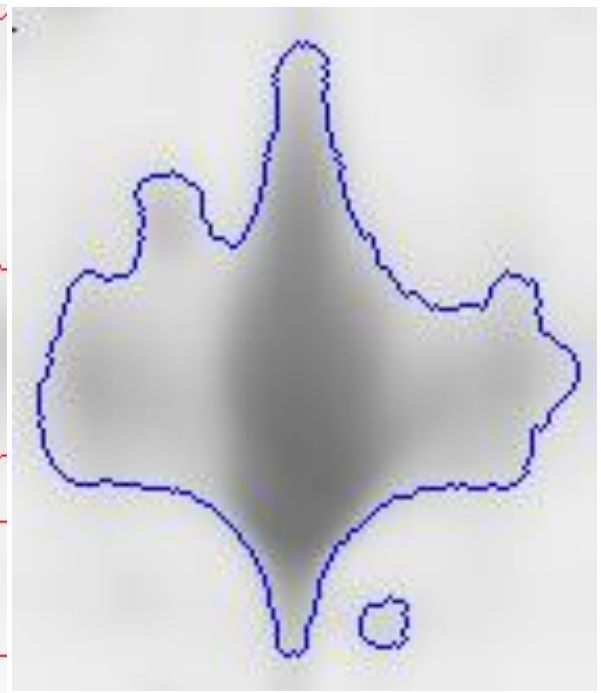


Ilustración 25 - ROI A_T2_Gel1 - Software propio

Como podemos observar que en este caso ocurre algo similar al caso anterior, donde existe una región que claramente es más profunda que las demás y regiones pequeñas a su alrededor que Melanie interpreta como separadas mientras que nuestro software las interpreta como una única región conectadas por algún tipo de camino más profundo debido a que existe un puente de gris entre las diferentes regiones.

6.3 B_T1_Gel1

Para la tercera imagen, el resultado obtenido por el software Melanie es el siguiente:

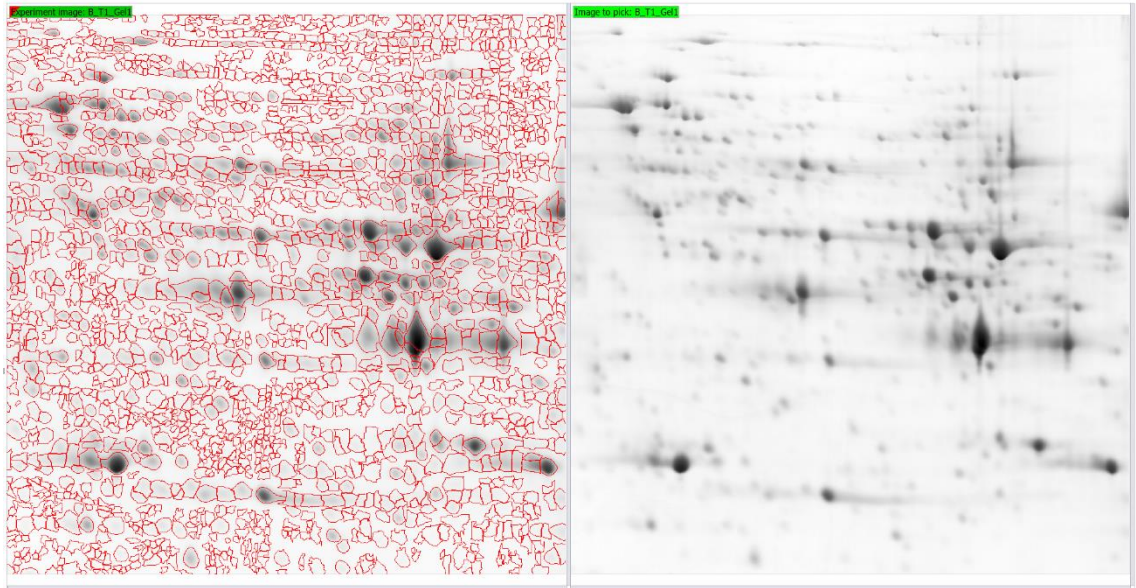


Ilustración 26 - Resultado Melanie B_T1_Gel1

Vemos que al igual que ocurre con los casos anteriores, la segmentación conseguida es buena, pero en las zonas que son completamente blancas existe una pequeña sobresegmentación.

El resultado obtenido por el programa desarrollado por nosotros es el siguiente:

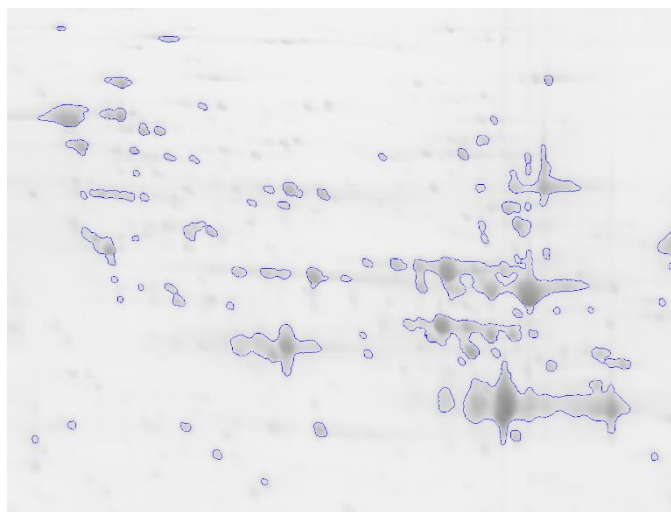


Ilustración 27 - Resultado implementación propia WaterShed B_T1_Gel1

Al igual que ocurre en el caso anterior con la primera imagen que hemos tratado, se puede apreciar que el software Melanie está realizando un preprocesamiento de la imagen añadiéndole oscuridad a las zonas de interés. Esto podemos comprobarlo utilizando el programa GIMP para analizar su histograma como hemos hecho en el caso anterior:

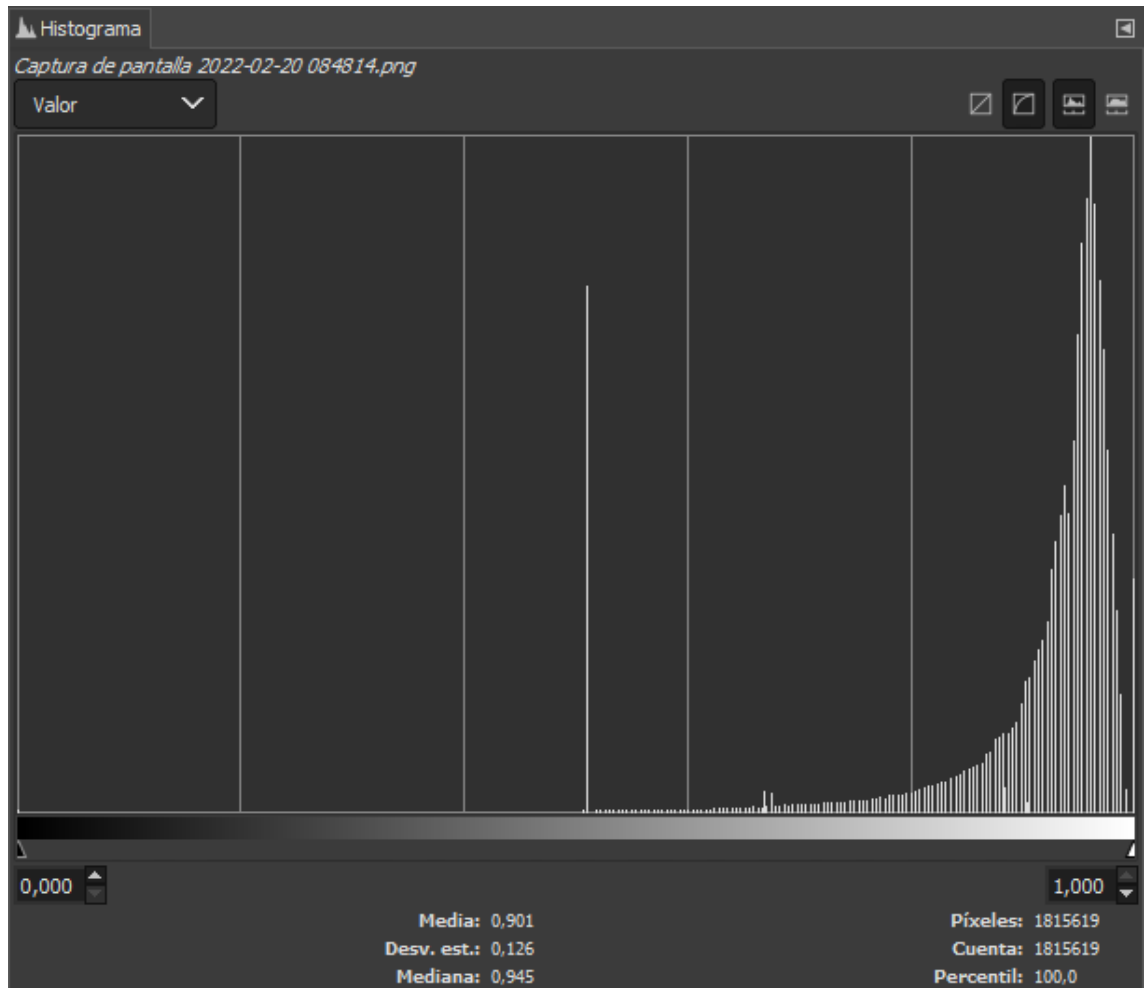


Ilustración 28 - Histograma imagen preprocesada por Melanie

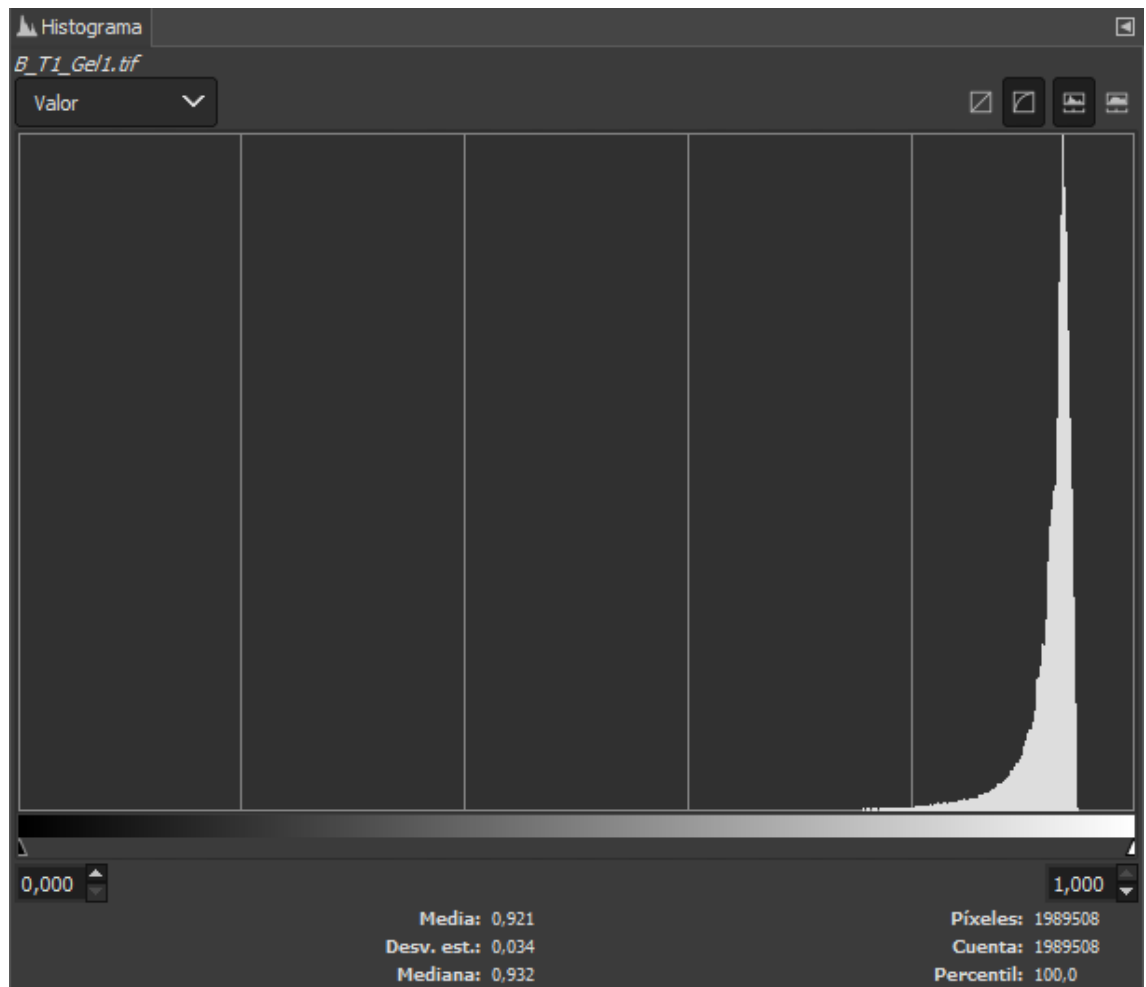


Ilustración 29 - Histograma imagen sin preprocesar

Al igual que en los casos anteriores, podemos ver claramente en los histogramas de las imágenes como la imagen preprocesada por Melanie posee una media inferior a la media de la imagen sin preprocesamiento, las cuales son 0.901 y 0.921 respectivamente.

Si nos detenemos en la segmentación como en el caso anterior, podemos ver que el software propio genera muchas menos regiones que el software Melanie, pero para tener una idea más clara vamos a verlo con datos:

- **Melanie:** 1836 regiones generadas
- **Software propio:** 95 regiones generadas

Podemos ver claramente que el software propio genera muchas menos regiones que el software Melanie. Si nos centramos en alguna mancha igual que en el caso anterior, podemos ver lo siguiente:

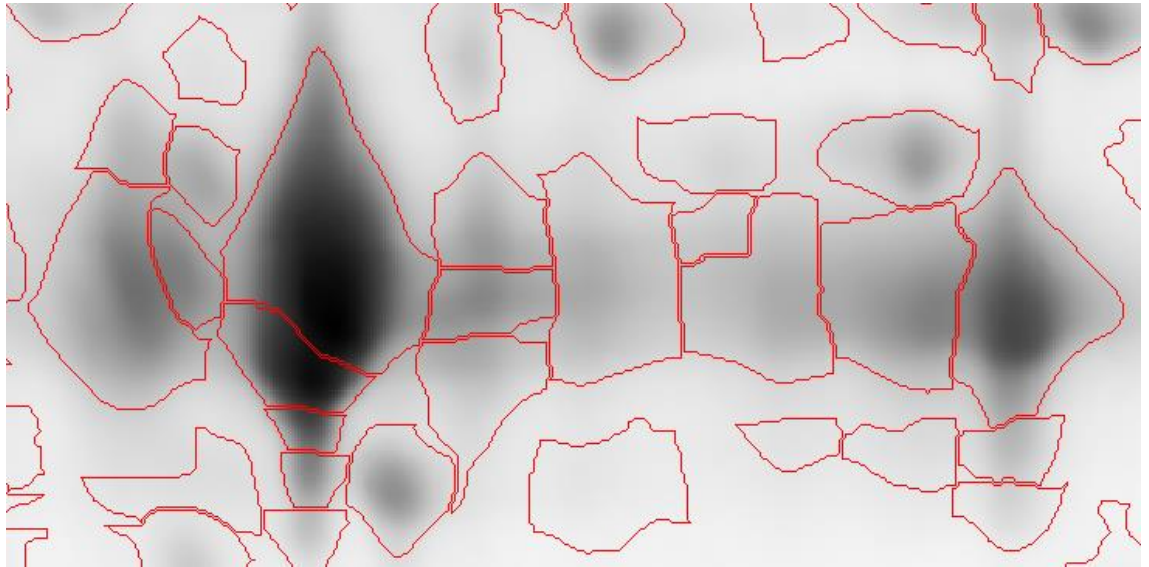


Ilustración 30 - ROI B_T1_Gel1 - Melanie

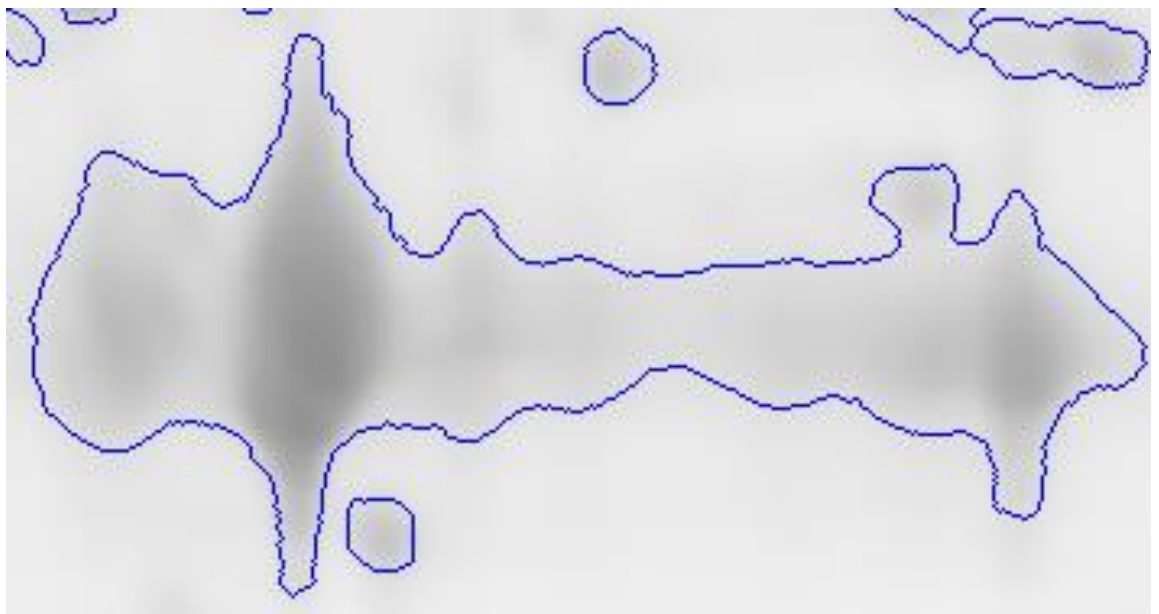


Ilustración 31 - ROI B_T1_Gel1 - Software propio

En este caso, ocurre lo mismo que en el caso de la primera imagen y es que podemos diferenciar dos zonas más profundas, las cuales Melanie ha segmentado en varias partes y nuestro software propio la ha tomado como una región entera debido a la conexión que existe mediante un tono de gris entre las dos manchas.

6.4 B_T2_Gel1

Para la cuarta imagen, el resultado obtenido por el software Melanie es el siguiente:

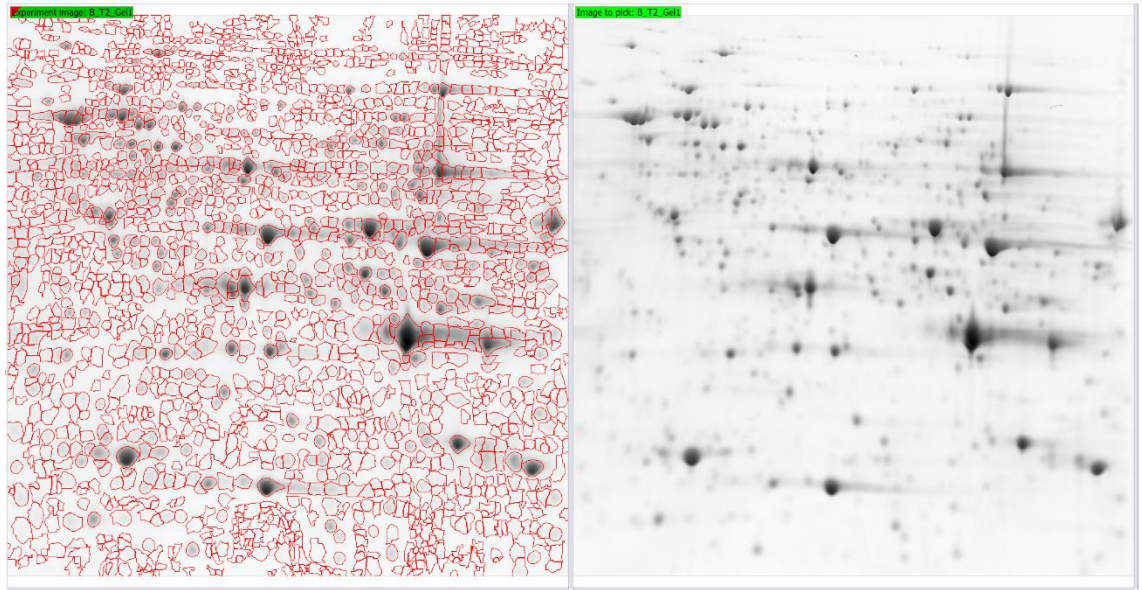


Ilustración 32 - Resultado Melanie B_T2_Gel1

En este caso, ocurre lo mismo que en los casos anteriores, se consigue muy buena segmentación en algunas zonas, pero ocurre que hay un exceso de segmentación en otras zonas.

El resultado obtenido por el software propio es el siguiente:

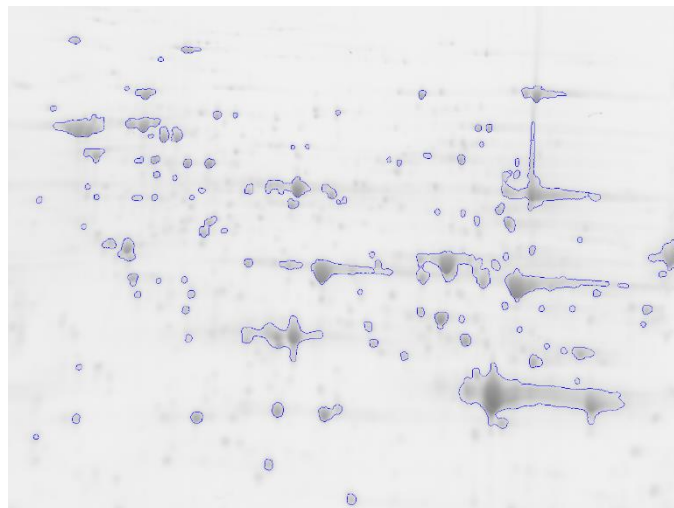


Ilustración 33 - Resultado implementación propia WaterShed B_T2_Gel1

Al igual que ocurre en el caso anterior con la primera imagen que hemos tratado, se puede apreciar que el software Melanie está realizando un preprocesamiento de la imagen añadiéndole oscuridad a las zonas de interés. Esto podemos comprobarlo utilizando el programa GIMP para analizar su histograma como hemos hecho en el caso anterior:

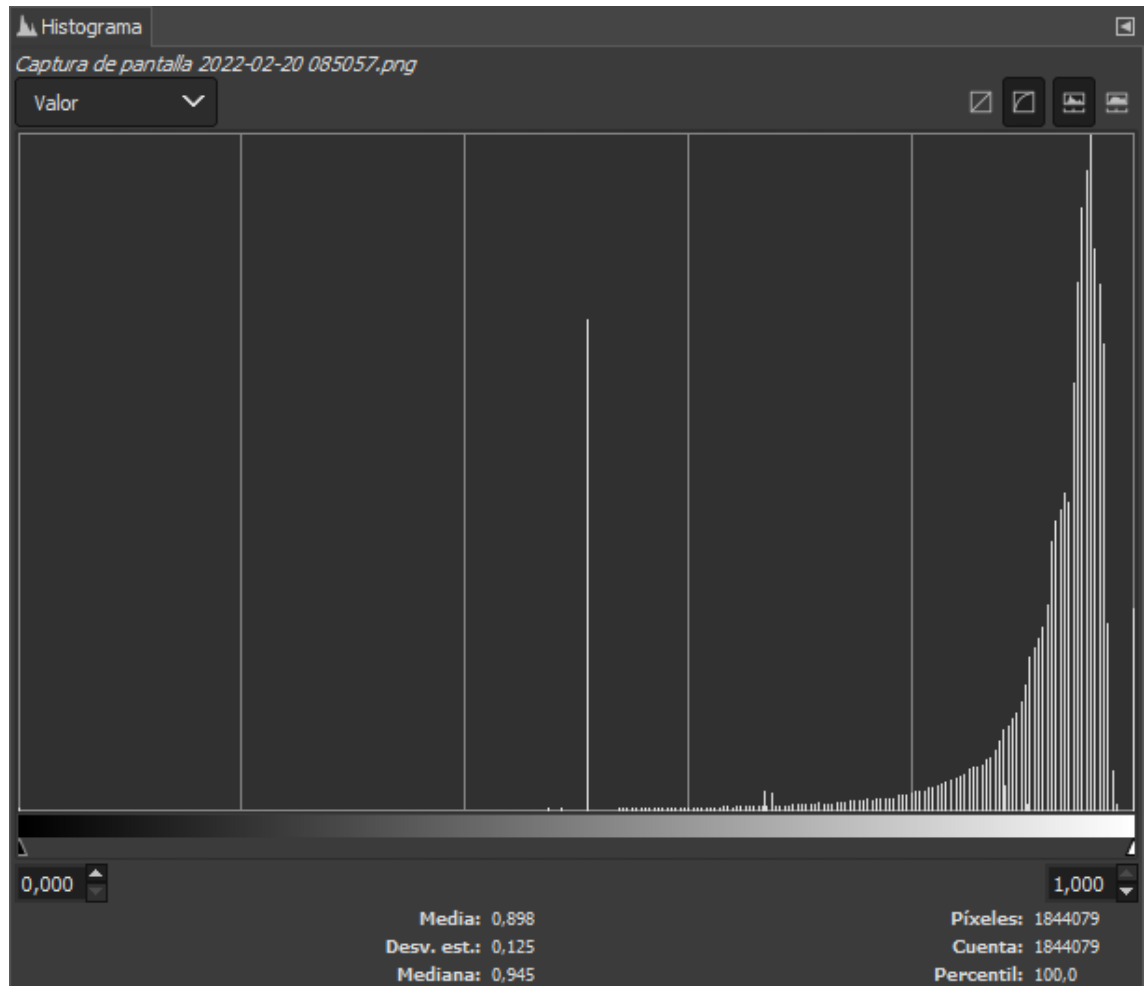


Ilustración 34 - Histograma imagen preprocesada por Melanie

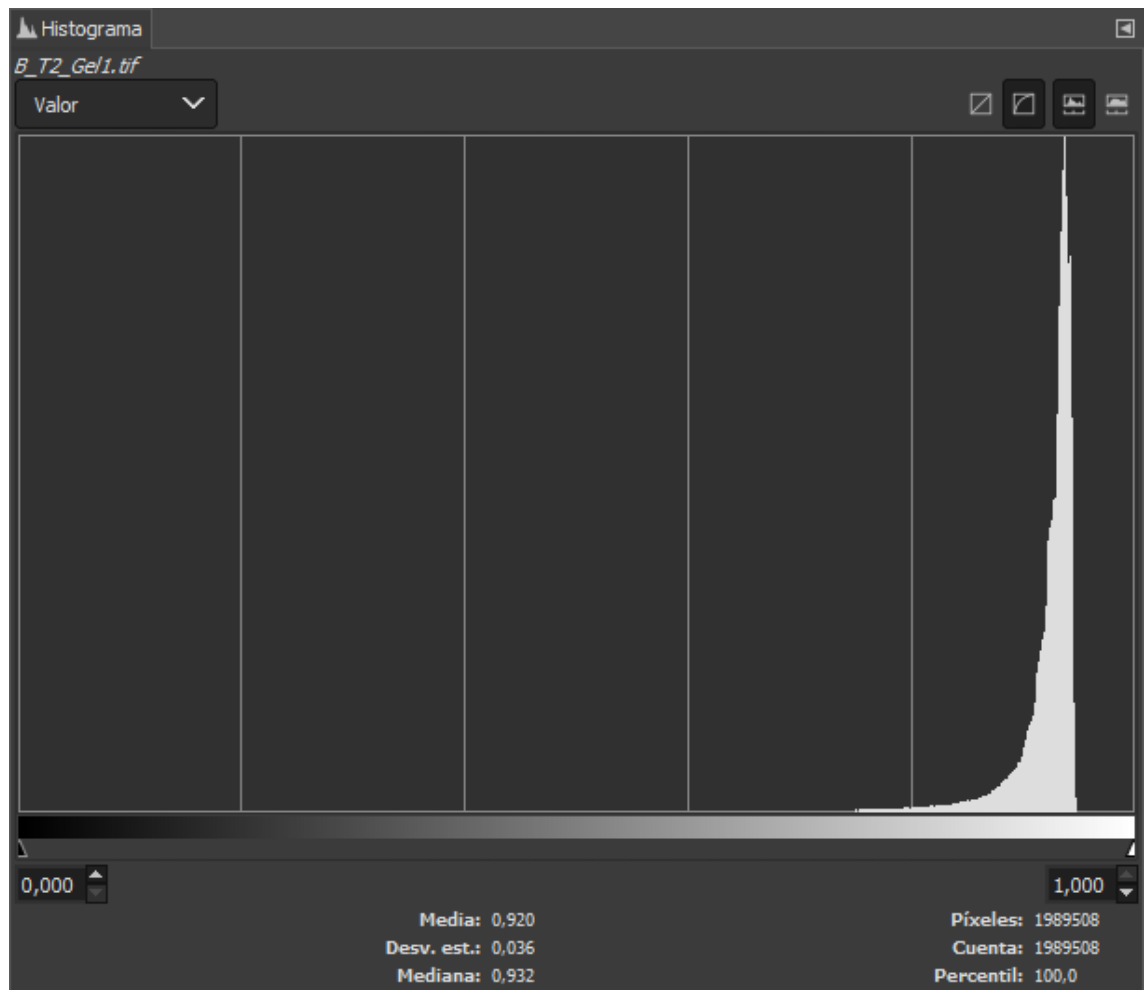


Ilustración 35 - Histograma imagen sin preprocesar

De forma similar a los casos anteriores, podemos observar que el histograma de la imagen preprocesada por el software Melanie posee una media inferior (0.898) al histograma de la imagen sin preprocesamiento (0.920) lo que nos indica que la imagen preprocesada por Melanie es más oscura que la imagen que aportamos de base.

Si nos detenemos en la segmentación como en el caso anterior, podemos ver que el software propio genera muchas menos regiones que el software Melanie, pero para tener una idea más clara vamos a verlo con datos:

- **Melanie:** 1858 regiones generadas
- **Software propio:** 128 regiones generadas

Podemos ver claramente que el software propio genera muchas menos regiones que el software Melanie. Si nos centramos en alguna mancha igual que en el caso anterior, podemos ver lo siguiente:

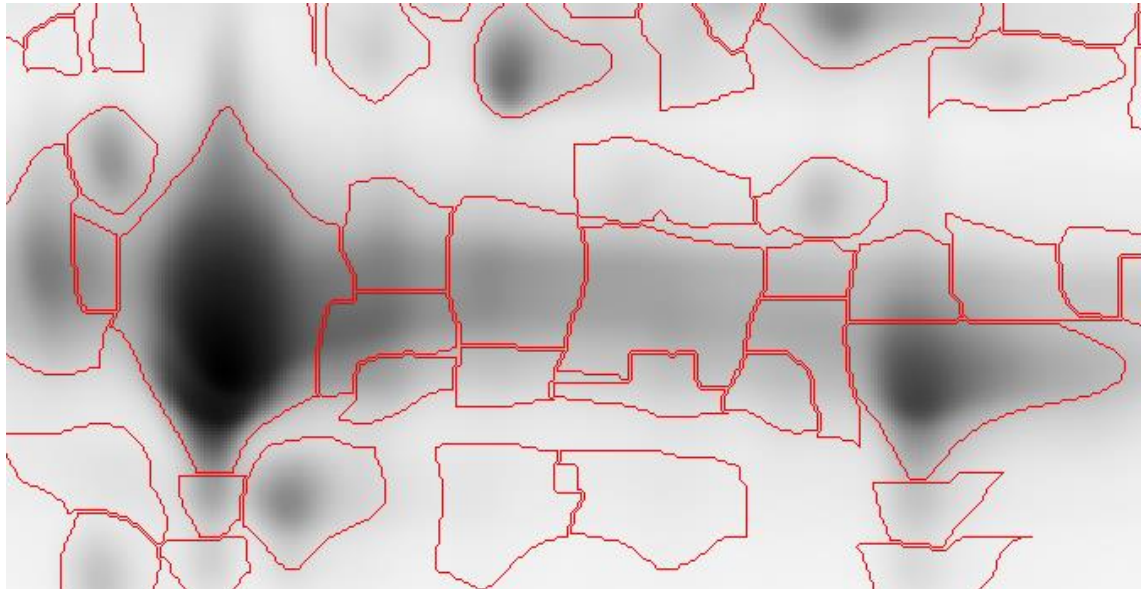


Ilustración 36 - ROI B_T2_Gel1 - Melanie

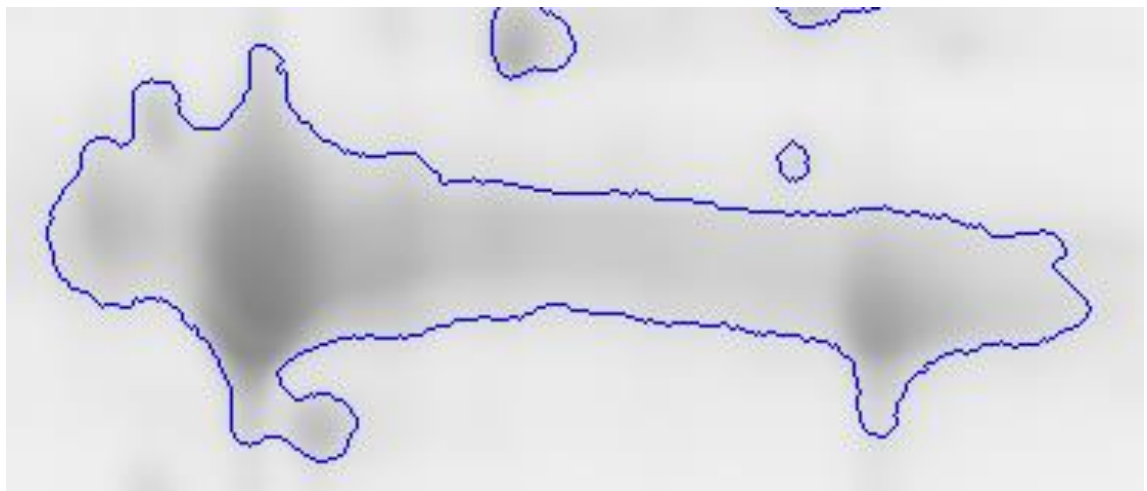


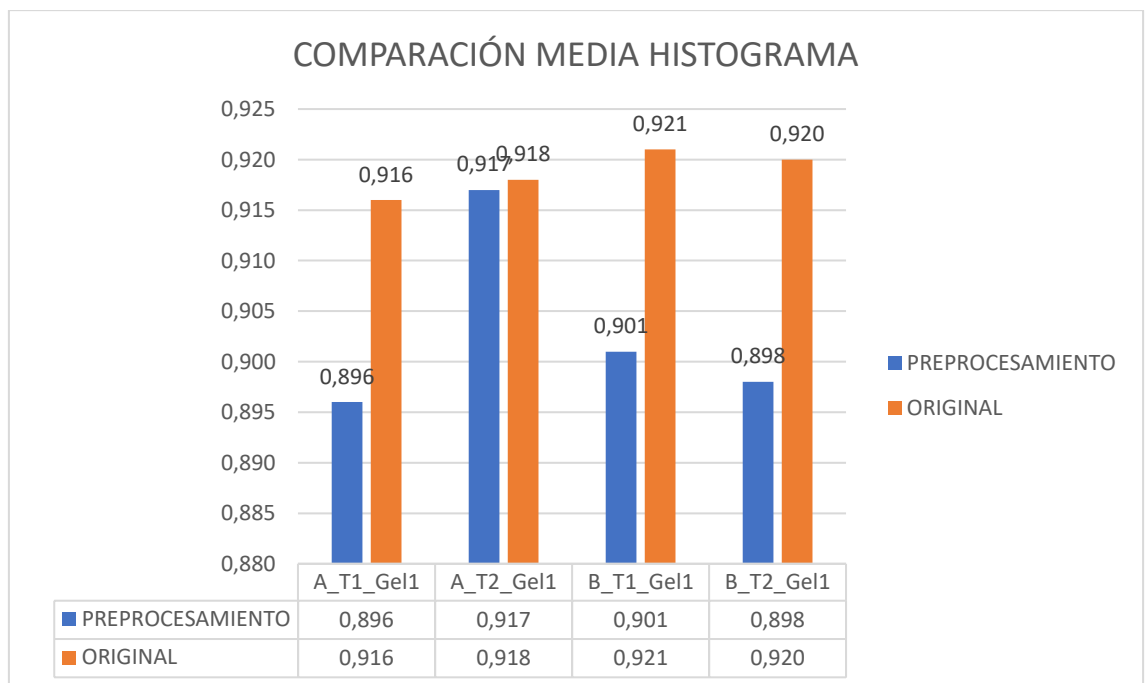
Ilustración 37 - ROI B_T2_Gel1 - Software propio

En este caso, ocurre algo similar al caso anterior y al primero, existen dos zonas claramente más profundas que en el software Melanie aparecen separadas en varias regiones y en nuestro software aparece de forma única.

CONCLUSIONES EXPERIMENTACIÓN

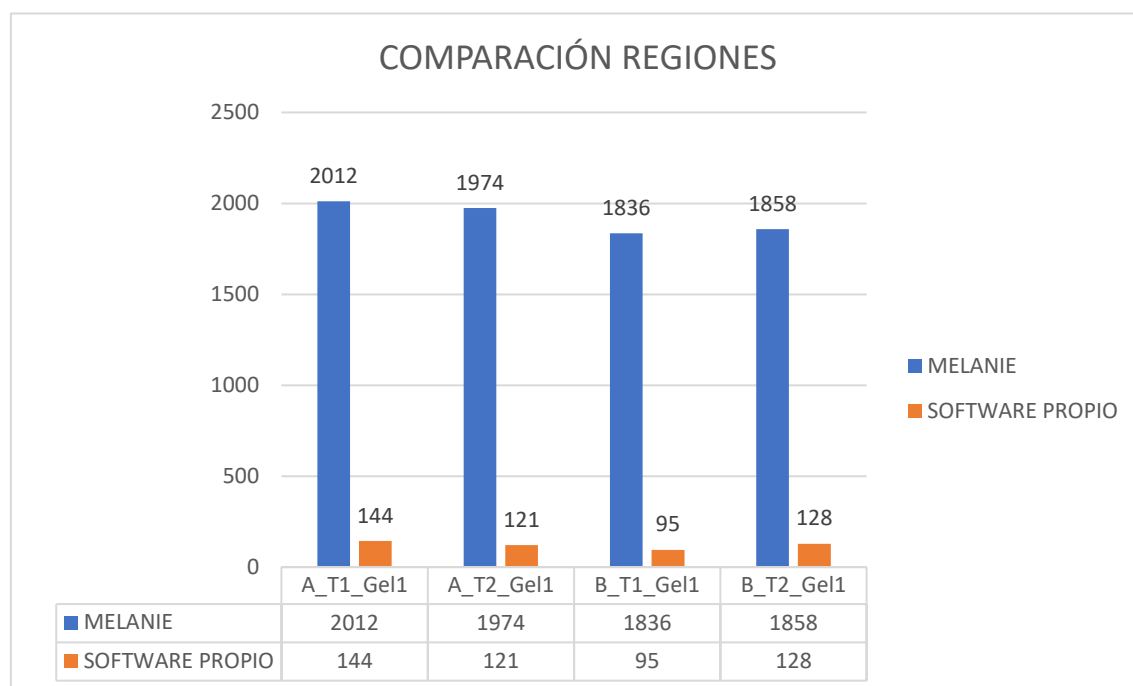
Tras la experimentación realizada, podemos determinar que nuestro software funciona de forma muy diferente al software Melanie.

En primer lugar, el software Melanie parte con algo de ventaja debido a que realiza un preprocesamiento de la imagen suministrada mejor que el que realizamos en nuestro software propio. Hemos determinado en cada experimento que el preprocesamiento ha aumentado el nivel de negros gracias al histograma de la imagen que nos ha proporcionado el programa de diseño gratuito GIMP. Podemos ver un resumen de los datos del histograma en la siguiente tabla:



Pero también es cierto, que incluso sin haber realizado este preprocesamiento mejorado del software Melanie, nuestro software ha sido capaz de determinar las regiones de interés sin ningún tipo de problema.

Por otra parte, se ha podido observar que el software Melanie produce una segmentación mucho mayor que nuestro software propio, tal y como se ha demostrado con el estudio de las regiones por cada imagen, del cual podemos ver un resumen en la siguiente tabla:



Esto puede deberse a varios motivos diferentes:

- **Algoritmo utilizado:** El primero de los motivos por lo que se puede producir la sobresegmentación es debido al algoritmo que se ha utilizado para la segmentación en sí, de forma que un algoritmo más optimizado podría obtener unos resultados mejores.
- **Falta de experiencia de configuración:** Una posible causa de dicha sobresegmentación puede ser el hecho de que no se haya configurado el programa de la forma correcta debido a que no poseo ninguna experiencia.
- **Necesidad de sobresegmentar:** El motivo que puede ser el más probable es que esas segmentaciones extra que realiza el algoritmo pueden llegar a tener un sentido médico que se escapa a mi comprensión. Esto es debido a que este software está enfocado a profesionales sanitarios e investigadores y no busca un procesamiento de imagen óptimo, sino que busca ofrecer la máxima información de la imagen.

7. CONCLUSIONES

El algoritmo WaterShed es un algoritmo muy versátil y eficiente de implementar, el cual ya posee implementaciones suficientemente robustas y estudiadas como la que incorpora la librería de tratamiento de imágenes OpenCV.

En este trabajo hemos podido estudiar la parte teórica del algoritmo, entendiendo su funcionamiento a un nivel más matemático y pudiendo llegar a interpretar un pseudo-código obtenido del artículo principal en el que se basa este trabajo, correspondiente a la implementación del algoritmo.

El software Melanie ha resultado ser un software muy completo y funcional que permite obtener unos resultados de segmentación muy buenos y simples de visualizar, además de ofrecernos la posibilidad de generar un análisis de la segmentación de una zona concreta.

Al comparar los resultados obtenidos por nuestro software propio y el software Melanie, hemos podido establecer diferentes causas que pueden presentarse a la explicación de la sobresegmentación producida en el software Melanie, siendo la más probable el enfoque que tiene el programa. Este programa está pensado para investigadores y personal sanitario más experto que no le interesa el procesamiento de imagen en sí, sino los resultados que pueden obtener con dicho procesamiento.

Esta claro que en las regiones donde se produce la sobresegmentación del software Melanie, hay algo que no somos capaces de apreciar debido a la falta de conocimiento médico.

Dejando de lado dicha diferencia y centrándonos en el rendimiento del software propio, hay que destacar la capacidad que ha tenido de determinar las regiones incluso con un nivel de gris menor al de las imágenes del software Melanie. Es cierto que algunas manchas no las ha llegado a interpretar como regiones, pero puede que realmente no posean valor en el análisis.

Aunque la principal cuestión es que se ha demostrado que una implementación básica y sencilla del algoritmo WaterShed con los parámetros adecuados para el tipo de segmentación que queremos realizar es capaz de ofrecer mucho rendimiento en un entorno real.

En un futuro, podrían incluirse mejoras a nuestro software propio, incluyendo la capacidad de colorear las zonas de mayor depresión correspondiente a las diferentes regiones, extraer información médica de la segmentación que se ha producido y poder realizar una segmentación aún más optimizada y eficaz.

Me gustaría concluir las conclusiones agradeciendo a todos las personas que han ayudado al desarrollo de este trabajo, tanto de forma activa como simplemente dejando libre su trabajo para que sea caso de estudio y se puedan realizar informes como este donde podamos seguir ampliando nuestro conocimiento.

8. 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/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/cabanille-smengual_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-water-shed-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/Segmentacion_\(procesamiento_de_imagenes\)](https://es.wikipedia.org/wiki/Segmentacion_(procesamiento_de_imagenes))

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

ANEXO

A1. CÓDIGO PROGRAMA PROPIO

En esta parte del anexo, se incluye el código que se ha utilizado para realizar la segmentación utilizando el “programa propio”.

Para que el código funcione, es necesario especificar la ruta en la que se encuentra la imagen que se va a analizar. En este caso, se incluye una carpeta Images con algunas imágenes de interés para analizar además de las analizadas en este trabajo.

Este código puede encontrarse además en GitHub, al cual podemos acceder mediante el siguiente enlace: [Placix5/Medical-image-segmentation](https://github.com/Placix5/Medical-image-segmentation)

```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt

#####
#####

def Segmentation(img_path):

    # Cargamos la imagen que vamos a analizar en dos variables, la
    # primera para realizar las diferentes transformaciones y la segunda de
    # ellas
    # para poder mostrarla al final y poder comparar visualmente
    img = cv.imread(img_path)
    img_original = cv.imread(img_path)

    # Convertimos la imagen a escala de grises para poder realizar el
    # tratamiento de una forma sencilla en un único canal
    img_grey = cv.cvtColor(img, cv.COLOR_BGR2GRAY)

    # Aplicamos un filtro Threshold con lo que buscamos poder diferenciar
    # el fondo de la imagen. Este filtro define un umbral de color, todos
    # aquellos píxeles
    # que no cumplan dicho umbral, se les proporciona un valor y aquellos
    # que sí lo cumplen se les proporciona otro
    ret, thresh = cv.threshold(img_grey, 0, 255,
cv.THRESH_BINARY_INV+cv.THRESH_OTSU)
```

```

    # Definimos un kernel para utilizarlo en la operación morfológica
    siguiente
    kernel = np.ones((3,3), np.uint8)
    # Aplicamos una operación de apertura para eliminar el ruido de la
    imagen
    opening = cv.morphologyEx(thresh, cv.MORPH_OPEN, kernel, iterations =
    2)

    # Extraemos la operación de dilatación para extraer el fondo seguro
    de la imagen
    sure_bg = cv.dilate(opening, kernel, iterations = 3)

    # Aplicamos una transformación por distancia a cada centro, de tal
    forma de que cuanto más alejado esté el pixel del centro, mayor será la
    probabilidad
    # de que ese área sea fondo. De esta forma conseguimos extraer lo que
    no es fondo de la imagen
    dist_transform = cv.distanceTransform(opening, cv.DIST_L2, 5)
    # Aplicamos un filtro Threshold para extraer el no fondo
    ret, sure_fg =
    cv.threshold(dist_transform, 0.1*dist_transform.max(), 255, 0)

    # Convertimos el no fondo para poder realizar operaciones con él
    sure_fg = np.uint8(sure_fg)
    # Extraemos las zonas desconocidas eliminándole al fondo la zona que
    sabemos que no es fondo
    unknow = cv.subtract(sure_bg, sure_fg)

    # Generamos los marcadores a partir de las componentes conexas de la
    imagen
    ret, markers = cv.connectedComponents(sure_fg)

    markers = markers + 1
    markers[unknow == 255] = 0

    # Aplicamos el algoritmo WaterShed a la imagen con los marcadores
    como semillas
    markers = cv.watershed(img, markers)

    # Dibujamos el contorno de color azul
    img[markers == -1] = [255, 0, 0]

    # Ejecutamos un umbral Threshold para quedarnos únicamente con las zonas
    de la imagen que sean de color azul
    th = cv.inRange(img, (255, 0, 0), (255, 0, 0)).astype(np.uint8)

    # Se invierte la imagen para separar los bordes
    th = cv.bitwise_not(th)

```



```

# Llamamos a connectedComponentsWithStats para extraer diferente
información de la imagen
nb_comp,output,sizes,centroids=cv.connectedComponentsWithStats(th,con
nectivity=4)

# Eliminamos el fondo
nb_comp-=1; sizes=sizes[1:,-1]; centroids=centroids[1:,:]

# Imprimimos por pantalla lo que realmente nos interesa, que en este
caso es el número de centroides
# que tiene la imagen, al cual tenemos que quitarle uno debido a que
la propia imagen también es segmentada en su totalidad
print(len(centroids) - 1)

# Mostramos la imagen original y la imagen con el contorno dibujado
cv.imshow("Imagen original",img_original)
cv.imshow("Imagen watershed",img)
cv.waitKey(0)

#####
#####

# Para hacerlo modular, se ha encapsulado el código en una función, de
esta forma podemos llamarlo cuantas veces queramos
# definiendo únicamente el path de la imagen que queremos analizar
img_path = "Images/A_T2_Gel1.tif"
Segmentation(img_path)

```

A2. METODOLOGÍA DE ELABORACIÓN DEL TD Y CARGA DE TRABAJO

En esta sección se expondrá la tabla de carga de trabajo, la cual recoge diferentes partes de elaboración del trabajo y la carga que ha supuesto cada una de dichas partes:

Fecha	Hora inicio	Hora fin	Interrupciones	Tiempo total	Actividad	Comentarios
27/12/21	16:30	21:00	1 Hora	3.5 Horas	Búsqueda de tema para la realización	
1/02/22	16:30	18:30		2 Horas	Planteamiento	
7/02/22	8:00	10:00		2 Horas	Desarrollo de la introducción	
10/02/22	18:00	20:00		2 Horas	Re-lectura de los papers	
11/02/22	16:30	20:30	1 Hora	3 Horas	Desarrollo marco teórico	
12/02/22	08:30	14:00	1.5 Hora	4 Horas	Desarrollo de implementación del WaterShed	
12/02/22	16:00	21:00	1 Hora	4 Horas	Desarrollo de implementación del WaterShed	
13/02/22	08:00	12:00	1 Hora	3 Horas	Búsqueda de recursos para la experimentación	
15/02/22	16:30	21:00	1 Hora	3.5 Horas	Búsqueda de recursos para la experimentación	
16/02/22	16:30	20:00	1 Hora	2.5 Horas	Búsqueda de recursos para la experimentación	
17/02/22	16:30	20:00	1 Hora	2.5 Horas	Desarrollo de código para experimentación	
18/02/22	16:30	20:00		3.5 Horas	Desarrollo del apartado experimental	
19/02/22	08:00	12:00		4 Horas	Revisión Melanie	
20/02/22	08:00	14:00	1 Hora	5 Horas	Desarrollo de nuevo código	
21/02/22	16:30	20:00	1 Hora	2.5 Horas	Finalización del documento	
25/02/22	16:30	23:00	1 Hora	5.5 Horas	Actualización del programa, actualización de la parte de experimentación y actualización de las conclusiones	