

# Aplicación de técnicas de procesamiento avanzado de imagen para la detección y discriminación de muestras histológicas humanas

Trabajo Fin de Grado

Juan Nájjar Alemany

*Tutor:*

Valery Naranjo Ornedo

*Co-tutor:*

Francisco José Peñaranda Gómez

13 de Septiembre de 2016

Computer Vision and Behaviour Analysis Lab - Instituto de Investigación e Innovación en  
Bioingeniería (i3B)



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



— TELECOM ESCUELA  
TÉCNICA VLC SUPERIOR  
DE UPV INGENIEROS  
DE TELECOMUNICACIÓN



# Agradecimientos

A Fran, por su infinita paciencia y ayuda.

A Adrián, por facilitarme aquel comienzo.

A mis amigos, por ser quienes son. Sabéis que esto sólo acaba de empezar.

A mi madre y mi hermano, por su apoyo incondicional.

A todos aquellos que me ayudan sólo con estar ahí.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Introducción . . . . .	2
1.2. Planteamiento del proyecto . . . . .	2
1.3. Objetivos del trabajo . . . . .	3
1.4. Guía del trabajo . . . . .	3
<b>2. Estado del arte</b>	<b>4</b>
2.1. Muestras histológicas . . . . .	5
2.2. Marcadores celulares histoquímicos. Ki-67 . . . . .	6
2.3. Patología digital. Escáneres WSI . . . . .	7
2.3.1. ¿Qué es un escáner WSI? . . . . .	7
2.3.2. ¿Cómo se realiza el escaneo? . . . . .	8
2.4. Conteo de células. Métodos más utilizados . . . . .	9
2.5. Conclusiones . . . . .	10
<b>3. Gestión y distribución de tareas</b>	<b>11</b>
3.1. Distribución de tareas y metodología . . . . .	12
3.2. Gestión del tiempo dedicado al trabajo . . . . .	13
<b>4. Desarrollo y métodos utilizados</b>	<b>16</b>
4.1. Introducción . . . . .	17
4.2. Librería <i>OpenSlide</i> . . . . .	17
4.2.1. <i>OpenSlide</i> en MATLAB . . . . .	18
4.3. Interfaz (GUI) en MATLAB . . . . .	20
4.3.1. Justificación . . . . .	20
4.3.2. Objetivo . . . . .	20
4.3.3. Funcionamiento . . . . .	20
4.3.4. Interfaces de carga y visionado . . . . .	22

---

4.4. Procesamiento por bloques. Problemática . . . . .	28
4.4.1. La función <i>blockproc</i> . . . . .	28
4.4.2. Visionado profundo de una imagen de máxima resolución . . . . .	29
4.4.3. Análisis de imagen por bloques . . . . .	29
4.5. Análisis de imagen . . . . .	30
4.5.1. Introducción . . . . .	30
4.5.2. Teoría de los métodos utilizados . . . . .	31
4.5.3. Procesado de la imagen y segmentación de células . . . . .	38
4.5.4. Conteo manual y máscara de comprobación . . . . .	46
<b>5. Resultados</b>	<b>48</b>
5.1. Introducción . . . . .	49
5.2. Métodos de separación de componentes de color . . . . .	49
5.2.1. PCA . . . . .	50
5.2.2. VCA . . . . .	51
5.2.3. ICA . . . . .	53
5.2.4. Deconvolución de color manual . . . . .	55
5.3. Procesado morfológico . . . . .	57
5.4. Segmentación . . . . .	60
5.4.1. Deconvolución manual . . . . .	60
5.4.2. Método VCA . . . . .	63
5.4.3. Contabilización . . . . .	65
<b>6. Conclusiones y trabajo futuro</b>	<b>70</b>
6.1. Introducción . . . . .	71
6.2. Métodos de separación de componentes de color . . . . .	71
6.3. Procesado morfológico . . . . .	72
6.4. Segmentación . . . . .	72
6.4.1. Contabilización . . . . .	72
6.4.2. Discusión . . . . .	73
6.5. Trabajo futuro . . . . .	75
<b>Bibliografía</b>	<b>76</b>

# Índice de figuras

2.1. Inserción de una muestra en parafina para crear el bloque que posteriormente será cortado en rebanadas. Imagen tomada de [5]. . . . .	5
2.2. Muestras tratadas con Ki-67 (izquierda) y Hematoxilina-Eosina o H&E (derecha) tomadas de muestras disponibles para el trabajo. . . . .	6
2.3. Escáner WSI del fabricante Ventana. Fotografía tomada de [13]. . . . .	8
2.4. A la izquierda, división de la muestra por el escaneo en mosaico y a la derecha por el método en línea. Imagen adaptada de [11]. . . . .	9
3.1. Diagrama temporal de la realización del trabajo. . . . .	15
4.1. Algunos de los campos que muestra la librería <i>OpenSlide</i> al abrir un archivo BIF. . . . .	17
4.2. Imagen asociada “macro” de una de las muestras disponibles. . . . .	18
4.3. Flujograma del funcionamiento de la interfaz de carga creado con Draw.io [19]. . . . .	23
4.4. Selección de una región de la imagen. . . . .	24
4.5. Desplegable con las distintas opciones de análisis. . . . .	25
4.6. Flujograma del funcionamiento de la interfaz de trabajo. . . . .	26
4.7. Aspecto de la interfaz de carga. . . . .	27
4.8. Aspecto de la interfaz de trabajo. . . . .	27
4.9. Representación del procesado por bloques de una imagen tomada de [20]. . . . .	28

4.10. Ejemplo de reducción de dimensionalidad. De izquierda a derecha: Datos originales y autovectores. 3D > Proyección sobre los dos autovectores de mayor variabilidad. 2D > Proyección sobre el mayor autovector. 1D. Ejemplo adaptado de [26]. . . . .	32
4.11. A la izquierda, se observa la representación de los puntos del conjunto de datos y la estimación de los <i>endmembers</i> en rojo. A la derecha, se observan los puntos proyectados del conjunto de datos y los tres <i>endmembers</i> encontrados. Imágenes obtenidas de la demostración 1 del código [31]. . . . .	33
4.12. Aplicación de VCA en imagen con tres colores. En la parte inferior de la imagen se observan los <i>endmembers</i> encontrados por el algoritmo. En la parte superior se encuentra la separación de la imagen original en tres imágenes donde, en cada una, se encuentran en blanco todos los píxeles que pertenecen a cada <i>endmember</i> . . . . .	34
4.13. Ejemplo de resultado de la deconvolución de color en HDAB tomada de [35]. De izquierda a derecha, imagen original > deconvolución componente azul > deconvolución componente marrón . . . . .	36
4.14. Representación de los mínimos y máximos en la aplicación del método <i>Watershed</i> tomada de [41]. . . . .	40
4.15. A la izquierda, resultado de aplicar la FRST y a la derecha la máscara de marcadores internos tras aplicar <i>imextendedmax(A,1)</i> . . . . .	42
4.16. A la izquierda, imagen original. A la derecha, aplicación de <i>bwdist</i> sobre la imagen complementaria de la original. Ejemplo tomado de la documentación de la función en MATLAB. . . . .	43
4.17. Flujograma de funcionamiento del análisis de imagen MATLAB. . . . .	45

4.18. De izquierda a derecha: Muestra original > Segmentación manual > Máscara de segmentación de células azules > Máscara de segmentación de células marrones. . . . .	47
5.1. Resultados obtenidos con el método PCA, apareciendo de izquierda a derecha la imagen original, la componente para las células marrones, la componente para las células azules y una tercera componente del fondo. . . . .	51
5.2. Resultados obtenidos con el método VCA. De izquierda a derecha, imagen original > componentes marrones > componente de azules > componente inutilizable. . . . .	52
5.3. Comparación de la misma muestra en distintas ejecuciones. . . . .	53
5.4. Resultados obtenidos con el método ICA. De izquierda a derecha, imagen original > células marrones > células azules > componente inservible. . . . .	54
5.5. Resultados obtenidos con el método de deconvolución manual. De izquierda a derecha, imagen original > deconvolución de células marrones > deconvolución de células azules > residuo. . . . .	56
5.6. A la izquierda, artefactos producidos en la deconvolución de color manual. A la derecha, artefactos producidos en una de las componentes del método PCA. . . . .	56
5.7. Resultados procesados del método de deconvolución manual. De izquierda a derecha, imagen de células marrones > imagen procesada de células marrones > imagen de células azules > imagen procesada de células azules. . . . .	58
5.8. Resultados procesados de VCA. De izquierda a derecha, imagen complementaria de células marrones > imagen procesada de células marrones > imagen complementaria de células azules > imagen procesada de células azules. . . . .	59

5.9. Resultados segmentación con el método de deconvolución manual de la muestra 1. De izquierda a derecha: original > marcadores > <i>watershed</i> > watershed sobre original. . . . .	60
5.10. Resultados segmentación con el método de deconvolución manual de la muestra 2. De izquierda a derecha: original > marcadores > watershed > watershed sobre original. . . . .	61
5.11. Resultados segmentación con el método de deconvolución manual de la muestra 3. De izquierda a derecha: original > marcadores > watershed > watershed sobre original. . . . .	61
5.12. Resultados segmentación con el método de deconvolución manual de la muestra 4. De izquierda a derecha: original > marcadores > watershed > watershed sobre original. . . . .	62
5.13. Resultados segmentación con el método de deconvolución manual de la muestra 5. De izquierda a derecha: original > marcadores > watershed > watershed sobre original. . . . .	62
5.14. Resultados segmentación con el método VCA de la muestra 1. De izquierda a derecha: original > marcadores > watershed > watershed sobre original. . . . .	63
5.15. Resultados segmentación con el método VCA de la muestra 2. De izquierda a derecha: original > marcadores > watershed > watershed sobre original. . . . .	64
5.16. Resultados segmentación con el método VCA de la muestra 3. De izquierda a derecha: original > marcadores > watershed > watershed sobre original. . . . .	64
5.17. Resultados segmentación con el método VCA de la muestra 4. De izquierda a derecha: original > marcadores > watershed > watershed sobre original. . . . .	65
5.18. Resultados segmentación con el método VCA de la muestra 5. De izquierda a derecha: original > marcadores > watershed > watershed sobre original. . . . .	65

---

5.19. Separación de células marrones en deconvolución manual y VCA. . . . .	69
--	----

# Índice de tablas

5.1. Número de células detectadas con la segmentación manual. . . . .	65
5.2. Número de células detectadas con la segmentación de la deconvolución de color manual. . . . .	66
5.3. Número de células detectadas con la segmentación del método VCA. . . . .	66
5.4. Índices obtenidos para la deconvolución de color manual. . . . .	68
5.5. Índices obtenidos para el método VCA. . . . .	68

# Resumen

Este trabajo fin de grado (TFG) se presenta como una primera aproximación al análisis de imágenes de microscopía de muestras histológicas humanas tratadas con el marcador celular Ki-67.

La justificación del análisis de estas imágenes radica en la importancia que tiene encontrar un método automático, preciso y fiable que, independientemente del patólogo, sea capaz de aportar información valiosa en el diagnóstico de distintas enfermedades. De esta forma, un buen análisis ayuda a realizar un pronóstico más acertado que puede permitir un seguimiento y tratamiento mejores en el paciente.

Para realizar el estudio se ha optado por la elección de las principales técnicas de tratamiento de imagen utilizadas en el campo de la histología realizando una comparativa entre ellas. Dicha comparativa sirve para exponer sus inconvenientes y ventajas con la finalidad de definir un camino para profundizar en el estudio de las muestras tñtadas con Ki-67. Todas estas técnicas han sido incorporadas en una interfaz gráfica propia, creada con el software MATLAB, capaz de trabajar con los formatos de archivos específicos de este tipo de imágenes gracias a la librería *Openslide*. Con esta interfaz se pretenden sentar las bases para la creación de una aplicación completamente funcional y utilizable en el campo médico.

## Resum

Aquest treball de final de grau (TFG) es presenta com a una primera aproximació a l'anàlisi d'imatges de microscòpia de mostres histològiques humanes tractades amb el tint Ki-67.

La justificació a l'anàlisi d'aquestes imatges radica en la importància que té trobar un mètode automàtic, precís i fiable que, independentment del patòleg, siga capaç d'aportar informació valuosa en el diagnòstic de distintes malalties. D'aquesta manera,

un bon anàlisi ajuda a realitzar un pronòstic més acurat que pot permetre un seguiment i tractament millor en el pacient.

Per a realitzar l'estudi s'ha optat per l'elecció de les principals tècniques de tractament d'imatge utilitzades en el camp de la histologia realitzant una comparativa entre elles. Dita comparativa serveix per exposar els seus inconvenients i avantatges amb la finalitat de definir un camí per a profunditzar en l'estudi de les mostres tintades amb Ki-67. Aquestes tècniques han sigut incorporades en una interfície gràfica pròpia que ha sigut creada amb el software MATLAB, el qual és capaç de treballar amb els formats d'arxius propis d'aquest tipus d'imatges gràcies a la llibreria *OpenSlide*. Amb aquesta interfície es pretén assentar les bases per a la creació d'una aplicació completament funcional i utilitzable en el camp mèdic.

## Abstract

This project's essay aims a first approximation to microscopy image analysis of histological human samples treated with Ki-67 dye.

The justification of this analysis has its point at the importance of finding an automatic, accurate and reliable method to get valuable information about illness diagnostic without pathologist intervention. Thereby, good analysis helps doctors to improve their monitoring and treatment in their patients.

To realize the study, main techniques in histology image treatment has been chosen. Advantages and disadvantages have been shown by comparing all choose techniques. The aim of that comparison is to define a proper way to deepen in the study of Ki-67 samples. These techniques have been included into an own graphical user interface. This interface has been created with MATLAB software, that allows to work with histology file formats through *OpenSlide* library. This interface aims to lay the groundwork to create a completely functional application usable in medic field.

---



# Capítulo 1

## Introducción

### Índice de contenidos

---

1.1. Introducción . . . . .	2
1.2. Planteamiento del proyecto . . . . .	2
1.3. Objetivos del trabajo . . . . .	3
1.4. Guía del trabajo . . . . .	3

---

## 1.1. Introducción

Este Trabajo Fin de Grado (TFG) surge de un proyecto del Computer Vision and Behaviour Analysis Lab, perteneciente al Instituto de Investigación e Innovación en Bioingeniería (*I3B*) de la Universidad Politécnica de Valencia, para el Hospital Clínico de Valencia.

## 1.2. Planteamiento del proyecto

La razón de ser de este TFG es la realización de un primer acercamiento a los métodos más usados en el tratamiento de las imágenes histológicas tratadas con el marcador celular denominado Ki-67, con la finalidad de sentar las bases de desarrollo de una aplicación capaz de ser usada por profesionales del mundo de la medicina. Normalmente, en el tipo de imágenes que competen a este trabajo, se pueden apreciar dos tipos de células: potencialmente cancerígenas en marrón y benignas en azul.

La necesidad de trabajar con estas imágenes radica en la importancia de la información que pueden llegar a proporcionar, ayudando con ello a mejorar el diagnóstico de enfermedades. Dado que el método de análisis manual es muy tedioso y consume demasiado tiempo, cada vez es más frecuente encontrar estudios sobre los métodos más fiables para realizar este proceso de manera automática [1] [2].

Todos los métodos comparten una serie de pasos desde la imagen en color original hasta el conteo final de células, a saber: un preprocesado de imagen que separe los distintos tipos de células lo mejor posible, un método de segmentación de células y un post-procesado para descartar falsos positivos.

Para la realización del presente TFG, se ha implementado una interfaz gráfica en MATLAB que es capaz de trabajar con las imágenes conocidas como *Whole Slide Image*, las cuales tienen un gran tamaño y requieren de librerías especializadas para su manejo. Haciendo uso de la interfaz, se recopilarán los métodos de análisis más empleados, se pondrán a prueba y se analizarán sus problemáticas, se compararán entre sí y se propondrán soluciones y vías de trabajo para el futuro.

Así pues, el propósito del análisis es realizar una discriminación de las células cancerígenas respecto de las benignas de forma eficaz y con la mínima interacción del usuario, obteniendo información de su número, proporción, etc.

---

### 1.3. Objetivos del trabajo

Los objetivos que se plantean en este trabajo son los siguientes:

1. Desarrollar una interfaz gráfica con la herramienta MATLAB que sea capaz de trabajar con imágenes de microscopía.
2. Analizar los distintos métodos de estudio de imágenes de muestras histológicas tratadas con el marcador celular Ki-67, obteniendo así un listado de problemas y fallos.
3. Comparar los métodos empleados para determinar cuál podría ser la mejor vía para seguir profundizando en el análisis dentro de los límites establecidos por los problemas encontrados.
4. Proponer nuevas líneas de trabajo y mejoras futuras tanto para la interfaz como para el análisis.

### 1.4. Guía del trabajo

En el Capítulo 2 se desarrolla el estado del arte actual, describiendo tanto el origen de las muestras histológicas utilizadas en el este TFG como los métodos para su obtención, algunas de sus características y el porqué de la necesidad de analizarlas.

En el Capítulo 3 se detalla la organización llevada a cabo para la elaboración de este TFG. Se descompone el proyecto en tareas más pequeñas e independientes y se establecen los tiempos dedicados a cada apartado.

En el Capítulo 4 se exponen los conceptos teóricos empleados para la realización de las distintas tareas así como el desarrollo y análisis detallado de las mismas.

En el Capítulo 5 se recopilan los resultados obtenidos por cada parte de la herramienta y se comparan con muestras segmentadas manualmente.

Por último, en el Capítulo 6 se describen las conclusiones a las que se ha llegado tras la elaboración del trabajo y se recopilan todos los inconvenientes encontrados durante el mismo, proponiendo nuevas opciones o mejoras.

---

# Capítulo 2

## Estado del arte

### Índice de contenidos

---

<b>2.1. Muestras histológicas</b> . . . . .	<b>5</b>
<b>2.2. Marcadores celulares histoquími- cos. Ki-67</b> . . . . .	<b>6</b>
<b>2.3. Patología digital. Escáneres WSI</b> . . . . .	<b>7</b>
2.3.1. ¿Qué es un escáner WSI? . . . . .	7
2.3.2. ¿Cómo se realiza el escaneo? . . . . .	8
<b>2.4. Conteo de células. Métodos más utilizados</b> . . . . .	<b>9</b>
<b>2.5. Conclusiones</b> . . . . .	<b>10</b>

---

## 2.1. Muestras histológicas

Se conoce como histología al estudio a través de un microscopio de las células, tejidos y órganos que componen a los seres vivos, por lo que su nacimiento se remonta al año 1600, cuando se incorporó el microscopio a los estudios anatómicos.

Para poder realizar dicho estudio, actualmente se sigue un proceso que se puede dividir en las siguientes partes[3] [4]:

- **Obtención de la muestra de biopsia:** El médico extrae del paciente un fragmento de tejido sospechoso que necesita de un análisis en profundidad. Este se conserva en formol y se envía al laboratorio.
- **Creación de las rebanadas:** Las muestras se deshidratan con alcoholes y posteriormente se introducen en algún tipo de resina o polímero, como la parafina, quedando conservadas en bloques para su almacenamiento como se observa en la Figura 2.1. Posteriormente, se introduce en un micrótopo que permite realizar cortes muy finos de la muestra, conocidos como *slides* o rebanadas.
- **Tintado de los *slides*:** Cada corte es tratado con un tinte especial en función del tejido y del análisis que se espera realizar. Estos tintes, como el Ki-67, hacen mucho más distinguibles los distintos tipos de células.
- **Digitalización:** La muestra es revisada por el patólogo a través de un microscopio para casos sencillos o digitalizada para casos complejos. El proceso de digitalización se realiza con unos escáneres especiales conocidos como escáneres *Whole Slide Image* o WSI.



Figura 2.1: Inserción de una muestra en parafina para crear el bloque que posteriormente será cortado en rebanadas. Imagen tomada de [5].

---

## 2.2. Marcadores celulares histoquímicos. Ki-67

El proceso de marcado consiste en aplicar un tinte especial a la muestra dependiendo del objeto de estudio de la misma. Estos tintes consisten en una mezcla de un antígeno acompañado de alguna enzima especial que producen una respuesta inmunológica conjunta sobre la muestra tratada generando una coloración característica y distinta según las células y los componentes usados, como se observa en la Figura 2.2.

Existe una gran cantidad de estos tintes, catalogados como enzimas, antígenos específicos para algunos tipos de tumores, genes supresores de tumores o marcadores de proliferación de los tumores. Con la ayuda de este tratamiento, los doctores pueden diagnosticar el estado de un cáncer y decidir si éste es maligno o benigno. Por ejemplo, en el caso del Ki-67 las células sospechosas de ser cancerígenas se colorean con un color marrón oscuro, por lo que si en una muestra existe una desproporcionada cantidad de células marrones puede ser indicativo de un cáncer avanzado [6].

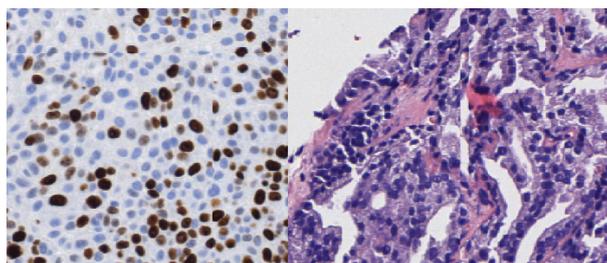


Figura 2.2: Muestras tratadas con Ki-67 (izquierda) y Hematoxilina-Eosina o H&E (derecha) tomadas de muestras disponibles para el trabajo.

El marcador celular Ki-67 es uno de los más utilizados a la hora de tomar decisiones para el tratamiento de algunos tipos de cáncer, como el de vejiga o próstata. La ventaja de este marcador, compuesto por una proteína, es que la misma se encuentra presente durante todas las fases activas de una célula (G(1), S, G(2), y mitosis [7]), especialmente en la fase de mitosis o división celular.

Ésta característica es la que permite hacer de este marcador un factor determinante en la obtención del estado de crecimiento de las células en una población celular, ya que una de las propiedades de una célula cancerígena es su disposición a proliferar continuamente y sin control.

En resumen, podría decirse que el Ki-67 es una buena forma de seguir de cerca el desarrollo de un tumor a través de los ciclos de división celular [6].

---

## 2.3. Patología digital. Escáneres WSI

Se conoce como patología digital a la gestión de la información patológica de las muestras histológicas de forma digital. Además, se incluye dentro de este campo tanto el análisis como el manejo, la digitalización y la interpretación de esa información, todo ello siempre de forma computarizada [8] [9].

Es por ese motivo que esta forma de trabajo es cada vez más indispensable, dado que supone un aumento considerable de la precisión y rapidez en el diagnóstico médico de múltiples enfermedades. Es esa importancia la que explica por qué hay cada vez mayor número de artículos científicos y líneas de investigación volcadas en encontrar métodos y algoritmos más sofisticados para trabajar con las imágenes de histología de manera precisa [10].

No sólo es importante el análisis de las imágenes, sino también su adquisición, y ésta ha pasado en cuestión de años del uso de cámaras digitales al uso de la tecnología más avanzada con los escáneres *Whole Slide Image*.

### 2.3.1. ¿Qué es un escáner WSI?

También conocido como microscopio virtual, el escáner es en realidad un conjunto de elementos [11] [12]. Las partes que componen el sistema completo son:

- Conjunto de lentes especiales para microscopía.
- Una fuente de luz (*Brightfield* o fluorescente).
- Mecanismos controlados digitalmente para manejar los *slides*.
- Un dispositivo de captación de imágenes de alta resolución.
- Un ordenador.
- Un software especializado para manipular y visualizar las imágenes. Normalmente cada vendedor tiene uno propio.

En resumen, el sistema digitaliza los *slides* uno a uno en una o múltiples resoluciones y después los une generando una única imagen de muy alta resolución. El primer microscopio dedicado a estas tareas fue creado por James Bacus en 1994 y actualmente existen múltiples vendedores, como Aperio o Ventana.

---



Figura 2.3: Escáner WSI del fabricante Ventana. Fotografía tomada de [13].

### 2.3.2. ¿Cómo se realiza el escaneo?

La adquisición de las imágenes del microscopio precisa de métodos sofisticados y muy precisos ya que se necesita trabajar a grandes resoluciones para preservar gran cantidad de detalles y poder visualizar toda la muestra de forma general o precisa.

Existen dos métodos comúnmente empleados para dicha digitalización: escaneo en mosaico o *tiling* y escaneo en línea [11] [12].

- **Escaneo en mosaico:** Esta forma de escaneo se realiza moviendo la muestra de forma controlada por ordenador a lo largo del dispositivo captador. Éste va tomando un gran número de fotografías de forma rectangular que posteriormente ensambla como si fueran piezas de un mosaico. A pesar de que se trabaja con muy altas resoluciones e imágenes donde hay un gran número de detalles, el solapamiento producido entre cada elemento de la composición es pequeño gracias a la precisión del movimiento de la muestra.
- **Escaneo en línea.** Con este método se simplifica el movimiento de la muestra dado que ésta es desplazada de forma lineal y en un único eje espacial de adquisición. Tras múltiples pases secuenciales en diferentes zonas del *slide*, se produce un grupo de imágenes en tiras o líneas que son reordenadas de forma sencilla.

En la Figura 2.4 se ilustran los dos métodos de escaneo sobre una muestra.

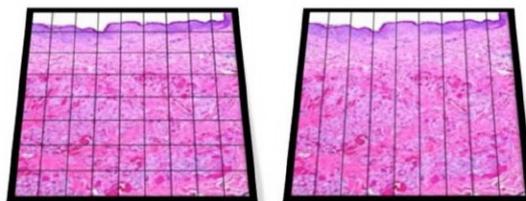


Figura 2.4: A la izquierda, división de la muestra por el escaneo en mosaico y a la derecha por el método en línea. Imagen adaptada de [11].

Una vez se ha realizado el escaneo, el software se encarga de almacenarlo en disco y la capacidad ocupada depende en gran medida de la resolución del sistema de escaneo. La resolución del escáner se determina por la lente utilizada en el microscopio para aumentar la imagen (por ejemplo,  $20\times$  o  $40\times$ ), la apertura numérica del objetivo y la calidad del fotodetector de la cámara. El uso de mayores resoluciones no sólo implica una mayor conservación de información sino que supone un enorme aumento del tamaño del archivo a crear. Las resoluciones típicas son de 0.25 pixel/micra (ppm) para un aumento de  $40\times$  y de 0.50 ppm para uno de  $20\times$ , lo cual implica que las fotografías sin comprimir puedan llegar a ocupar varios gigabytes.

El almacenaje se realiza usando algoritmos de compresión sin pérdidas, como JPEG2000 o TIFF [14], que generan archivos más manejables. Normalmente se genera un archivo que contiene no sólo imágenes, sino mucha información adicional. En concreto, la gran mayoría de dispositivos generan una pirámide de imágenes donde se encuentra la imagen en alta resolución (dependiendo del grado de aumento del escáner), así como las versiones diezmadas de dicha imagen. Además, se incorpora información del escáner, información de las lentes y la densidad de píxeles, entre otras cosas.

El software especializado se encarga de poder visualizar o analizar las enormes imágenes, dado que éstas son imposibles de visualizar por métodos convencionales. Un ejemplo de dicho software es *OpenSlide* [15], que será analizado en profundidad en la Sección 4.2. Una vez digitalizada la muestra comienza la extracción de la información.

## 2.4. Conteo de células. Métodos más utilizados

La finalidad del análisis de las muestras histológicas es poder realizar un diagnóstico médico preciso frente a una enfermedad, pudiendo así predecir su evolución o incluso el tratamiento que deberá seguir el paciente. En el caso del cáncer, que es la enfermedad de la que se pretende obtener información en este TFG, es interesante para el patólogo

---

poder conocer la relación entre células cancerígenas y células benignas en zonas concretas de la muestra. Esto implica conocer el número de ambos tipos de células, su localización y la proporción de unas respecto de las otras.

En el caso del Ki-67, interesa obtener el índice de proliferación Ki-67, ya que indica la población de células malignas. Para su medición, existen diversos métodos [16] :

- **Estimación “eye-ball”**. Consiste en obtener el porcentaje de células positivas bajo el Ki-67 sin contar realmente las células, haciéndolo de forma aproximada. Esto se realiza escaneando la muestra completa a una potencia de aumento pequeña.
- **Conteo “a ojo” a través de un microscopio**. Consiste en el conteo de células en tiempo real por parte del médico de una zona del *slide* que previamente ha sido seleccionada como punto caliente de células positivas en Ki-67.
- **Conteo manual de una imagen digital**. En lugar de en tiempo real, la zona de interés se imprime sobre papel y se realiza el recuento a mano de las células.
- **Conteo automático a través de un programa analizador**. Consiste en el uso de software especializado para aplicar un tratamiento de imagen que permita distinguir en una región seleccionada de una WSI los distintos tipos de elementos e interprete los resultados proporcionando la información deseada al patólogo.

## 2.5. Conclusiones

Obtener un análisis preciso con los métodos manuales es posible, pero puede ser un proceso realmente largo, tedioso y con grandes probabilidades de fallo, dado que depende en gran medida del estado de la persona que realice el estudio en ese momento además de su criterio personal.

Es por esa razón que la búsqueda de métodos precisos que realicen el análisis de forma automática es cada vez más importante y esto requiere de la toma de decisiones respecto a cómo se va a querer tratar la imagen. A lo largo del presente trabajo, se describirán distintos caminos para realizar el conteo de células y se compararán entre sí para concluir cual podría ser más fiable en el caso del Ki-67.

---

## Capítulo 3

# Gestión y distribución de tareas

### Índice de contenidos

---

3.1. Distribución de tareas y metodología . . . . .	12
3.2. Gestión del tiempo dedicado al trabajo . . . . .	13

---

### 3.1. Distribución de tareas y metodología

Dado que la naturaleza de este trabajo es una mezcla de programación en MATLAB y estudio de métodos de análisis de imagen, se ha subdividido en tareas realizables de forma independiente. Concretamente se ha dividido en cinco partes progresivas en el tiempo desde la asignación del proyecto hasta su entrega. La metodología empleada, así como las partes, es la siguiente:

- **Fase de documentación.** Aunque ésta ha estado activa durante toda la realización, ya sea buscando nuevos métodos de análisis de imagen o información de última hora para poder obtener un trabajo más completo, se ha concentrado sobre todo en los primeros meses del trabajo. Dado que las tareas tienen una gran componente teórica, médica o técnica, casi toda la información ha sido extraída de fuentes reconocidas como el *Institute of Electrical and Electronics Engineers* (IEEE) y otras fuentes de publicaciones científicas. La dificultad de esta fase ha sido el desconocimiento casi total de la componente médica del proyecto, así como de la comprensión de nuevos métodos de tratamiento de imagen. Cabe decir que han sido de gran ayuda los conocimientos adquiridos en la asignatura Tratamiento Digital de la Imagen cursada durante la carrera para poder realizar con éxito el trabajo.
  - **Fase de creación de la interfaz gráfica.** Debido al total desconocimiento del sistema de creación de interfaces de MATLAB y su manejo, gran parte de la complejidad de esta fase ha sido el aprendizaje de cómo trabaja MATLAB a nivel de código y variables con las *Graphic User Interfaces* o GUIs. En primer lugar, se realizaron bocetos sobre papel del aspecto deseado del programa y las funcionalidades a incorporar. Con el tiempo disponible para la realización del trabajo, se han conseguido gran parte de los objetivos propuestos. Los no conseguidos han sido bien por limitaciones del software o por enfoques erróneos del problema. En el capítulo 4.3 se detallarán los problemas encontrados y las posibles soluciones o líneas de trabajo en esas direcciones.
  - **Fase de pruebas y elección de métodos.** Tras obtener un compendio de los métodos más empleados en el tratamiento de imágenes histológicas, se ha empleado el software *ImageJ* [17] con los plugins adecuados para ver, de manera aproximada, cuales iban a ser los resultados esperados de la aplicación en MATLAB.
-

- **Fase de análisis de imagen.** Se considera la parte más extensa y el cuerpo de este Trabajo Final de Grado. Para su realización se ha optado por la elección de distintos métodos de tratamiento de imagen, como la deconvolución del color, para obtener imágenes en escala de grises que permitan diferenciar los dos tipos de células. Estos métodos, basados en los artículos originales donde fueron descritos o versiones mejoradas, han sido programados desde cero siguiendo las publicaciones o tomados de *toolboxes* disponibles en la comunidad de MATLAB. Tras la obtención de las imágenes en escala de grises, se ha optado por un proceso personalizado para las muestras disponibles con el fin de adecuar las imágenes y segmentarlas con el método *watershed* con marcadores, obteniendo los resultados y descubriendo así las limitaciones del método. Además, se ha realizado un conteo manual de células para distintas muestras, pudiendo así comparar los métodos automáticos con el manual.

La mayor problemática encontrada ha sido el manejo de la *Whole Slide Image*, ya que su tamaño imposibilita el análisis de toda la imagen obligando a escoger una región de un cierto tamaño que evite la saturación del sistema, como se explica en el Capítulo 4.4. Por otra parte, se ha consumido una gran cantidad de tiempo buscando métodos alternativos, haciendo pruebas y eligiendo los métodos que se han incluido en el trabajo.

- **Fase final: redacción de la memoria.** Organización de la extensa bibliografía y creación y búsqueda de material de apoyo para el desarrollo. Desarrollo en LATEX de la memoria.

Con esta fragmentación, cada parte ha recibido un tiempo adecuado para alcanzar la gran mayoría de objetivos.

## 3.2. Gestión del tiempo dedicado al trabajo

La duración del mismo pretendía ser de 6 meses pero la compaginación con las prácticas de empresa obligó a dilatar ese tiempo hasta Septiembre, por lo que la gestión del tiempo para cada una de las partes descritas anteriormente ha sido la siguiente:

- **Documentación.** *Febrero-Marzo.* Dos meses dedicados a la recopilación de *papers*, literatura e información variada acerca de las muestras biológicas tratadas con Ki-67. Aprendizaje de la creación y manejo de GUIs en MATLAB. Estudio y comprensión de los métodos de análisis de imágenes biológicas y de las librerías
-

necesarias para trabajar con imágenes de microscopio. Reuniones con los tutores del proyecto y planteamiento de las líneas de trabajo a seguir así como de los objetivos y resultados deseados.

- **Creación de la interfaz gráfica en MATLAB.** *Marzo-Abril.* Un mes poniendo en práctica los conocimientos adquiridos sobre GUIs. Realización de bocetos en papel y pruebas de manejo, adecuando la interfaz a la idea de programa práctico e intuitivo para cualquier usuario. Composición de archivos explicativos de ayuda para cada opción de la interfaz. Pruebas con pequeños programas propios para asegurar el funcionamiento y corrección de errores.
  - **Pruebas con software externo.** *Abril-Mayo.* Uso de librerías externas como *OpenSlide* y aplicaciones como *ImageJ* en muestras similares a las proporcionadas por la patóloga para la realización de este trabajo. Obtención de resultados aproximados que facilitan la elección de los métodos a incluir en el trabajo.
  - **Análisis de imagen y procesamiento de imágenes de gran tamaño.** *Mayo-Junio-Julio.* Realización del cuerpo del programa, donde se pueden distinguir tres bloques:
    - **Procesado de imágenes por bloques.** *Mayo:* Ha sido necesario comprender en profundidad el funcionamiento de la función *blockproc* de MATLAB, así como los *ImageAdapters* para poder leer, escribir y modificar las grandes imágenes extraídas con la librería *OpenSlide*. Servirá como base para líneas futuras de trabajo.
    - **Análisis de imagen.** *Junio-Julio.* Se realiza la comparación de los distintos métodos de procesamiento de imagen elegidos y se optimizan para obtener los resultados en las muestras disponibles.
    - **Análisis de datos e imagen. Resultados.** *Julio.* Recopilación de los resultados, extracción de información y muestra de la misma por pantalla.
  - **Redacción de la memoria y preparación de la defensa.** *Agosto.* Composición de la memoria y organización de la bibliografía.
-

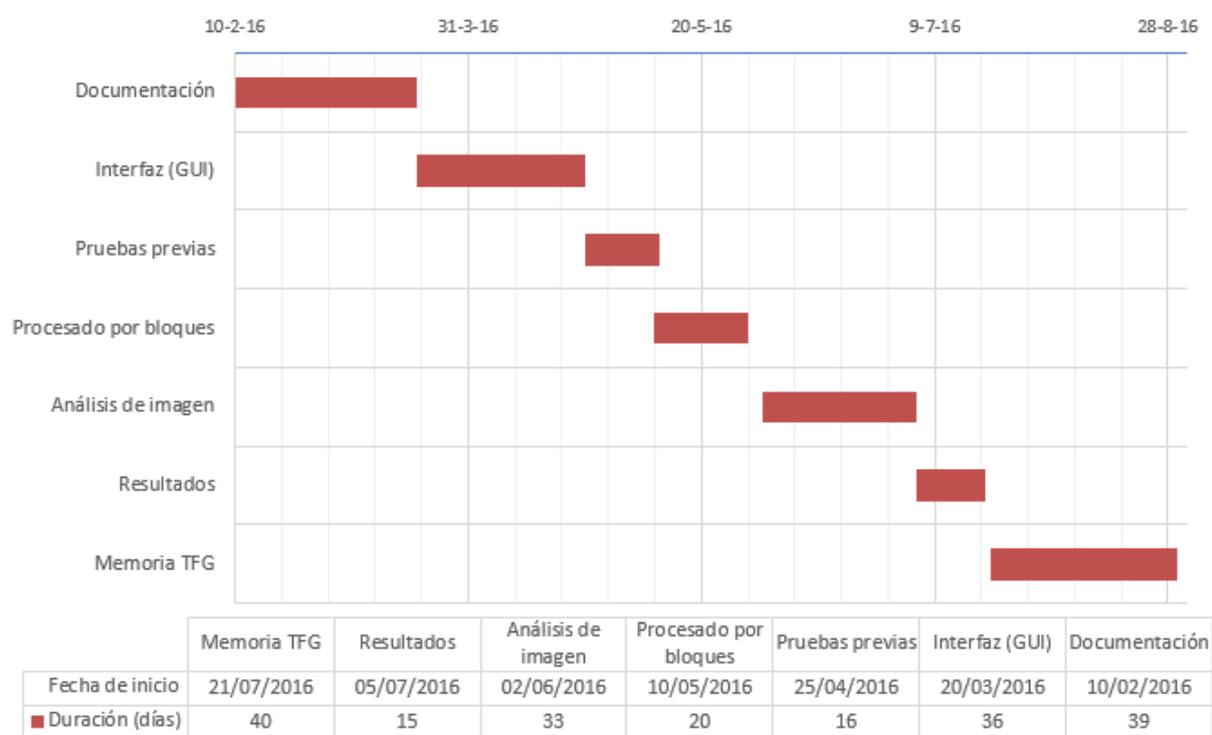


Figura 3.1: Diagrama temporal de la realización del trabajo.

# Capítulo 4

## Desarrollo y métodos utilizados

### Índice de contenidos

---

<b>4.1. Introducción</b>	<b>17</b>
<b>4.2. Librería <i>OpenSlide</i></b>	<b>17</b>
4.2.1. <i>OpenSlide</i> en MATLAB	18
<b>4.3. Interfaz (GUI) en MATLAB</b>	<b>20</b>
4.3.1. Justificación	20
4.3.2. Objetivo	20
4.3.3. Funcionamiento	20
4.3.4. Interfaces de carga y visionado	22
<b>4.4. Procesamiento por bloques. Problemática</b>	<b>28</b>
4.4.1. La función <i>blockproc</i>	28
4.4.2. Visionado profundo de una imagen de máxima resolución	29
4.4.3. Análisis de imagen por bloques	29
<b>4.5. Análisis de imagen</b>	<b>30</b>
4.5.1. Introducción	30
4.5.2. Teoría de los métodos utilizados	31
4.5.3. Procesado de la imagen y segmenta- ción de células	38
4.5.4. Conteo manual y máscara de comprobación	46

---

## 4.1. Introducción

Como se ha descrito anteriormente, la finalidad de este trabajo es sentar las bases para desarrollar una interfaz de usuario sencilla capaz de trabajar con imágenes tomadas con sistemas WSI y, a su vez, analizarlas con distintos métodos de procesamiento de imágenes. A continuación, se describirán para cada apartado los conceptos necesarios para su realización.

## 4.2. Librería *OpenSlide*

*OpenSlide* [15] es una librería de libre distribución que proporciona un método para visualizar y explorar los archivos creados por los microscopios WSI de los principales fabricantes del mercado. Actualmente ofrece soporte en distintos lenguajes de programación, habiéndose usado para este TFG tanto su interfaz en Java como las librerías adaptadas a MATLAB [18].

El primer paso para entender cómo están estructuradas las imágenes del fabricante Ventana, del cual se han obtenido las muestras disponibles para el trabajo, pasa por explorarlas con el visor de *OpenSlide*.

En la Figura 4.1 se muestran algunas de las propiedades contenidas dentro de los archivos BIF propios del fabricante *Ventana*.



key	value
openslide.comment	level=0 mag=20 quality=90
openslide.level-count	10
openslide.level[0].downsample	1
openslide.level[0].height	49359
openslide.level[0].width	61376
openslide.level[1].downsample	2
openslide.level[1].height	24680
openslide.level[1].width	30688
openslide.level[2].downsample	4
openslide.level[2].height	12340
openslide.level[2].width	15344
openslide.level[3].downsample	8
openslide.level[3].height	6170
openslide.level[3].width	7672
openslide.level[4].downsample	16

Figura 4.1: Algunos de los campos que muestra la librería *OpenSlide* al abrir un archivo BIF.

Como se puede observar, dentro del archivo de trabajo se encuentra mucha información además de la imagen de interés. En el caso de las muestras trabajadas, se dispone de una pirámide de imágenes de 10 niveles (siendo 0 el de mayor resolución), donde se especifica el factor de diezmado respecto al original, el tamaño de la imagen de cada nivel, la fecha de creación, el nivel de aumento ( $20\times, 40\times$ ), el fabricante y un largo etcétera. Entendiendo el contenido de las muestras a tratar, se puede comenzar a hacer uso de la librería para MATLAB.

### 4.2.1. *OpenSlide* en MATLAB

A continuación se detallarán las funciones de la librería que han sido utilizadas y cuál ha sido su uso.

- **`openslide_load_library`.**

Carga la librería de *OpenSlide* en MATLAB para poder utilizar las funciones pertinentes.

- **`openslidePointer = openslide_open(slideName)`.**

A partir del directorio completo del archivo a analizar, genera un puntero a dicho archivo necesario para trabajar con el resto de funciones.

- **`openslide_read_associated_image(openslidePointer, imageName)`.**

Normalmente, además de la pirámide de imágenes, el archivo incluye otras imágenes que pueden ser de interés, como una miniatura de la imagen global o una fotografía de la muestra “macro”, como la que se observa en la Figura 4.2. En este caso se ha utilizado la miniatura para una previsualización de la imagen seleccionada por el usuario en la interfaz de carga.



Figura 4.2: Imagen asociada “macro” de una de las muestras disponibles.

- **`openslide_get_property_names(slidePtr)`.**

A partir del puntero al archivo, obtiene el nombre de todas las propiedades en el orden que se observan en la Figura 4.1.

- **`propertyValue = openslide_get_property_value(openslidePointer, propertyName)`.**

Una vez obtenido el nombre de las distintas propiedades, se puede hacer uso de los valores de las mismas con esta función. Por ejemplo, durante la realización de la interfaz se hace uso de esta función un gran número de veces para obtener los datos de tamaño de las imágenes en distintos niveles de la pirámide.

---

**Ejemplo:** *Se desea saber la altura y anchura de la imagen en el nivel uno de la pirámide.*

Hay que recordar que, como se ve en la Figura 4.1, las propiedades de dimensión de las imágenes comienzan a partir del tercer campo, que corresponde al factor de diezmado del nivel 0 de la pirámide. Como se quiere obtener la altura y anchura del nivel uno, se creará una expresión que obtenga su posición para cualquier nivel. Para este caso, las posiciones siete y ocho.

```
propertyNames = openslide_get_property_names(slidePtr);
posicion.altura=3*(nivel_piramide+1)+1;
Posición de la altura del nivel requerido en el vector
propertyNames
posicion.anchura=3*(nivel_piramide+1)+2;
alto= str2double(openslide_get_property_value(slidePtr,...
propertyNames(posicion.altura)));
ancho= str2double(openslide_get_property_value(slidePtr,...
propertyNames(posicion.anchura)));
```

- **`openslide_read_region(openslidePointer,xPos,yPos,width,height,level)`.** Esta función lee una región rectangular de la imagen del nivel especificado por *level*. Para ello, se han de introducir como parámetros el puntero al archivo, el origen de coordenadas del rectángulo (*xPos,yPos*) y el tamaño del rectángulo (*width,height*).

Es necesario tener en cuenta que todas las imágenes que se carguen constan de 4 componentes, el canal de transparencia o alfa y los tres canales RGB, por lo que durante todo el proceso de emplean los últimos 3 canales (RGB).

**Ejemplo:** Muestra de una región de 200x200 con origen en 0,0 (esquina superior izquierda de la imagen) del nivel de pirámide deseado.

```
imagen= openslide_read_region(slidePtr,0,0,200,200, 'level',
nivel_piramide);
imagen=imagen(:, :, 2:end);
imshow(imagen)
```

- **`openslide_close()`.** Cierra la librería *OpenSlide*.
-

## 4.3. Interfaz (GUI) en MATLAB

### 4.3.1. Justificación

La interfaz gráfica es la piedra angular del trabajo ya que es la encargada de permitir al usuario seleccionar una imagen de microscopía en formato BIF, visualizarla y analizarla de manera sencilla e intuitiva.

La necesidad de implementar esta interfaz radica en la imposibilidad que existe de trabajar con los archivos propios de las imágenes histológicas de manera usual, con visores o analizadores convencionales. Para trabajar con este tipo de imágenes existen dos opciones: usar el software especializado que ofrecen las marcas vendedoras de WSI o crear un software propio con alguna librería de código abierto como es el caso de *OpenSlide*.

### 4.3.2. Objetivo

La misión principal de la interfaz gráfica será la de asegurar en todo momento que el usuario trabaje con una imagen que su ordenador sea capaz de soportar. Dado que al leer una imagen con MATLAB éste la carga directamente en RAM, se ha optado por limitar el tamaño de las imágenes. Esto se ha conseguido gracias a dos factores:

- La función *memory* de MATLAB, la cual se encarga de obtener la información de la memoria disponible para trabajar, y el cálculo previo del tamaño de la imagen que se pretende cargar antes de hacerlo, obtenido a través de *OpenSlide*.
- Una vez seleccionada una imagen que el usuario sea capaz de manejar, el otro objetivo de la interfaz es permitir, realizar un análisis con un método elegido por el usuario entre los propuestos en este trabajo.

Con ambos objetivos en mente, se han creado dos interfaces dependientes cuyo funcionamiento se detalla a continuación.

### 4.3.3. Funcionamiento

El primer paso para crear una interfaz gráfica es sencillo: decidir cuáles serán sus funcionalidades. Con esa idea, lo más aconsejable es realizar unos bocetos en papel de la distribución de botones y opciones y más tarde intentar plasmar todas esas ideas en código.

---

Para el caso que compete a este TFG, se han creado dos interfaces distintas y dependientes entre sí para asegurar el cumplimiento de objetivos.

Ambas interfaces han sido creadas con la *toolbox* que incorpora MATLAB para tal fin. Esta herramienta permite configurar de forma gráfica la disposición de botones y funciones de la interfaz para posteriormente configurar cada una de sus características a partir de una función de MATLAB que controlará el sistema.

Dicha función constará de tres partes:

- **Función de inicialización.** MATLAB la crea automáticamente al generar una nueva GUI y no ha de ser modificada. Define el programa como una interfaz.
- **Función de apertura u *OpeningFcn*.** Se encarga de inicializar variables, figuras, interpretar los datos recibidos de alguna llamada externa y, en definitiva, presentar la interfaz tal como debe verse.
- **Función de salida u *OutputFcn*.** En caso de necesitar que la interfaz devuelva un resultado, el código para mostrarlo se incluirá dentro de esta función.

El resto del código se programa como una función normal de MATLAB, con la diferencia de que cada función creada no necesita terminarse con el comando *end*, sino que van todas en cascada. Además, cada botón agregado a través de la *toolbox* creará automáticamente una función *callback* que será la ejecutada cada vez que se presione dicho botón.

#### Las variables *handles* y *hObject*.

La característica fundamental de las GUIs en MATLAB es que disponen de dos elementos para comunicar las funciones que la constituyen: *hObject* y *handles*.

La variable *hObject* puede definirse como un espacio común donde la interfaz deposita toda la información de su estado actual. En ella se podrían encontrar todas las variables globales que pueden ser utilizadas en cualquier función de la interfaz si se sigue el procedimiento adecuado.

La variable *handles* es una estructura utilizada en cada función de la interfaz sobre la que se pueden almacenar distintas variables. Al comienzo de una función se pueden cargar las variables globales con *handles = guidata(hObject)* y al final, si se quieren actualizar dichas variables o añadir nuevas, se utiliza el comando *guidata(hObject, handles)*.

---

**Ejemplo:** Realizar una función dentro de la interfaz que calcule una suma y utilizar el resultado en otra función de la misma interfaz.

```
function [] = suma(hObject)
```

Se crea la variable resultados dentro de la estructura handles.

```
handles.resultado=50+50;
```

Se almacenan las estructuras de la función en el espacio hObject

```
guidata(hObject, handles)
```

```
function [] = resta (hObject)
```

Se obtienen las estructuras almacenadas en hObject

```
handles = guidata(hObject);
```

Se crea una nueva variable en handles que usa el valor de la suma

```
handles.resultado2=handles.resultado-10;
```

Se almacenan los nuevos handles de nuevo en el hObject.

```
guidata(hObject, handles)
```

#### 4.3.4. Interfaces de carga y visionado

##### Interfaz de carga

La interfaz de carga tiene como finalidad permitir al usuario cargar un archivo BIF y previsualizar su miniatura obtenida a través de la librería *OpenSlide*. En caso de haber elegido una imagen que no es la deseada, se permite la opción de descargar la imagen del programa y probar de nuevo con otra opción.

Tras pulsar sobre la miniatura, se determinará la dimensión del archivo que el sistema es capaz de mostrar. Para ello se calculará el tamaño en *bytes* que ocupará en memoria cada imagen de la pirámide desde el nivel de máxima resolución y de forma decreciente, hasta encontrar una resolución que no sobrepase la memoria disponible del sistema.

```
memoria_elemento_piramide(i+1)=width*height*4*8;
```

```
sys.PhysicalMemory.Available >memoria_elemento_piramide(i+1)
```

Una vez seleccionada la imagen adecuada, se llamará a la segunda interfaz, a la que se le enviará la información del puntero *OpenSlide* del archivo, las dimensiones de la imagen a ser mostrada, el nivel de pirámide seleccionado y el directorio del archivo BIF.

```
visor(slidePtr,width,height,nivel_piramide,PathName,FileName)
```

---

En la Figura 4.3 se representa la estructura de funcionamiento de la interfaz de carga.

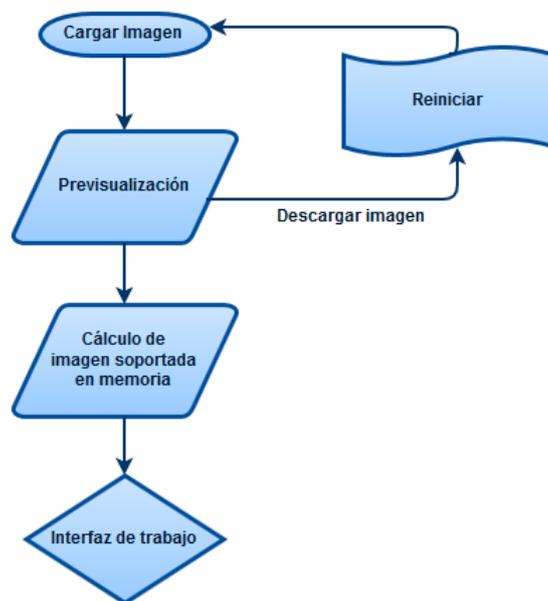


Figura 4.3: Flujograma del funcionamiento de la interfaz de carga creado con Draw.io [19].

### Interfaz de trabajo

Es el entorno real de usuario del trabajo. Funciona como visor de la imagen de la muestra seleccionada y permite guardar lo que se está viendo en distintos formatos de imagen.

La imagen presentada al usuario será aquella de la pirámide de imágenes del archivo BIF que pueda ser cargada por el usuario sin saturar el sistema. Para ello las propiedades de dicha imagen habrán sido previamente calculadas en la interfaz de carga.

Sus funcionalidades, además de las de desplazarse por la imagen y hacer *zoom* sobre la misma, se describen a continuación.

#### ■ Seleccionar región

Haciendo uso de las funciones *imrect* y *getPosition* de MATLAB se puede crear una herramienta que permita al usuario seleccionar una región rectangular de la imagen visualizada, como se muestra en la Figura 4.4.

### Ejemplo

Creación del rectángulo mediante el ratón.

```
h=imrect;  
rectangulo = wait(h);  
rectangulo=getPosition(h);  
delete(h);
```

La variable *rectángulo* recibirá su contenido cuando el usuario haga doble clic una vez seleccionada la región de interés. El resultado consta de 4 elementos, en la primera columna se tiene el valor inicial del eje X y el número de píxeles que ocupa la región en el eje X. En la segunda columna se tiene lo mismo para el eje Y.

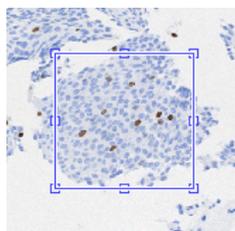


Figura 4.4: Selección de una región de la imagen.

Tras la selección de la región deseada y si el visor no se encuentra ya en el nivel de la pirámide de imágenes de máxima resolución (en cuyo caso simplemente se hará zoom en la región seleccionada), comenzará el cálculo de tamaños en memoria para realizar la carga de la nueva imagen.

En primer lugar, se calculará el tamaño que tiene la imagen en el nivel de pirámide actual. Posteriormente, y de forma decreciente, se calculará el tamaño de la imagen desde el nivel actual hasta el de máxima resolución. Para cada descenso de nivel, se calculará el ratio por el que se ha de escalar la región seleccionada por el usuario calculando el factor que hay entre la dimensión del nivel original (*full\_height*) y la del nivel inferior (*sub\_height*). De esta forma, se obtiene la correspondencia de la imagen seleccionada por el usuario en una imagen de mayor resolución.

```
wratio=round(sub_height/full_height);
```

Tras obtener el ratio, se multiplicará por la variable *rectángulo* y se calculará el tamaño en bytes que tiene la nueva imagen en memoria. El proceso se repetirá hasta que la imagen no pueda ser cargada en memoria, con lo que las condiciones

---

de parada son o bien no quedar más niveles en la pirámide o bien alcanzar la restricción impuesta por el límite de memoria.

Una vez alcanzada la condición de parada, se actualizarán las variables de nivel de la pirámide actual y se cargará la nueva imagen.

- **Original**

El botón etiquetado como *Original* devolverá la interfaz, sea cual sea el estado actual, a la imagen original que fue mostrada al cargar el programa.

- **Guardar**

El botón Guardar abrirá una ventana del explorador de archivos con la finalidad de guardar la imagen actualmente visualizada en la interfaz en el sistema, ya sea en formato PNG o JPEG.

- **Analizar**

Mostrará un desplegable, presentado en la Figura 4.5, con las distintas opciones de análisis de imagen. Tras la selección de uno de los métodos, se procederá a realizar el estudio, cuyo proceso será descrito posteriormente. Una vez realizado el análisis, se presentarán los resultados y se guardarán en el sistema.

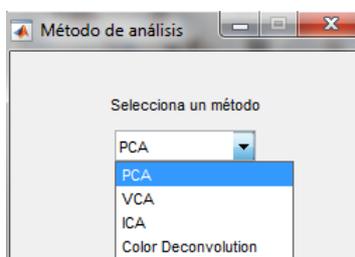


Figura 4.5: Desplegable con las distintas opciones de análisis.

Por último, en la Figura 4.6 se representa el funcionamiento de la interfaz de trabajo en forma de flujograma.

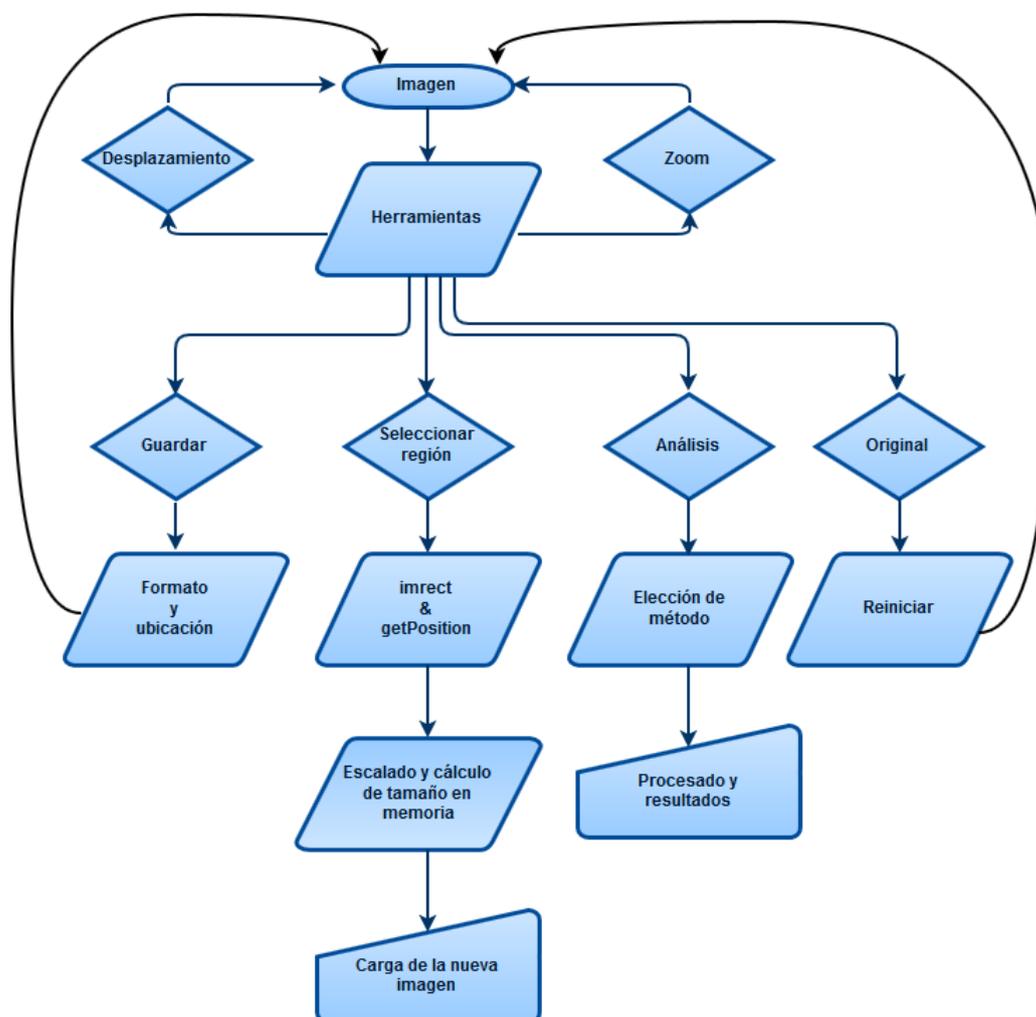


Figura 4.6: Flujograma del funcionamiento de la interfaz de trabajo.

Más adelante se representará de forma extendida el funcionamiento del apartado de análisis que aparece simplificado en este diagrama.

Como detalle, en las Figuras 4.7 y 4.8 se observa el aspecto de las dos interfaces con un archivo cargado.

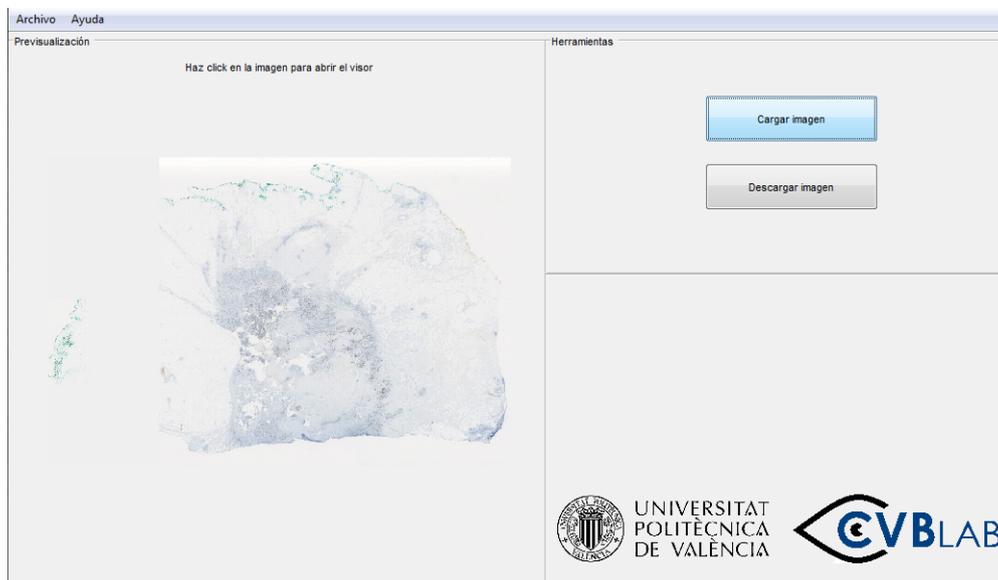


Figura 4.7: Aspecto de la interfaz de carga.

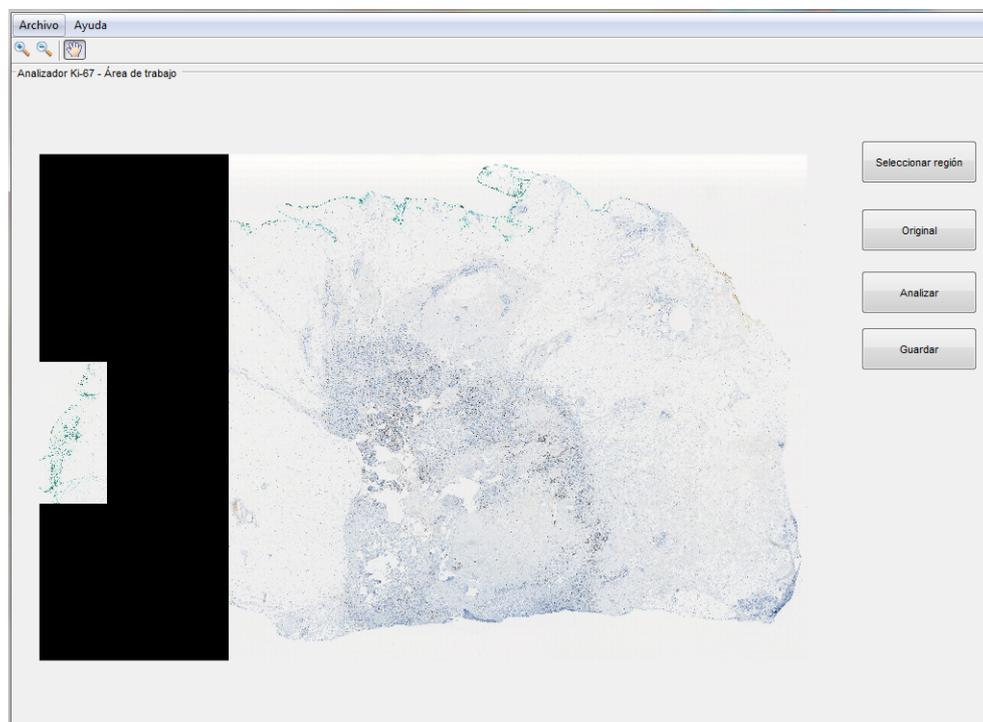


Figura 4.8: Aspecto de la interfaz de trabajo.

## 4.4. Procesamiento por bloques. Problemática

Dada la problemática del tamaño de las imágenes de microscopía y la imposibilidad de cargarlas y visualizarlas en su máxima resolución sin un software especializado, en primera instancia se buscaron métodos disponibles en MATLAB para poder solventar dichos inconvenientes.

### 4.4.1. La función *blockproc*

MATLAB cuenta con una función para tratar con archivos demasiado grandes para ser cargados como variables en RAM. Dicha función se denomina *blockproc*.

La función *blockproc*, en su forma estándar, divide la imagen en formato TIFF, NITF, o JPEG2000 en bloques de un tamaño especificado y, uno a uno, les aplica una función de procesamiento personalizable por el usuario. Posteriormente, va colocando esos bloques en el mismo orden que estaban originariamente creando el resultado de la Figura 4.9. La forma de “coser” dichos bloques es también personalizable, permitiendo incluso variar el grosor de la línea de la costura para hacerla casi invisible.



Figura 4.9: Representación del procesamiento por bloques de una imagen tomada de [20].

Ahora bien, de dicha forma la función no puede ayudar mucho en el objetivo planteado, ya que requiere que la imagen esté cargada en memoria. Pero existe la forma de configurar *blockproc* para realizar dicho procesamiento por bloques sin cargar la imagen entera, sólo sus bloques.

Siguiendo la serie de tutoriales de *Steve Eddins* en Mathworks [20] [21] [22] sobre el procesamiento de grandes imágenes y los métodos que aparecen en [23], se plantearon distintas alternativas que por algún motivo u otro no han sido suficientes para cumplir los objetivos propuestos. A continuación se describirán las medidas tomadas, las cuales quedarán como bases para futuras líneas de trabajo con la finalidad de mejorar la interfaz y su rendimiento.

---

#### 4.4.2. Visionado profundo de una imagen de máxima resolución

En lugar del método utilizado en la interfaz gráfica realizada fue una primera intención implementar un sistema de visualización similar al utilizado en las aplicaciones de mapas online como GoogleMaps. De esta forma se consigue que se muestre en pantalla la imagen completa en baja resolución y a medida que se le aplique zoom se vayan cargando imágenes de mayor resolución dando la sensación de continuidad.

MATLAB cuenta con una forma de realizar dicho proceso gracias a la función *rset* [22]. Dicha función crea una versión multiresolución de un archivo TIFF almacenado en disco y lo muestra por pantalla con la herramienta *imtool* de MATLAB. La no implementación de este método en la interfaz se ha debido a que, de momento, no es posible visualizar los archivos *rset* sobre una figura de visualización de MATLAB y han de ser obligadamente visualizados con *imtool* lo cual imposibilita la integración del visor con la interfaz, dejando dos instancias de programas separadas.

#### 4.4.3. Análisis de imagen por bloques

Además de la opción de analizar una región específica de la imagen, se abordó la alternativa del análisis de la imagen completa en alta resolución. Esto llevaría mucho tiempo pero permitiría obtener un resultado global de toda la muestra histológica en lugar de buscar zonas específicas. Para su realización, se recurrió a la ayuda, una vez más, de los tutoriales creados por *Steve Eddins*, referentes a los *ImageAdapters* [21] y al procesado por bloques [20].

##### **ImageAdapters.**

Dado que *blockproc* sólo funciona con imágenes en formato TIFF, NITF, o JPEG2000 y en el caso de interés se está utilizando un archivo BIF, existe un método para engañar a la función *blockproc* y que crea que la entrada está en uno de esos formatos. Se les conoce como *ImageAdapters*.

A partir del ejemplo proporcionado por Steve, se han realizado un par de *ImageAdapter* propios para hacerlos compatibles con la librería *OpenSlide*, de forma que se puedan leer y escribir las imágenes por bloques con la función *openslide\_read\_region()*. De esta forma, se ha creado una función que extrae la imagen de alta resolución del archivo BIF y la guarda en el disco duro en formato TIFF. A continuación, se repite el proceso leyendo el archivo del disco y procesándolo por

---

bloques con alguna función de análisis de imagen de las propuestas.

Uno de los motivos por los que el método no ha sido implementado en el programa final ha sido debido a que las características locales de los bloques no son suficientes para realizar una segmentación homogénea de la imagen completa. Para poder realizar el procesado por bloques sería necesario buscar una forma de compartir la información relevante para la segmentación entre los distintos bloques.

Además, existe una problemática que no se ha sabido resolver correctamente respecto a la elección del tamaño de los bloques [24]. Los archivos de imagen TIFF pueden ser ordenados a la hora de componerse de dos formas distintas, en *strips* o *tiles*. Dependiendo de esta construcción de la imagen, ha de realizarse la selección del tamaño de bloque de una forma u otra para que sea eficiente. Se ha optado por un método de escritura de la imagen en *strips* para su posterior lectura de la misma forma, pero dependiendo del tamaño de bloque elegido se pueden llegar a observar repeticiones en la imagen.

Por estos motivos, el procesado y visionado por bloques queda como tarea pendiente para futuras mejoras.

## 4.5. Análisis de imagen

### 4.5.1. Introducción

Dada la alta resolución de las imágenes histológicas, la cantidad de detalles tanto en tamaño, forma, color o textura que se puede encontrar es desproporcionadamente alta. Esto complica seriamente el objetivo del análisis, que en pocas palabras consiste en distinguir dos tipos de objetos: células azules y células marrones. La tarea de segmentar dichas células ya es de por sí complicada sin tantos elementos, como se verá en el apartado correspondiente, por lo que se precisa de algún método que reduzca esa cantidad de información sin perder la de interés.

A continuación se expondrá el contenido teórico de los métodos matemáticos más usados para extraer características de una imagen en la que existen mezclas, como la mezcla de colores en el caso de las imágenes de histología. Estos métodos convierten una única imagen RGB en distintas imágenes en escala de grises en las que se acentúan características concretas.

---

### 4.5.2. Teoría de los métodos utilizados

#### Análisis de Componentes Principales

El Análisis de Componentes Principales, PCA [25] en inglés, es una técnica que realiza una transformación lineal del conjunto de datos situándolos en unos nuevos ejes de coordenadas donde se maximiza la separación entre los distintos elementos.

Esta transformación se realiza calculando los autovectores de la matriz de covarianza de los ejes de coordenadas. Estos autovectores se ordenan de mayor a menor varianza explicada, definida por sus autovalores asociados. Así, los ejes que proporcionan una mayor separación entre los elementos del conjunto de datos quedan en primer lugar. De esta forma, se reduce la dimensionalidad del conjunto de datos ya que, una vez realizada la transformación, es posible seleccionar solamente las primeras componentes y descartar las demás, que apenas contienen información relevante y normalmente contiene gran parte del ruido aleatorio. Un ejemplo de esta reducción se muestra en la Figura 4.10

La transformación del método PCA se realiza en los siguientes pasos:

- Traslación del origen de referencia del conjunto de datos original al punto medio o centroide de los mismos.
  - Normalización de los datos por su desviación estándar.
  - Cálculo de la matriz de covarianzas de la muestra y, posteriormente, cálculo de los autovalores de dicha matriz.
  - Ordenación de mayor a menor variabilidad, la cual se indica por la magnitud de los autovalores. Los primeros autovalores tienen asociadas las mayores varianzas.
  - Cálculo de los nuevos ejes de referencia a partir de los autovectores correspondientes a los autovalores, lo cual es posible ya que estos vectores cumplen las condiciones para formar una base ortogonal. Para reducir la dimensionalidad, los datos se proyectan sobre los autovectores con una mayor varianza asociada.
-

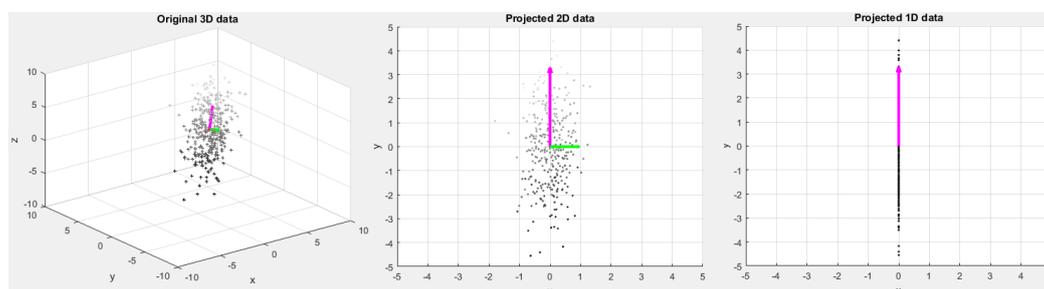


Figura 4.10: Ejemplo de reducción de dimensionalidad. De izquierda a derecha: Datos originales y autovectores. 3D > Proyección sobre los dos autovectores de mayor variabilidad. 2D > Proyección sobre el mayor autovector. 1D. Ejemplo adaptado de [26].

Para la aplicación de todos los métodos en la interfaz de este TFG, tanto PCA como los que se describirán en adelante, se parte de las siguientes líneas:

```
[a,b,c] = size(im); La variable im es la imagen a analizar
im = reshape(double(im),a*b, []); Valores RGB en columnas
Nc = 3; Número de endmembers / Componentes principales
X = -log((im+eps)/255); Valores de Absorbancia
```

La aplicación del método PCA para obtener las imágenes procesadas se ha realizado de la siguiente manera:

```
[coeff,score] = pca(X,'NumComponents',Nc);
La variable coeff contiene los coeficientes de las componentes
principales y la variable score contiene la representación de X
en el nuevo espacio.
for i = 1:Nc
preprocesado(:, :, i) = uint8(255 * mat2gray((reshape(score(:, i) ...
, a, b))));
end
```

### *Vertex Component Analysis*

La finalidad del método *Vertex Component Analysis*, VCA [27] en inglés, es encontrar el número de sustancias puras o *endmembers* que se ubiquen en un conjunto de datos donde aparecen mezclados distintos espectros. Dicho de otra forma, es un proceso que mediante iteraciones busca los puntos extremos o *endmembers* de un conjunto de datos como pueden ser los valores RGB de una imagen.

En el caso de las imágenes HDAB, los *endmembers* serían píxeles donde se considera que la sustancia es pura, es decir, un píxel para la hematoxilina, otro para el DAB y otro para el fondo.

El algoritmo VCA aprovecha dos características fundamentales. La primera, que los *endmembers* son los vértices de un *simplex* y la segunda que la transformación afín [28] de un *simplex* es otro *simplex*. Un *simplex* [29] representa la envoltura convexa más simple que se pueda dar para un conjunto finito de  $(n+1)$  puntos independientes afines a un espacio euclídeo de dimensión  $n$  o mayor. Por tanto, los vértices de esa envoltura o los puntos extremos del conjunto de datos serán los *endmembers* que interesa encontrar. Es decir, estos vértices representan una forma de representar los datos de forma que las variables estén más separadas entre sí.

Para su uso en MATLAB se ha utilizado el código desarrollado por *José Nascimento* y *José Bioucas Dias* [31] incorporándolo a la parte de análisis del programa. El algoritmo crea un subespacio del espacio original a partir de un número de *endmembers* dado por el usuario y proyecta el resto de puntos como combinaciones de los *endmembers* encontrados, como se observa en la Figura 4.11. El *endmember* resultante de cada iteración será el extremo de la proyección y equivaldría a una de las sustancias puras del conjunto de datos.

Éste método sólo es eficaz cuando realmente existen píxeles que representen dicha sustancia pura, pero en el caso de las imágenes utilizadas, la iluminación y la heterogeneidad del tejido analizado hace muy difícil localizar un píxel puro para cada tinte.

Como se verá más adelante, la deconvolución de color podría considerarse un caso particular del VCA ya que lo que se hace es fijar los factores de la transformación de ejes que se aplica para obtener los valores de las componentes deseadas.

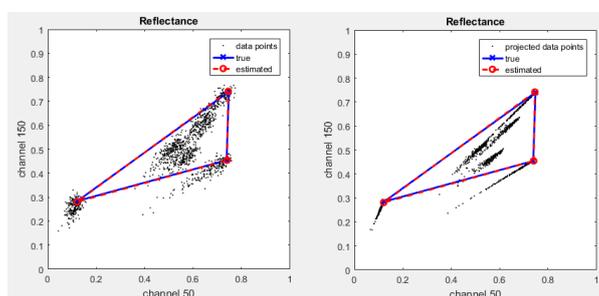


Figura 4.11: A la izquierda, se observa la representación de los puntos del conjunto de datos y la estimación de los *endmembers* en rojo. A la derecha, se observan los puntos proyectados del conjunto de datos y los tres *endmembers* encontrados. Imágenes obtenidas de la demostración 1 del código [31].

La aplicación de la función para obtener las imágenes procesadas, cuyo ejemplo de funcionamiento se muestra en la Figura 4.12, se ha realizado de la siguiente manera:

```
[Ae, indice, Rp]= vca(X', 'Endmembers', Nc);
Ae es la matriz con los coeficientes de descomposición Rp es la
matriz con los píxeles proyectados.
s_est = (pinv(Ae) * Rp)';
s_est contiene los mapas de abundancia, que son los coeficientes
que tienen los endmembers para cada píxel
for i = 1:Nc
preprocesado(:, :, i)=uint8(255*(reshape(s_est(:, i), a, b)));
end
endmembers = X(indice, :); En la variable indice está la posición
de los píxeles endmembers
endmembersRGB = reshape(uint8(exp(-endmembers)*255), 1, Nc, 3);
```

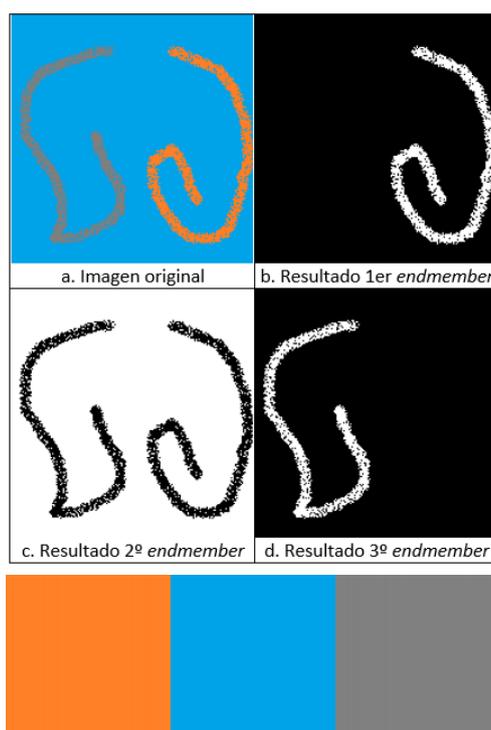


Figura 4.12: Aplicación de VCA en imagen con tres colores. En la parte inferior de la imagen se observan los *endmembers* encontrados por el algoritmo. En la parte superior se encuentra la separación de la imagen original en tres imágenes donde, en cada una, se encuentran en blanco todos los píxeles que pertenecen a cada *endmember*.

### Análisis de Componentes Independientes

El método del Análisis de Componentes Independientes, ICA [32] en inglés, tiene como finalidad encontrar las componentes estadísticamente independientes entre sí en un conjunto de datos no gaussianos. Un ejemplo de esto sería una grabación de distintos sonidos, como pueden ser un instrumento y una voz, a través de dos micrófonos. El resultado es una mezcla de señales con características particulares. ICA es pues un método que intenta obtener como resultado la separación de esa mezcla a partir de la expresión:

$$x(t) = A * s(t) \quad (4.1)$$

donde  $x(t)$  sería el conjunto de datos de que se disponen,  $s(t)$  sería el conjunto de variables independientes y  $A$  una matriz columna de transformación cuadrada.

Por lo que estimando la matriz  $A$  podemos obtener las componentes independientes de la siguiente forma:

$$s(t) = A * A^{-1} * x(t) \quad (4.2)$$

El punto de partida del método ICA es suponer que las componentes  $s(t)$  son estadísticamente independientes. Esto quiere decir que el valor de una componente no afecta al valor de otra de las componentes y viceversa, por lo que se asume que en la señal mezclada originalmente habían componentes independientes.

Hay muchos algoritmos distintos de aplicación del ICA y en este trabajo se ha utilizado el conocido como *FastICA* [32]. El *FastICA* es un método iterativo que a partir de un vector inicial de pesos  $w$ , busca nuevos pesos de forma que la proyección de la Expresión 4.3 maximice la no gaussianidad.

$$w^T * x(t) \quad (4.3)$$

Antes de comenzar las iteraciones, los datos han de ser centrados y “blanqueados”. Esto quiere decir que han de tener media nula y que la varianza de todas sus variables ha de ser igual a uno. Tras esto, se inicializa de forma aleatoria el vector de pesos. Posteriormente, comienza un método iterativo hasta que los pesos anteriores y los actuales converjan en una misma dirección. Tras la convergencia, el vector de pesos  $w$  pasa a ser  $A$  y se aplica la Ecuación 4.1.

---

Los pasos de la iteración son dos:

- **Actualización de los pesos.** Se guarda el valor del último vector de pesos disponible y se calculan los nuevos pesos.
- **Decorrelación de los pesos calculados.** Se realiza la decorrelación de los pesos y se normalizan, dando lugar a los nuevos pesos.

Para su implementación en el trabajo se ha utilizado la adaptación a MATLAB de *Brian Moore* [33], que implementa el algoritmo *FastICA*.

```
[Zica, A, T, muica] = myICA(X', Nc);
for i = 1:Nc
preprocesado(:, :, i) = uint8(255*mat2gray((...
reshape(Zica(i, :)', a, b))));
end
```

### Deconvolución de color manual

La técnica de la deconvolución de color, propuesta por *C. Ruifrok* y *Dennis A. Johnston* [34], es un método creado para mejorar el análisis de las imágenes de muestras histológicas tratadas con algún marcador celular.

Su finalidad es separar los valores de absorbancia de las componentes RGB de la imagen en distintas imágenes, cada una con los valores individuales de absorbancia de cada componente del tinte. Por poner un ejemplo, teniendo una imagen de una muestra tratada con Ki-67, donde se aprecian las células azules, marrones y el fondo, lo que interesa es separar las células azules y marrones en imágenes distintas como se muestra a continuación en la Figura 4.13.

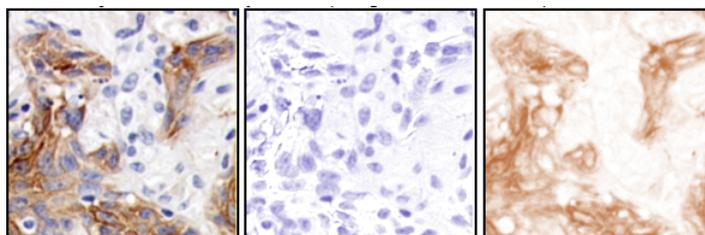


Figura 4.13: Ejemplo de resultado de la deconvolución de color en HDAB tomada de [35]. De izquierda a derecha, imagen original > deconvolución componente azul > deconvolución componente marrón

La implementación del método consiste en transformar la imagen de RGB a sus

---

valores de densidad óptica (OD) como se indica en [34]

$$OD = -\log\left(\frac{I}{I_o}\right) \quad (4.4)$$

Donde  $I$  es la imagen original,  $I_o$  la intensidad de la luz transmitida (se toma como valor de referencia 255) y  $OD$  la imagen en el espacio de densidad óptica.

Posteriormente, asumiendo que cada componente del tinte se caracteriza por una densidad óptica concreta para cada una de las componentes del espacio de origen, para un espacio RGB se obtiene una matriz de transformación  $M$  de 3x3 (una fila por cada componente del tinte y una columna por cada componente del espacio de la imagen). Es importante mencionar que los valores se obtienen manualmente a partir de muestras donde se encuentra exclusivamente el color de cada componente del tinte, sin mezclar, para evitar la contribución de otros colores. Por tanto, para obtener la imagen  $d$  donde cada componente corresponde a un color del tinte, se realiza la operación:

$$d = M^{-1} * OD \quad (4.5)$$

Para el uso del método en el trabajo se ha empleado una *toolbox* [36] para MATLAB que incorpora ciertas mejoras en la aplicación del algoritmo.

Para la obtención de la matriz de transformación se ha utilizado el software ImageJ [17] ya que, aunque las imágenes del tinte HDAB se asemejan en colores a las de Ki67, los resultados obtenidos con los valores de referencia para las muestras disponibles son bastante mejorables, puesto que no realiza una correcta separación de los distintos tipos de objetos. El software tiene la opción [35] de calcular los valores de la matriz de transformación a partir de una región seleccionada por el usuario y, dado que no se disponen de muestras únicas de cada componente de color, se ha seleccionado una región de un píxel en el centro de las células más características por su alta coloración. Dado que no es un método perfecto, las imágenes resultantes contienen un alto componente de ruido que habrá que filtrar posteriormente.

Con este método, la matriz de coeficientes obtenida y por tanto empleada ha sido:

$$M = \begin{pmatrix} 0.56234616 & 0.6493857 & 0.51192284 \\ 0.7156873 & 0.6182272 & 0.32494134 \\ 0.5400141 & 0.5400141 & 0.6455769 \end{pmatrix}$$


---

### 4.5.3. Procesado de la imagen y segmentación de células

Como se mencionó en el planteamiento del proyecto, para la realización del tratamiento de las imágenes se ha seguido una serie de pasos que serán descritos a continuación:

#### Métodos de deconvolución del color

Los métodos descritos anteriormente (PCA, ICA, VCA y la deconvolución de color manual) se aplican sobre la imagen RGB extraída mediante la interfaz de trabajo con la finalidad de obtener una serie de imágenes en escala de grises donde sea más fácil distinguir los dos tipos de células que se quieren segmentar por separado.

#### Preprocesado

Antes de proceder a segmentar las células, es necesario eliminar el ruido y adecuarlas, en la medida de lo posible, a una imagen ideal donde sólo se encuentren los elementos susceptibles de ser contabilizados. Para este proceso se han utilizado operadores morfológicos [37] [38], en concreto se ha usado la apertura y cierre por reconstrucción seguida de un cierre común y una erosión. Como guía del proceso se ha optado por una combinación de los métodos descritos en [39] y [40].

Este preprocesado conlleva una problemática, como se verá en la interpretación de los resultados, y es que el tamaño de las células varía según las muestras. Esto quiere decir que hay que ser muy cauteloso a la hora de aplicar los operadores morfológicos ya que, dependiendo de la dimensión de los elementos, se pueden eliminar objetos de interés que no deberían descartarse.

La elección del tamaño mínimo de los objetos deberá elegirse para cada imagen haciendo una interpretación del tamaño aproximado de las células que se desean segmentar. Como solución, se ha optado por incorporar una opción en la que el usuario seleccione con un recuadro la célula más pequeña del tipo a segmentar y se toma el valor del lado más largo del rectángulo como radio mínimo para el tamaño de los objetos que se quieren encontrar. Para la toma de resultados se ha optado por una solución de compromiso fijando un valor útil para las muestras a  $20 \times (8)$  y otro para las muestras a  $40 \times (16)$ .

---

Por tanto, la implementación para  $40\times$  quedaría de la siguiente forma:

```
r_min=16;
r_max=r_min*3;
se = strel('disk', 4);
im_proc = imdilate(im_proc, se);

Apertura por reconstrucción.
se = strel('disk', r_min);
Ie = imerode(im_proc, se);
Iobr = imreconstruct(Ie, im_proc);

Cierre por reconstrucción
se2 = strel('disk', r_min-10);
Iobrd = imdilate(Iobr, se2);
Iobrcbr = imreconstruct(imcomplement(Iobrd),
imcomplement(Iobr));
Iobrcbr = imcomplement(Iobrcbr);

Suavizado de bordes
se3=strel('disk', 4);
Iobrcbr = imclose(Iobrcbr, se3);
Iobrcbr = imerode(Iobrcbr, se3);
```

En orden de ejecución, en primer lugar se realiza una pequeña limpieza de la imagen eliminando pequeños elementos oscuros que se encuentren en la imagen en escala de grises resultado de los métodos de análisis de imagen.

Posteriormente se emplea una apertura-cierre por reconstrucción para eliminar detalles menores que  $r_{min}$  y eliminar imperfecciones. El radio mínimo puede ser seleccionado por el usuario como se ha mencionado anteriormente, pero se ha elegido un valor de compromiso para la realización de las pruebas.

La elección de la apertura-cierre por reconstrucción y no la versión morfológica de la misma se debe a que en la versión por reconstrucción se conservan los bordes de los elementos mantenidos.

Por último se realiza un cierre y una erosión con elementos estructurantes muy pequeños para redondear bordes y evitar irregularidades.

---

### Segmentación: *watershed* con marcadores

Tras la limpieza de la imagen y dejando, en la medida de lo posible, solamente visibles los elementos a segmentar, se puede comenzar el proceso de segmentación. En primer lugar, se han probado métodos basados en la binarización de la imagen como el de *Otsu* [23], el cual elige el mejor umbral de binarización suponiendo un histograma bimodal. Sin embargo, la selección del umbral supone una dificultad añadida dependiendo de la imagen ya que no hay iluminación uniforme. Además, la binarización no resuelve el problema de la segmentación de células superpuestas.

Dado que el resultado de las dos fases de preprocesado devuelve imágenes en escala de grises y no se desea perder información, se optó por aplicar el método *watershed* para el tipo de imágenes mencionado, es decir, el método de *watershed* con marcadores [23]. El método *watershed* considera los valores de intensidad de la imagen como un relieve topográfico. En otras palabras, consiste en la “inundación” de la imagen partiendo desde los puntos mínimos como si fueran el fondo de cuencas vacías (*catchment basins* en inglés) y los bordes de dichas cuencas fueran los máximos relativos. Cuando dos de esas supuestas cuencas vacías, siendo llenadas por “fuentes” distintas, desbordan y se unen, se crea una línea divisoria entre ellas, denominada *línea watershed*, tal y como se muestra en la Figura 4.14.

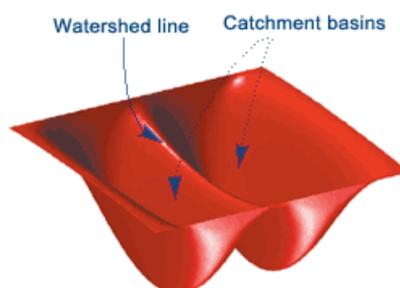


Figura 4.14: Representación de los mínimos y máximos en la aplicación del método *Watershed* tomada de [41].

Con esto en cuenta, la aplicación de marcadores consiste en establecer manualmente los mínimos de la imagen para definir por dónde comenzará a llenarse la misma. Siendo más precisos y siguiendo con el ejemplo de las cuencas, los marcadores suponen localizar y dar forma a las cuencas.

En concreto, se definen dos tipos de marcadores: externos e internos. La búsqueda de los marcadores internos consistirá en localizar puntos lo más interiores posible del objeto a segmentar, en este caso, las células. Posteriormente, se impondrán dichos puntos como mínimos de la imagen. La búsqueda de los marcadores externos sigue el

mismo patrón pero en este caso se pretende encontrar una región que rodee al elemento a segmentar y después imponerla como un mínimo.

Una vez obtenidos los dos tipos de marcadores, se realiza la *watershed*, no sobre la imagen original, sino sobre la imagen gradiente. Esto es así porque el gradiente de una imagen devuelve como resultado píxeles claros (niveles de intensidad altos) en los bordes de los objetos y píxeles oscuros (niveles de intensidad bajos) en el resto de la imagen, por lo que se acota el área a segmentar.

La aplicación del método y la búsqueda de los marcadores se ha implementado tomando como referencia métodos de [39], [23] y [40], siendo este último un método que emplea la *Fast Radial Symmetry Transform* [42].

#### ■ Marcadores internos

Para la búsqueda de los **marcadores internos** se ha empleado la *Fast Radial Symmetry Transform*, o FRST, con el código adaptado por Gareth Loy [43]. Éste método detecta aquellos puntos que tienen una alta simetría radial a través del gradiente. Concretamente, a partir de una serie de radios de los elementos de interés, para cada radio se forman dos imágenes proyección a través del gradiente: una del gradiente magnitud y otra de la orientación de los objetos. Estas proyecciones se van completando con las contribuciones de los distintos píxeles a la simetría radial, obteniendo como resultado una imagen donde se observan píxeles oscuros en los objetos más brillantes con simetría radial y píxeles claros en los menos brillantes.

Dado que las células son objetos normalmente con forma redondeada, se ha escogido este método para la creación de la máscara de marcadores internos. A continuación se detallará su aplicación y la elección de parámetros.

```
radii = r_min:2:r_max;
So = FRST(Iobrcbr, radii, 0.05, 1, 1);
So = uint8(255*mat2gray(So));
Somin = imextendedmax(So, 1);
```

La función *FRST* tiene como parámetros de entrada la imagen preprocesada, un vector de radios, un valor de umbral, un parámetro de restricción de simetría (desde simetría bilateral hasta estrictamente simetría radial), y el modo de funcionamiento (gradiente magnitud u orientación).

Para la aplicación se ha elegido un rango de radios desde  $r_{min}$  a  $r_{max}$  como se define en [42], sin incluir todos los elementos del rango. Además, se ha mantenido

---

el umbral por defecto de la función, se ha elegido una restricción de simetría intermedia y, por último, se ha elegido el modo en que sólo se utiliza la orientación como información para componer el resultado como se describe en [40], ya que puede mejorar el resultado en casos de bajo contraste entre células y fondo.

La imagen obtenida se escala entre 0 y 255 y, dado que los píxeles de interés están marcados con valores altos, se utiliza la función  $imextendedmax(A,x)$  para binarizarla. Esta función elimina los máximos cuyo valor estén una cantidad  $x$  por debajo de los máximos que le rodean, por lo que siendo estrictos con la variable  $x$  se consiguen pequeños puntos en el centro de los objetos detectados como se muestra en la Figura 4.15.

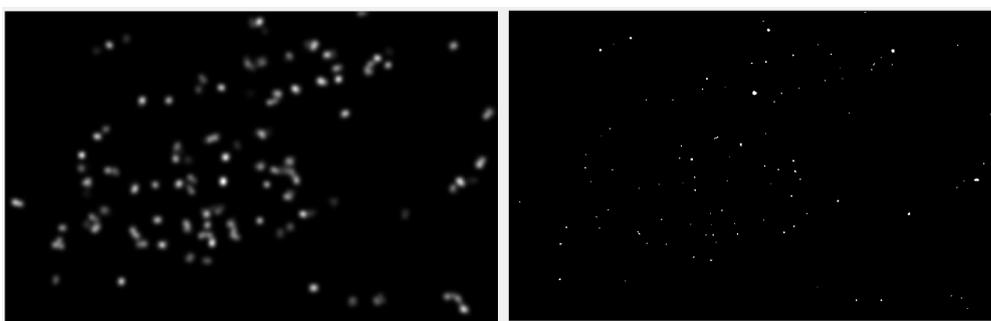


Figura 4.15: A la izquierda, resultado de aplicar la FRST y a la derecha la máscara de marcadores internos tras aplicar  $imextendedmax(A,1)$ .

- **Marcadores externos**

En el caso de los **marcadores externos**, se ha optado por utilizar la máscara de marcadores internos, que contiene la unidad más pequeña en que pueden representarse los objetos detectados, aplicándole la función *watershed* como se indica en [39].

La transformada distancia (función *bwdist* en MATLAB) de una imagen binaria devuelve una imagen donde se representa la distancia entre cada píxel de valor 1 y el más cercano de valor distinto de 1. Esto es útil para células que están solapadas, puesto que la imagen representa ese solapamiento de forma que sea más perceptible la existencia de más de un objeto, como se puede observar en la Figura 4.16.

La máscara de marcadores externos se obtiene por tanto de la siguiente forma:

```
D = ~bwdist(~Somin);
DL = watershed(D);
em = DL == 0;
```

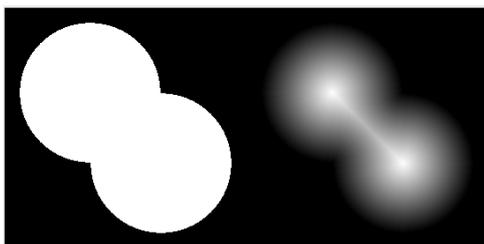


Figura 4.16: A la izquierda, imagen original. A la derecha, aplicación de *bwdist* sobre la imagen complementaria de la original. Ejemplo tomado de la documentación de la función en MATLAB.

#### ■ Segmentación

Como se mencionó anteriormente, la segmentación con marcadores se realiza sobre la imagen gradiente, por lo que los últimos pasos tras obtener ambos marcadores serían los siguientes:

```
hy = fspecial('sobel');
hx = hy';
Iy = imfilter(double(Iobrcbr), hy, 'replicate');
Ix = imfilter(double(Iobrcbr), hx, 'replicate');
gradmag = sqrt(Ix.^2 + Iy.^2);
g2 = imimposemin(gradmag, Somin|em);
L2 = watershed(g2);
```

La función *imimposemin(A,B)* impone los mínimos de B sobre la imagen A, por lo que la imagen resultante es apta para realizar la segmentación *watershed*.

#### Postprocesado

La segmentación *watershed* devuelve como resultado una imagen de etiquetas, donde cada objeto detectado está etiquetado con un número distinto. Para afinar un poco más la detección de células, se realiza un pequeño postprocesado que elimine falsos positivos, como objetos demasiado grandes para ser considerados células u objetos con forma demasiado irregular.

```
s = regionprops(L2, 'Area', 'Solidity');
Segmen_area = [s.Area];
Solidity = [s.Solidity];
allowableAreaIndexes = (Segmen_area <= pi * radiomax^2) & ...
(Solidity > 0.6);
keeperIndexes = find(allowableAreaIndexes);
```

```
keeperBlobsImage = ismember(L2, keeperIndexes);  
L2 = bwlabel(keeperBlobsImage, 8);
```

La función *regionprops* obtiene las características especificadas de una imagen de etiquetas como la resultante de la segmentación *watershed*. Dado que desde el principio se han acotado las dimensiones de los objetos con un radio máximo, la primera de las condiciones de cribado será que los objetos segmentados sean menores que el área de un círculo de radio máximo. La segunda de las condiciones será que las regiones sean algo convexas, ya que las células, por su forma, deberían serlo. Para ello, se utiliza la propiedad *Solidity*, que es la división  $\text{Área}/\text{Área región convexa}$ , donde el valor 1 sería el resultado de una región totalmente convexa.

Para dar una visión global del proceso de análisis de imagen, en la Figura 4.17 se representa el ciclo de trabajo que se realiza.

En último lugar cabe mencionar que se ha intentado adaptar a MATLAB el reciente algoritmo propuesto en el artículo *STAIN DECONVOLUTION OF HISTOLOGY IMAGES VIA INDEPENDENT COMPONENT ANALYSIS IN THE WAVELET DOMAIN* [2], pero la publicación deja algunas dudas sin resolver y la implementación se ha dejado a la espera de la publicación del desarrollo detallado que, según palabras de uno de los autores (con el cual se estableció contacto vía e-mail durante la realización de este TFG), están ultimando para su publicación. Éste método desarrolla una nueva aplicación del ICA, llamada *Subband Independent Component Analysis* o SBICA, donde se puede trabajar con señales dependientes buscando la independencia de su descomposición en componentes de banda estrecha, utilizando transformadas *Wavelet* en las distintas componentes de color.

---

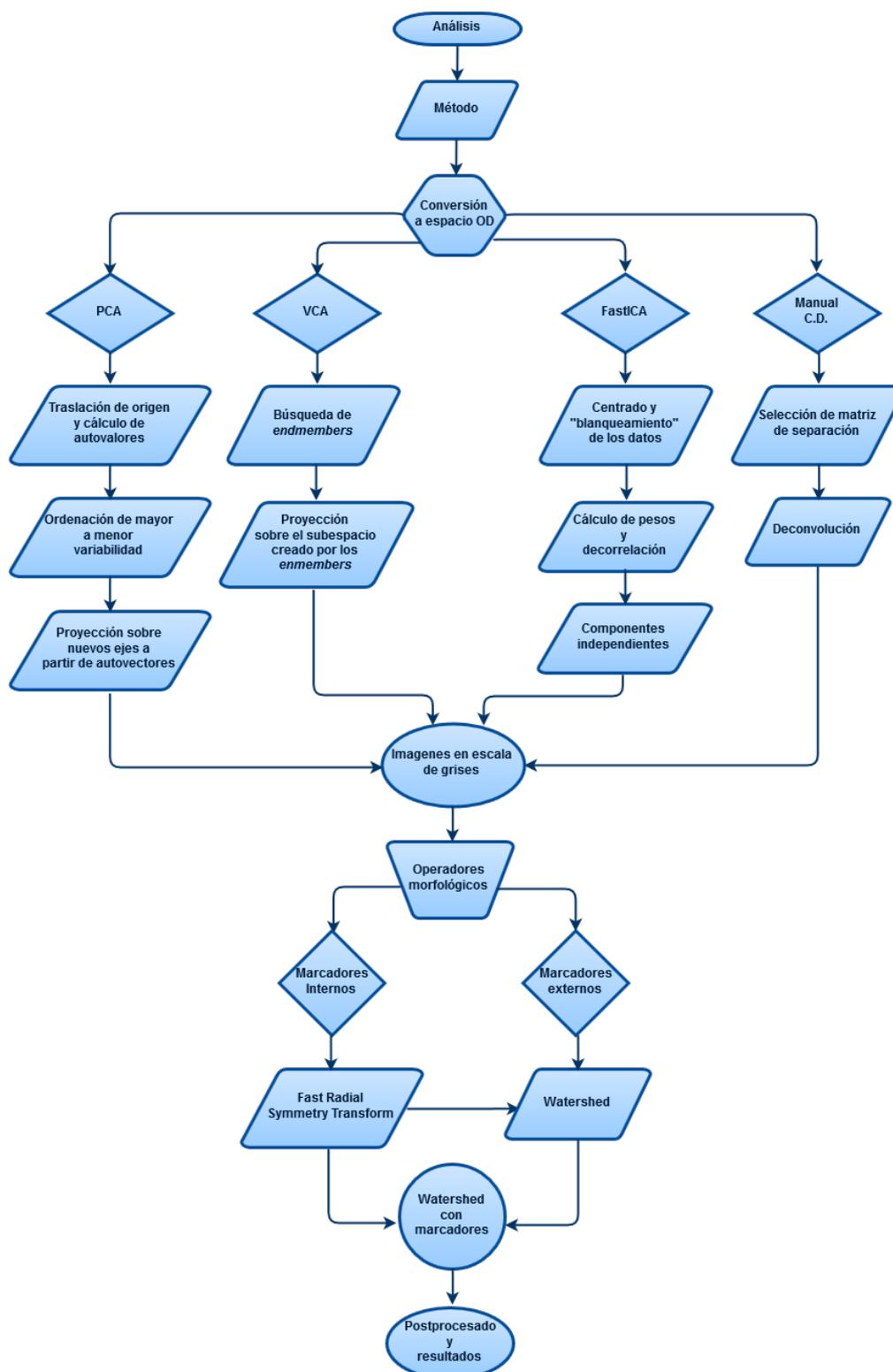


Figura 4.17: Flujograma de funcionamiento del análisis de imagen MATLAB.

#### 4.5.4. Conteo manual y máscara de comprobación

Como último paso en el desarrollo del trabajo, se tomaron imágenes características de zonas relevantes de las muestras disponibles con ayuda de la interfaz. Existen dos tipos de imágenes: las obtenidas de las muestras proporcionadas por la patóloga del Hospital Clínico, con un aumento de  $20\times$ , y las imágenes de la muestra de *Ki-67* del fabricante Ventana con un aumento de  $40\times$  disponible en la página de *Openslide* [15]. Con ellas se realizó una segmentación manual rodeando las células consideradas malignas en rojo y las células azules en verde y se creó una máscara de comprobación para poder comparar los métodos programados con el método manual, como se observa en la Figura 4.18.

Para la segmentación manual se dispuso de la ayuda de una tableta digitalizadora que agilizase el proceso de marcación de las células. Los colores elegidos han sido el rojo puro  $\{255,0,0\}$  y el verde puro  $\{0,255,0\}$ , con lo que la máscara se extrae binarizando la imagen segmentada separando por un lado los elementos cuyos valores de verde y rojo sean puros. Este proceso, como se indicó en el apartado 2.5, es un proceso largo y tedioso que además requiere tomar decisiones sobre qué células se consideran como marrones o no. Dicha complicación radica en que la aplicación del tinte no es perfecta y hay células que quedan marcadas como azules con pequeñas pinceladas de marrón, por lo que quedan a elección de quien realiza la segmentación. Esta problemática está totalmente alineada con las dificultades de interpretación que tiene el patólogo a la hora de emitir un diagnóstico.

Los datos recogidos de la segmentación manual se expondrán junto con los obtenidos en el análisis durante el Capítulo 6.

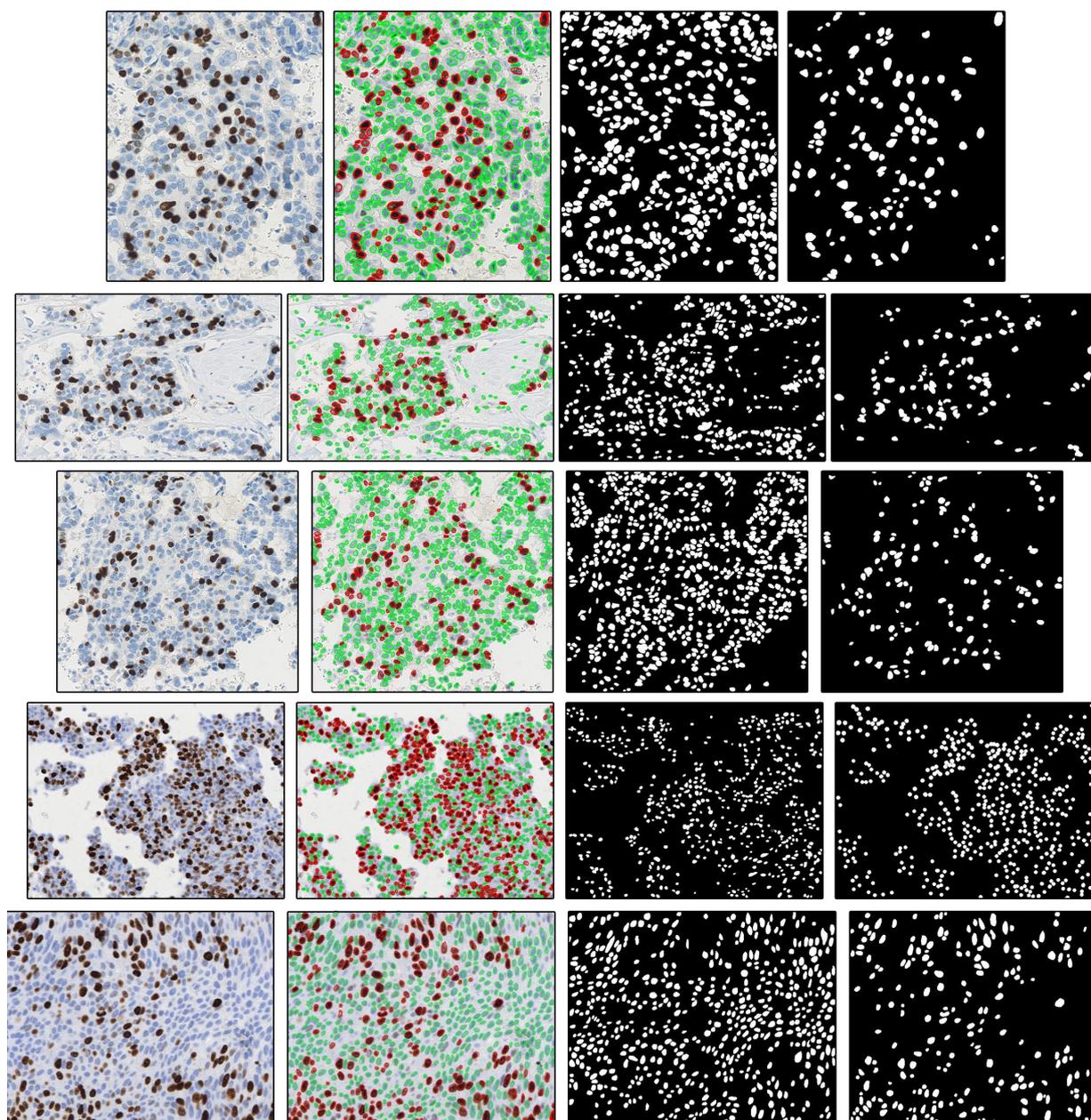


Figura 4.18: De izquierda a derecha: Muestra original > Segmentación manual > Máscara de segmentación de células azules > Máscara de segmentación de células marrones.

# Capítulo 5

## Resultados

### Índice de contenidos

---

<b>5.1. Introducción</b> . . . . .	<b>49</b>
<b>5.2. Métodos de separación de componentes de color</b> . . . . .	<b>49</b>
5.2.1. PCA . . . . .	50
5.2.2. VCA . . . . .	51
5.2.3. ICA . . . . .	53
5.2.4. Deconvolución de color manual . . . . .	55
<b>5.3. Procesado morfológico</b> . . . . .	<b>57</b>
<b>5.4. Segmentación</b> . . . . .	<b>60</b>
5.4.1. Deconvolución manual . . . . .	60
5.4.2. Método VCA . . . . .	63
5.4.3. Contabilización . . . . .	65

---

## 5.1. Introducción

A lo largo de este capítulo se presentarán y analizarán todos los resultados gráficos obtenidos de las distintas partes que componen el análisis de imagen. Posteriormente, en el Capítulo 6, se puntualizarán las conclusiones más importantes extraídas del estudio.

Para organizar las imágenes, los resultados se dividen en los siguientes apartados:

- **Métodos de separación de componentes de color.** Resultados obtenidos en la aplicación de los métodos PCA, VCA, ICA y deconvolución de color manual, descritos en el Capítulo 4.5. Las muestras disponibles se dividen en dos grupos: resolución a  $40\times$  (muestras 1, 2 y 3) y  $20\times$  (muestras 4 y 5).
- **Procesado morfológico.** Una vez elegidos los métodos donde se aprecia una separación de componentes efectiva, se presentan los resultados obtenidos con el procesado descrito en la Sección 4.5.3, el cual busca limpiar la imagen de objetos indeseados y resaltar más los resultados del punto anterior.
- **Segmentación.** Para finalizar el análisis de imagen, se muestran las imágenes obtenidas en la fase de segmentación descrita en la Sección 4.5.3. En concreto, se han recogido los resultados de los marcadores internos y externos, la máscara de segmentación final y esta última superpuesta con la imagen original.
- **Índices de evaluación.** Tras el análisis, se presentarán los resultados numéricos obtenidos en cuanto a número de células detectadas de cada uno de los tipos. Además, se introducirán algunos de los índices de evaluación de la segmentación utilizados en análisis de imágenes médicas [45] que permitirán comparar de una forma objetiva los resultados entre métodos.

## 5.2. Métodos de separación de componentes de color

El primer paso del análisis de imagen lo compone la selección de un método que permita separar las componentes de la imagen de forma que sean diferenciables. Estos métodos reciben como entrada la transformación de la imagen original del espacio RGB al espacio OD descrito en la teoría de la Sección 4.5.2, ya que en este espacio las componentes de las combinaciones de los tintes se consideran lineales debido a la Ley de Lambert-Beer [44].

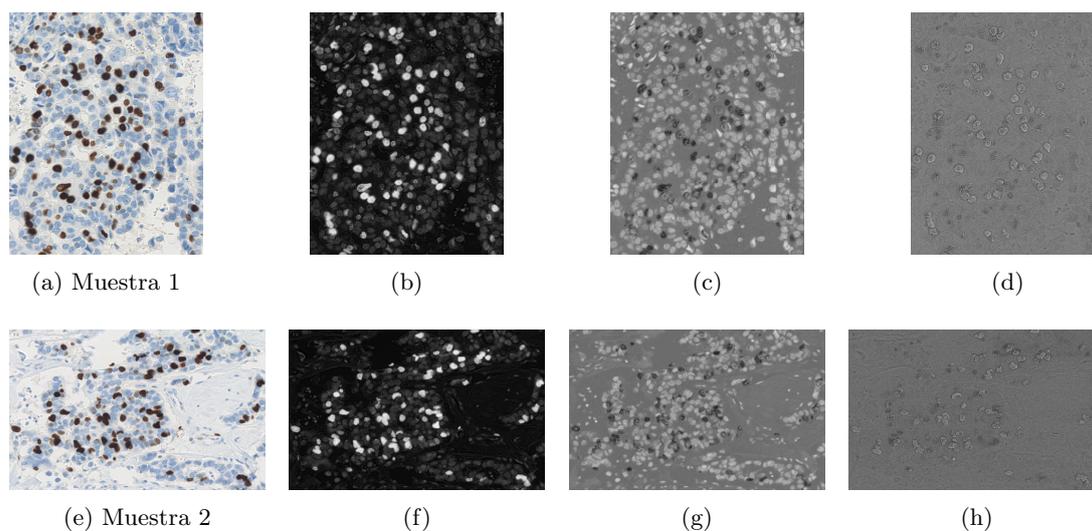
---

### 5.2.1. PCA

En la Figura 5.1 se muestran los resultados para las muestras con el método PCA, apareciendo de izquierda a derecha la imagen original, la componente para las células marrones, la componente para las células azules y una tercera componente del fondo. Como se observa en dichos resultados, la reducción de dimensionalidad del método PCA no aporta una separación clara de las distintas componentes.

En el caso de la primera componente de las imágenes de resultados de la segunda columna de la Figura 5.1, las células marrones quedan más definidas en niveles de intensidad altos que el resto de células (que se siguen observando) por lo que es posible que con un tratamiento de imagen posterior distinto al empleado puedan obtenerse resultados aceptables. Para el caso de las azules (tercera columna), es más complicado puesto que, aunque a simple vista se distinguen de las marrones, estas últimas no tienen niveles de intensidad uniformes, por lo que al hacer la imagen complementaria siguen quedando restos que se confunden con las células azules. La poca precisión del método puede ser debida a que la medida de la variabilidad, objetivo del método PCA, en este tipo de imágenes no es una característica suficientemente importante para diferenciar entre dos tipos de tintes, ya que los colores no son uniformes ni únicos dentro de una misma célula.

En vista de estos resultados, el método podría ser viable realizando un procesamiento de imagen posterior más elaborado y distinto al establecido para este trabajo por lo que no se continuará con el estudio de los resultados de PCA.



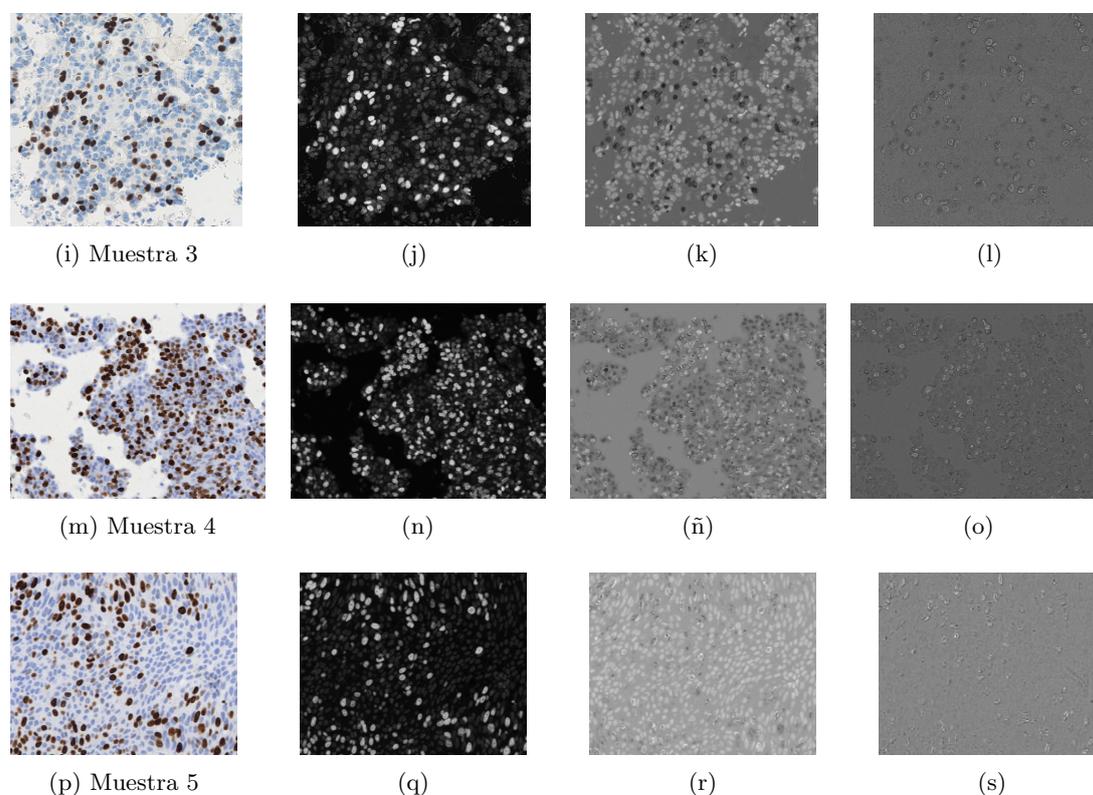
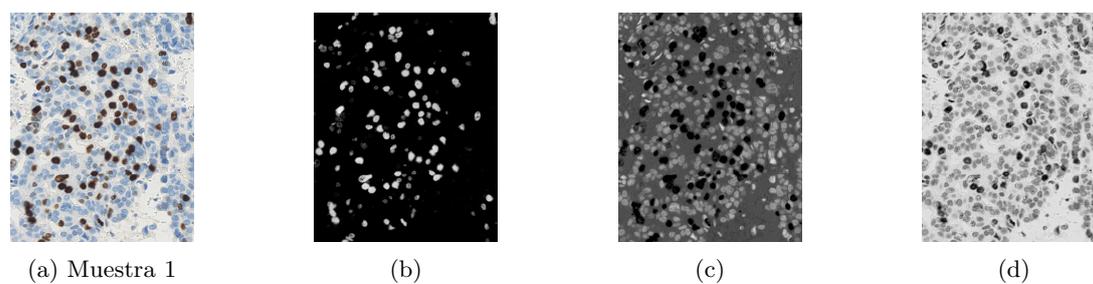


Figura 5.1: Resultados obtenidos con el método PCA, apareciendo de izquierda a derecha la imagen original, la componente para las células marrones, la componente para las células azules y una tercera componente del fondo.

### 5.2.2. VCA

En este método es necesario diferenciar entre los resultados obtenidos para las muestras de  $40\times$  y las de  $20\times$ , que se muestran en la Figura 5.2, donde la primera columna corresponde a la imagen original, la segunda a la separación para las células marrones, la tercera a la separación de células azules y la última a una componente no utilizable.



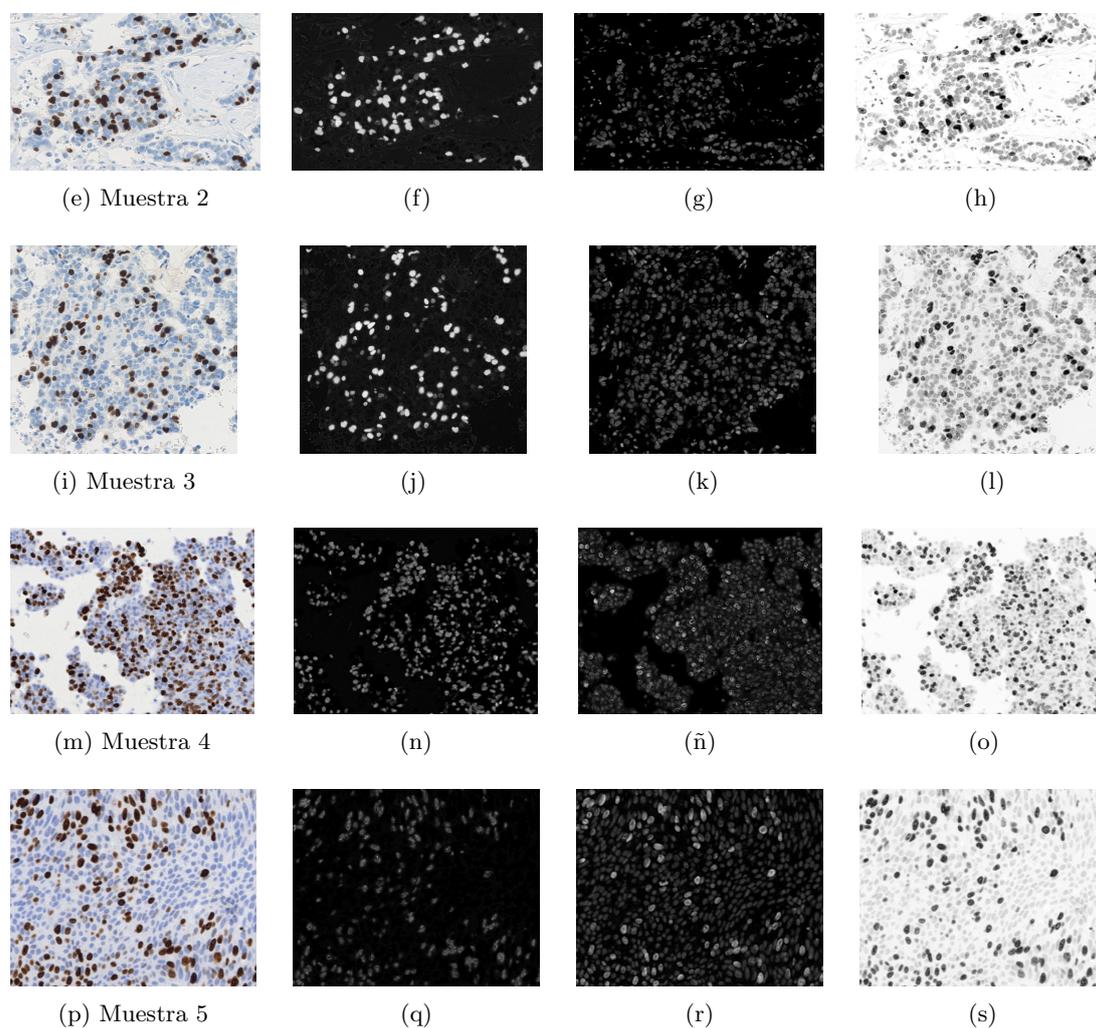


Figura 5.2: Resultados obtenidos con el método VCA. De izquierda a derecha, imagen original > componentes marrones > componente de azules > componente inutilizable.

En las primeras muestras ( $40\times$ ) generalmente se observa una separación eficaz de las células de color marrón en alguna de las componentes y, en ocasiones, de las azules. Como comparativa con el segundo tipo de muestras ( $20\times$ ), este último genera resultados más difíciles de valorar puesto que se pierden gran cantidad de detalles realizando una peor separación. En estas últimas muestras, las componentes azules directamente no pueden separarse.

Como se aprecia en las imágenes  $f$  y  $q$  de la Figura 5.2, la separación en  $40\times$  es más uniforme y clara que en las del segundo grupo. Aún así, en algunos casos aparecen buenos resultados para las células marrones, por lo que se mostrarán más adelante.

Uno de los problemas del método es que la estimación inicial de los *endmembers* se realiza de manera aleatoria, por lo que el resultado no es siempre estable y dependiendo de la inicialización no siempre se encuentran los mismos *endmembers* ni en el mismo orden. Es por eso que en ocasiones se observan distintos resultados para una misma muestra en las mismas condiciones o incluso varía el orden en que se muestran.

Un ejemplo de esto se observa en la Figura 5.3, donde a la izquierda se muestra la imagen complementaria de una de las componentes obtenidas en una ejecución del método. En ella se observa una buena separación de las células azules, ya que las marrones no aparecen. Sin embargo, en la imagen de la derecha, que también es la imagen complementaria del resultado para ayudar a la visualización, se observan todas las células sin especial distinción ya que también aparecen las marrones.

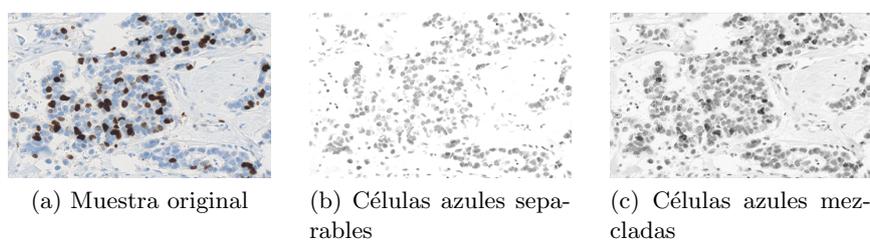


Figura 5.3: Comparación de la misma muestra en distintas ejecuciones.

### 5.2.3. ICA

Como se observa en los resultados de la Figura 5.4, al igual que en VCA, el orden de las componentes y los resultados también varían. La diferencia es que en este caso ninguna de las componentes realiza una separación clara de las células. Esto puede ser debido a que las componentes de color no son totalmente independientes, ya que el azul no es un azul puro y el marrón es una mezcla de otros colores. En la representación, se ordenan de forma que la segunda columna muestre la componente donde las células marrones se ven más oscuras respecto al resto y en la segunda columna se muestre la componente donde las azules sean visibles y se distingan de las marrones, aunque no parezcan separables.

En base a la imposibilidad de separar fácilmente los objetos con los métodos propuestos en este trabajo, se descarta continuar el desarrollo de los resultados con el método ICA.

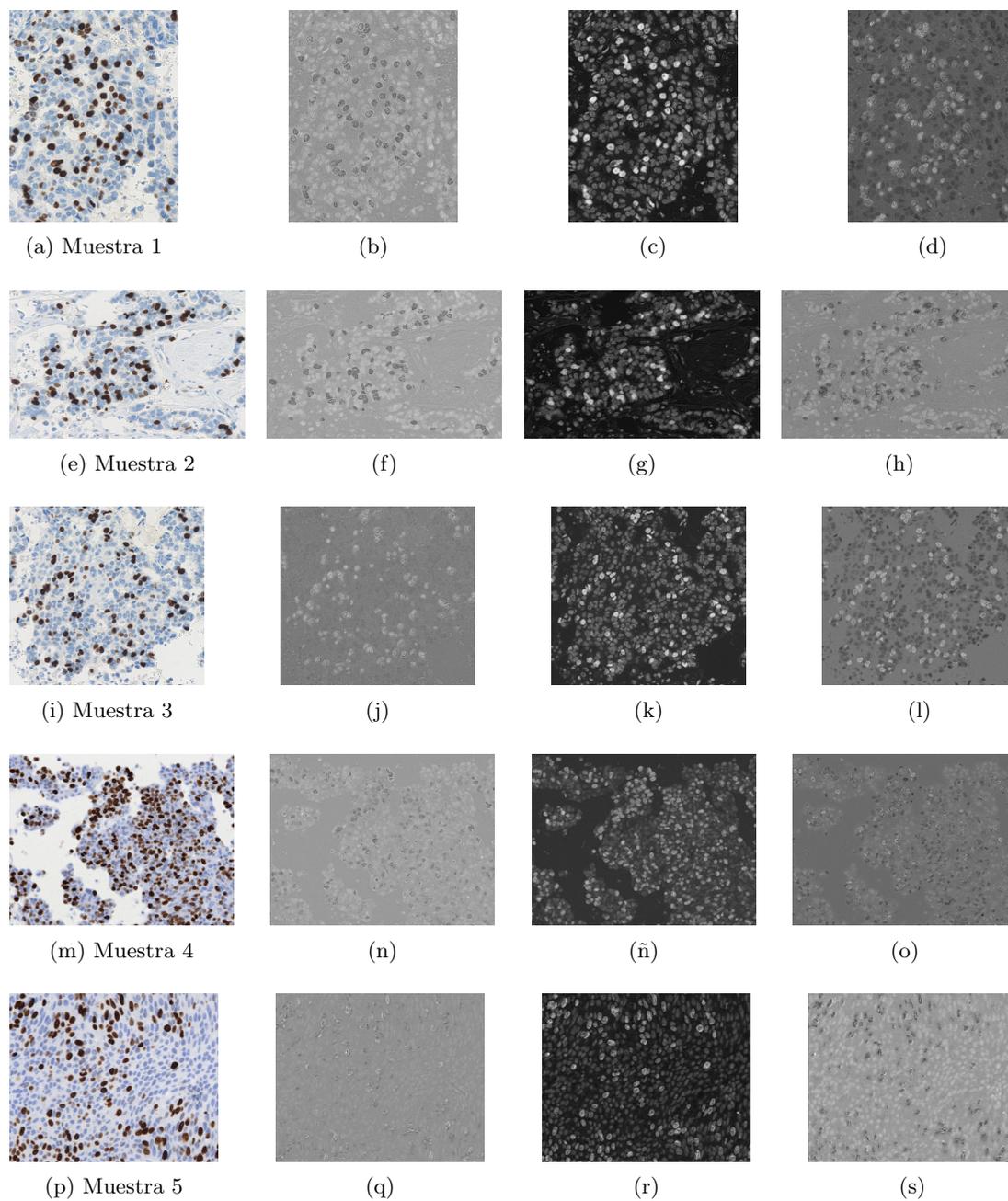
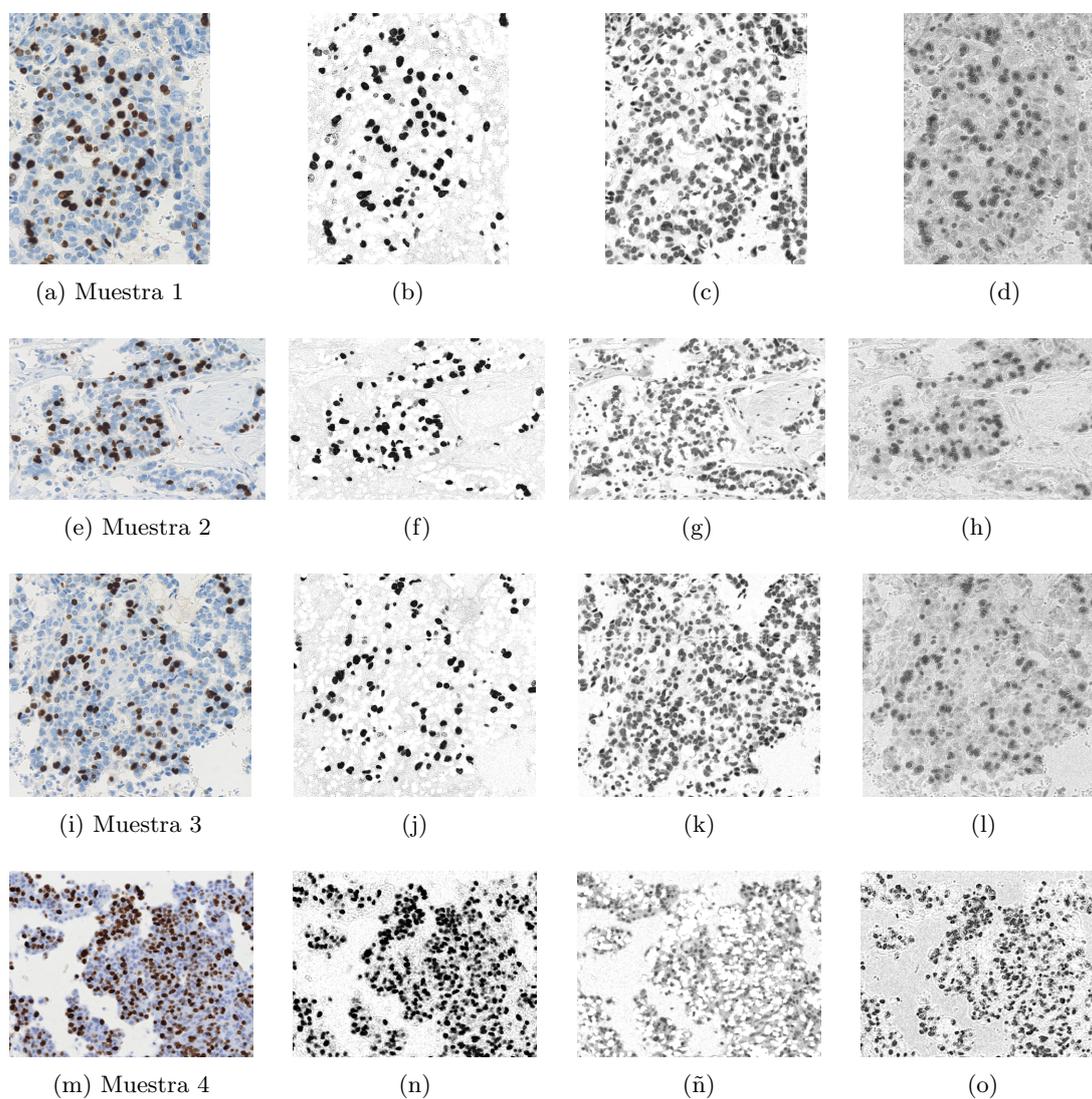


Figura 5.4: Resultados obtenidos con el método ICA. De izquierda a derecha, imagen original > células marrones > células azules > componente inservible.

### 5.2.4. Deconvolución de color manual

En la Figura 5.5 se presentan los resultados proporcionados por la deconvolución de color manual, apareciendo de izquierda a derecha la imagen original, la separación de componentes marrones, la separación de componentes azules y el residuo. Como se observa en las imágenes, a pesar de haber realizado la separación de tintes con unos coeficientes obtenidos a mano y no totalmente optimizados para el tinte Ki-67, los resultados son muy buenos.



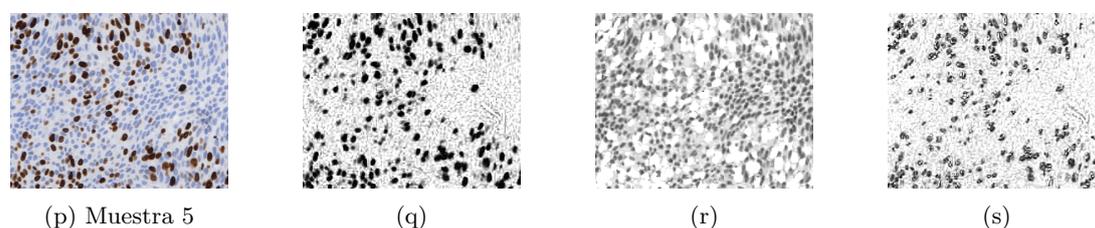


Figura 5.5: Resultados obtenidos con el método de deconvolución manual. De izquierda a derecha, imagen original > deconvolución de células marrones > deconvolución de células azules > residuo.

A diferencia del método VCA, donde sólo es viable separar las células marrones, con la deconvolución de color manual quedan totalmente diferenciados los dos tipos de células debido a que se ha introducido una solución específica para el problema que se presenta. Podría decirse que la deconvolución de color manual es una versión específica del método VCA donde los *endmembers* se asignan a mano con la matriz de separación. En ese aspecto, aunque el método en sí es manual porque hay que introducirle los parámetros del tipo de tinte a estudiar, se puede automatizar ya que se dispone de las matrices de coeficientes para numerosos tipos de marcadores celulares. Hay que tener en cuenta que la introducción de los tintes de forma manual no es trivial puesto que aunque se calibre con unos parámetros concretos hay una gran influencia de la composición de los tintes y la configuración del microscopio. Si alguna de esos parámetros varía significativamente, la calibración manual necesitaría ser recalculada para esas nuevas condiciones.

Es importante mencionar que en el grupo de imágenes de  $20\times$  aparecen ciertos artefactos en algunas de las componentes, como se observa en la Figura 5.6 para el método PCA y para la deconvolución de color manual, donde se produce con especial énfasis. Para el resto de métodos los artefactos aparecen en menor medida, pero si se prueba a aplicar los algoritmos de análisis sobre las imágenes en RGB estas son casi irreconocibles en los resultados.

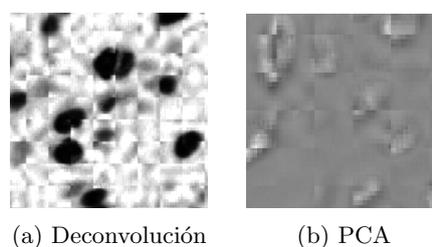


Figura 5.6: A la izquierda, artefactos producidos en la deconvolución de color manual. A la derecha, artefactos producidos en una de las componentes del método PCA.

Dado que las irregularidades no aparecen en las imágenes de  $40\times$  y todas han sido guardadas y tratadas de la misma forma a partir del archivo BIF, es coherente afirmar que la aparición puede ser debida a la formación de la imagen por parte del sistema WSI utilizado, como por ejemplo la compresión de la imágenes. Dicha compresión puede haber sido realizada de forma similar a JPG aproximando regiones de  $M\times N$  píxeles, lo cual distorsiona los colores en las transiciones entre estos elementos cuadrados de compresión por lo que dichas discontinuidades entre bloques son muy bruscas. Para confirmar esta hipótesis sería necesario obtener nuevas muestras modificando la configuración del microscopio.

### 5.3. Procesado morfológico

Tras la obtención de las imágenes en escala de grises es preciso realizar una limpieza de la imagen para mejorar los resultados obtenidos. Vistas las características de las imágenes obtenidas, los métodos candidatos a continuar el camino del preprocesado serían el VCA y la deconvolución manual. Concretando más, VCA sería viable para segmentar las células marrones ya que como se observa en la Figura 5.2, sólo una componente realiza una separación medianamente buena mientras que el resto de componentes no permite obtener una imagen donde se conserven las células azules por separado. A pesar de ello, se han tomado los resultados para los casos en que las células azules eran diferenciables salvo en las muestras 4 y 5 que no se ha conseguido una solución aceptable. En el caso de la deconvolución manual, se produce una separación bastante satisfactoria para las tres primeras muestras. Para las dos últimas, debido a la definición de las imágenes y a la aparición de artefactos, la calidad de la segmentación no es tan buena.

En las Figuras 5.7 y 5.8 se representa el procesado morfológico de las células marrones y azules de las imágenes obtenidas de los métodos de deconvolución manual y VCA en las Figuras 5.5 y 5.2. La limpieza de las imágenes se ha realizado de la forma explicada en el Apartado 4.5.3, realizando para el caso del método VCA la imagen complementaria de los resultados. Para las imágenes de  $40\times$  se ha establecido como radio mínimo 16 y el tamaño del elemento estructurante de la primera dilatación 4 mientras que para las de  $20\times$  el radio se define como 8 y el elemento estructurante como 2.

---

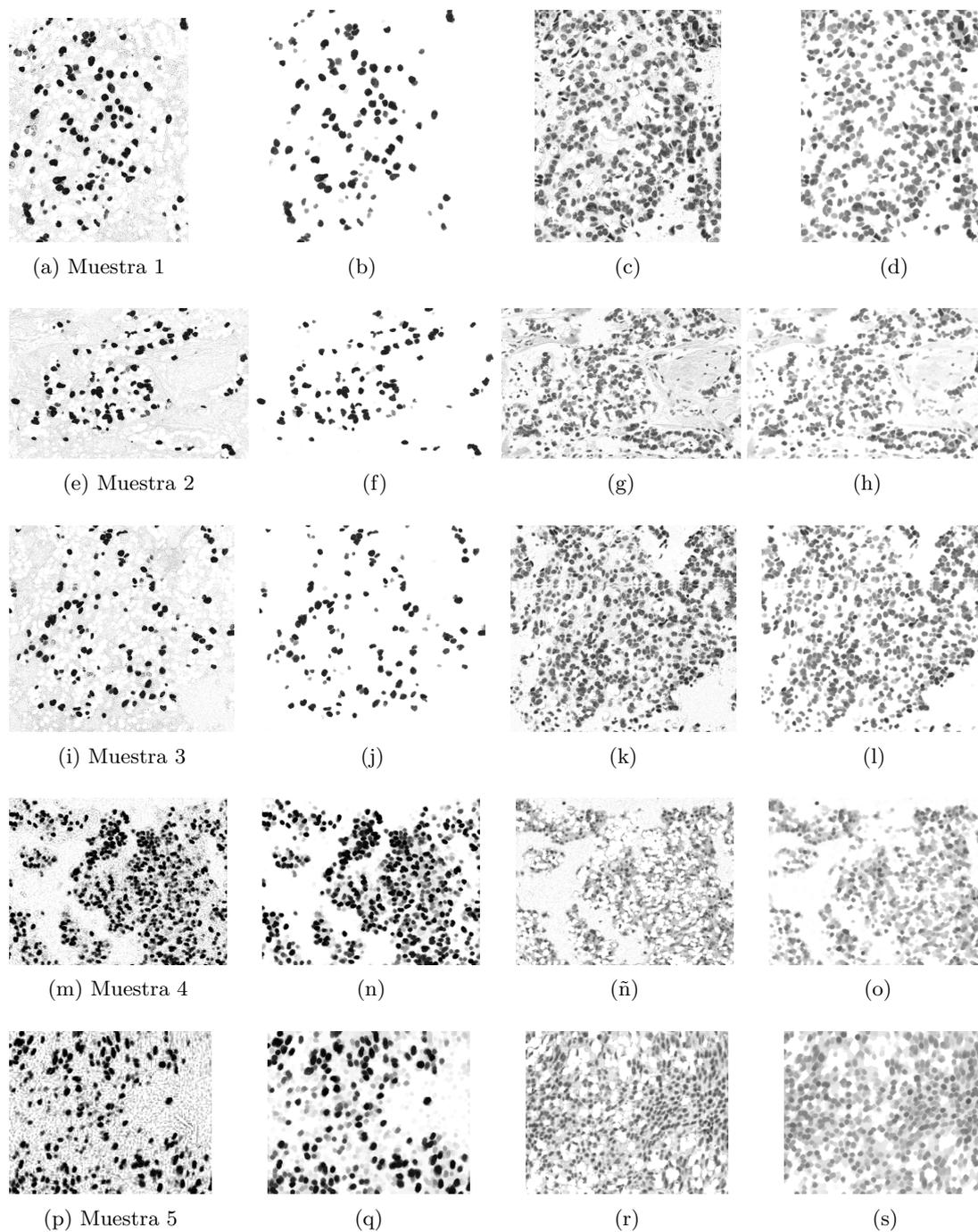


Figura 5.7: Resultados procesados del método de deconvolución manual. De izquierda a derecha, imagen de células marrones > imagen procesada de células marrones > imagen de células azules > imagen procesada de células azules.

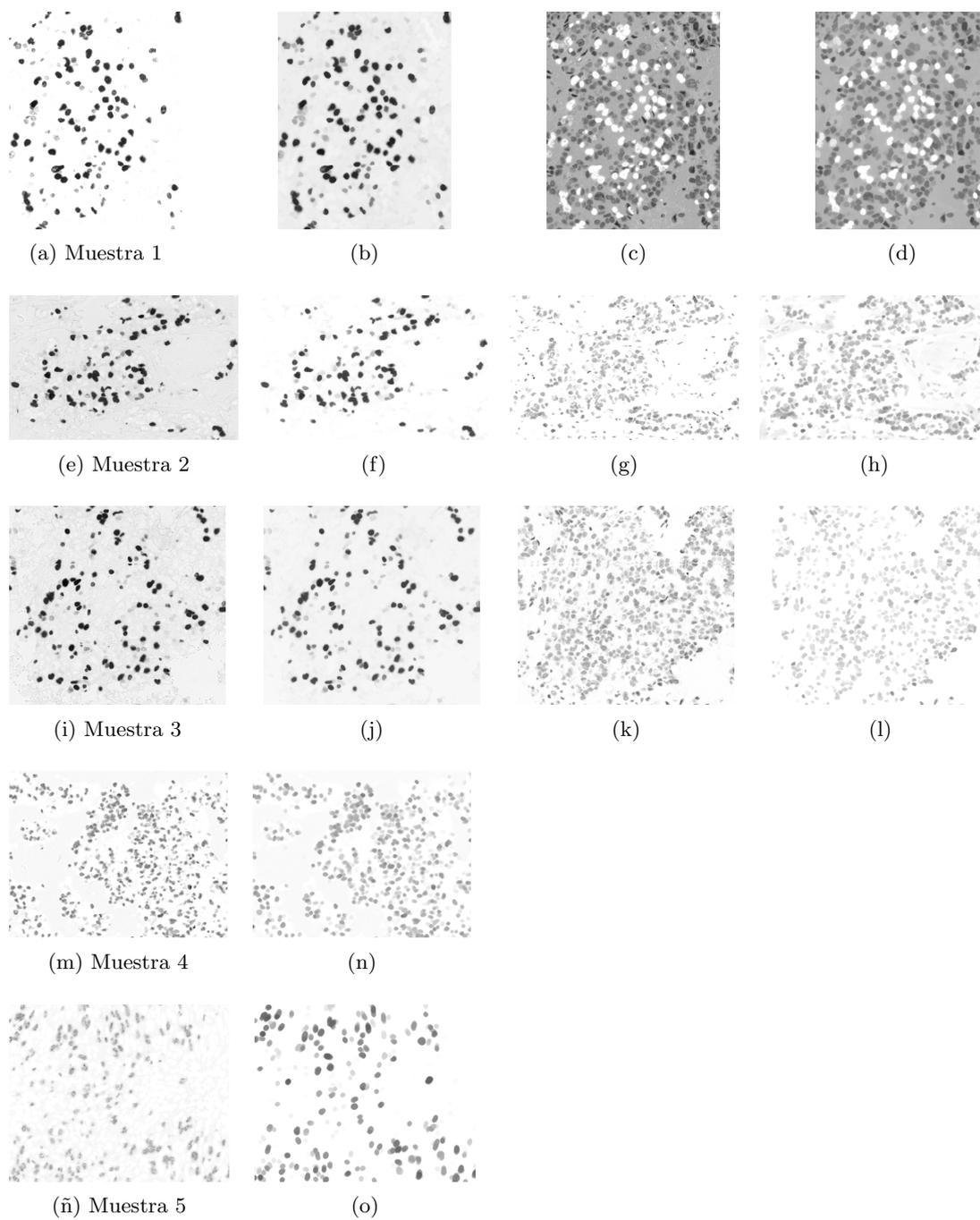


Figura 5.8: Resultados procesados de VCA. De izquierda a derecha, imagen complementaria de células marrones > imagen procesada de células marrones > imagen complementaria de células azules > imagen procesada de células azules.

## 5.4. Segmentación

### 5.4.1. Deconvolución manual

En el caso de la deconvolución, como se muestra en las Figuras 5.9, 5.10 y 5.11, se consiguen unos buenos resultados tanto para marrones como para azules. De izquierda a derecha, en la primera columna se observa la imagen original. En la segunda columna se observa la imagen original con los marcadores exteriores calculados con el método *watershed* superpuestos en verde y los marcadores internos calculados con la FRST en rojo. En la tercera columna aparece la segmentación *watershed* final, donde a cada objeto detectado se le aplica un color y en la última columna aparece la máscara *watershed* final superpuesta de nuevo sobre la imagen original.

A pesar de que los resultados son buenos, se hace difícil contabilizar con precisión las células que aparecen superpuestas entre sí ya que es complicado separarlas. El método podría mejorarse con la elección eficiente de la matriz de deconvolución y unos valores precisos de tamaños a la hora de realizar el preprocesado.

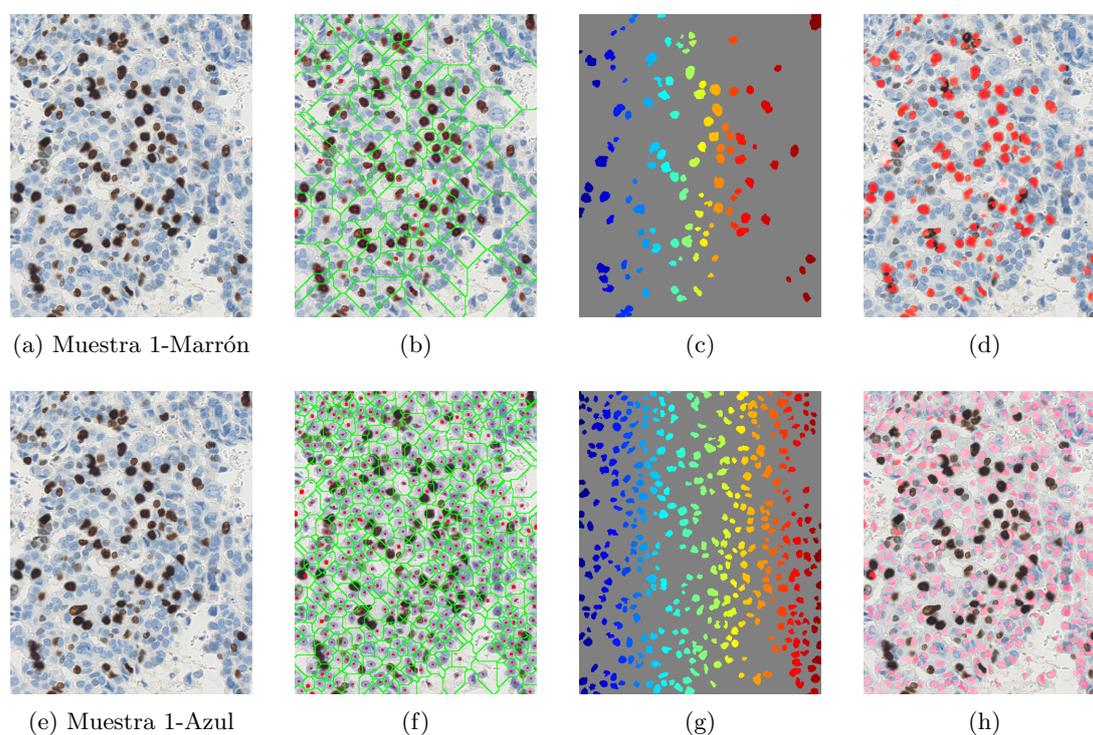


Figura 5.9: Resultados segmentación con el método de deconvolución manual de la muestra 1. De izquierda a derecha: original > marcadores > *watershed* > watershed sobre original.

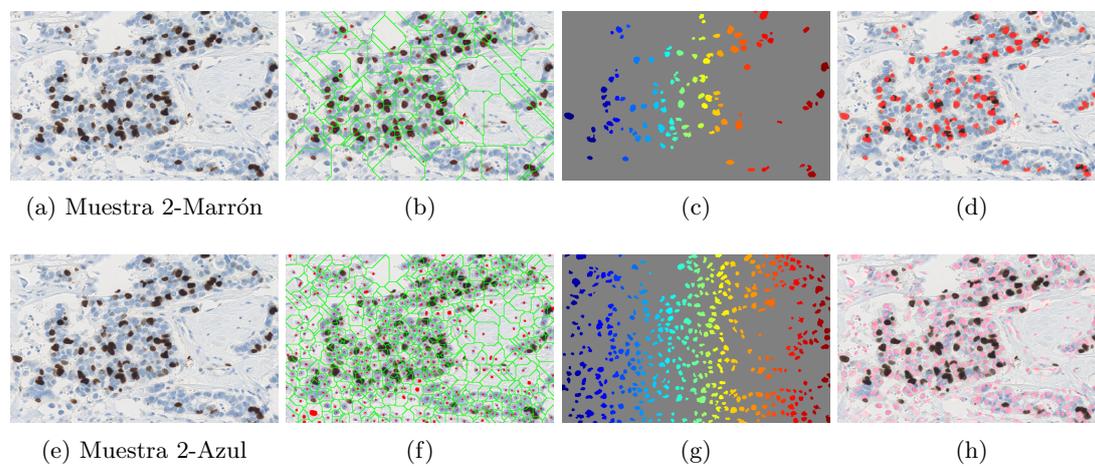


Figura 5.10: Resultados segmentación con el método de deconvolución manual de la muestra 2. De izquierda a derecha: original > marcadores > watershed > watershed sobre original.

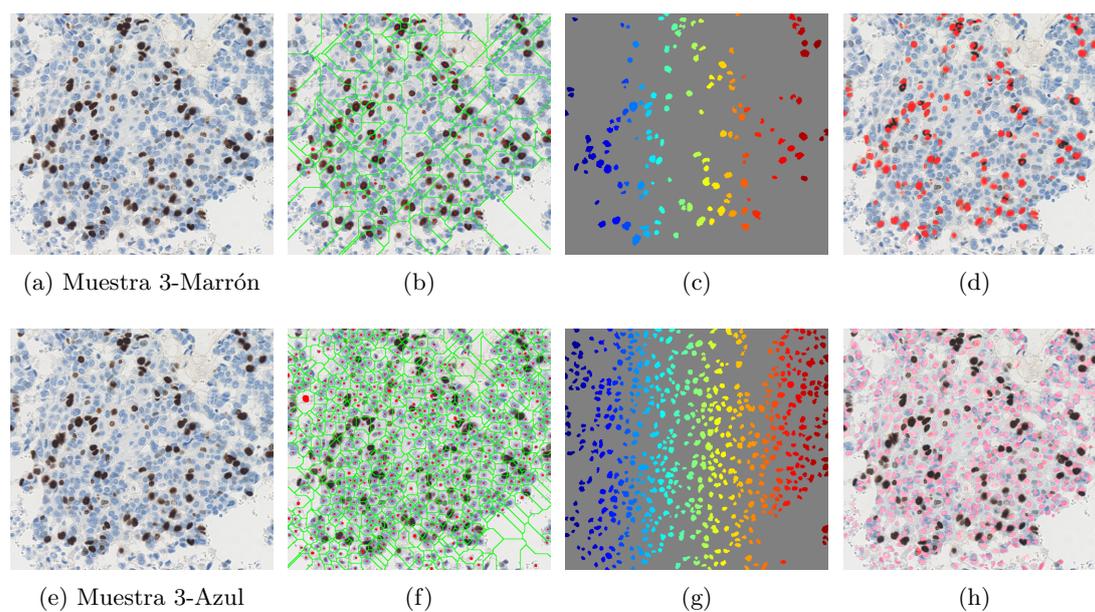


Figura 5.11: Resultados segmentación con el método de deconvolución manual de la muestra 3. De izquierda a derecha: original > marcadores > watershed > watershed sobre original.

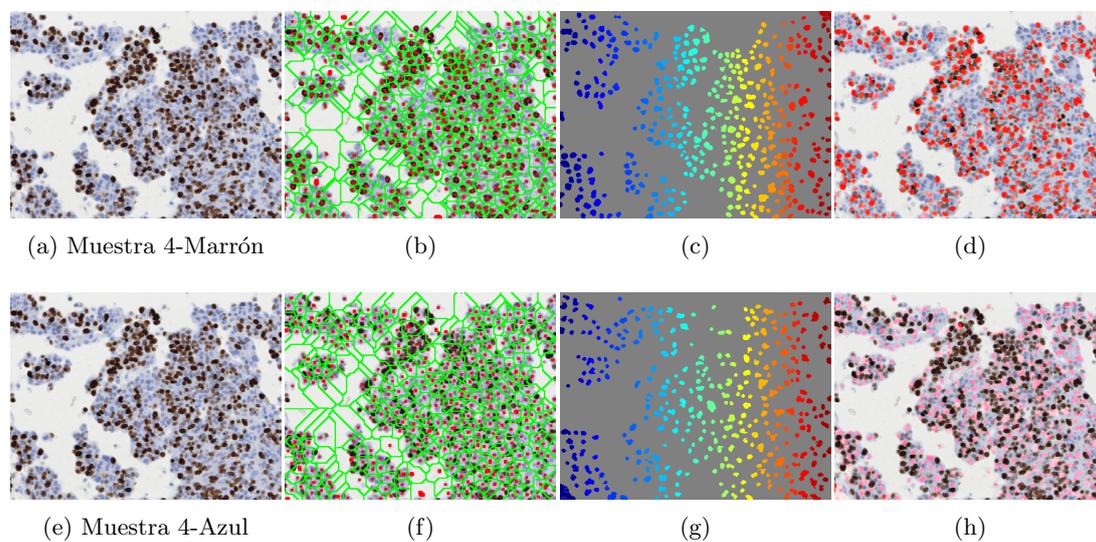


Figura 5.12: Resultados segmentación con el método de deconvolución manual de la muestra 4. De izquierda a derecha: original > marcadores > watershed > watershed sobre original.

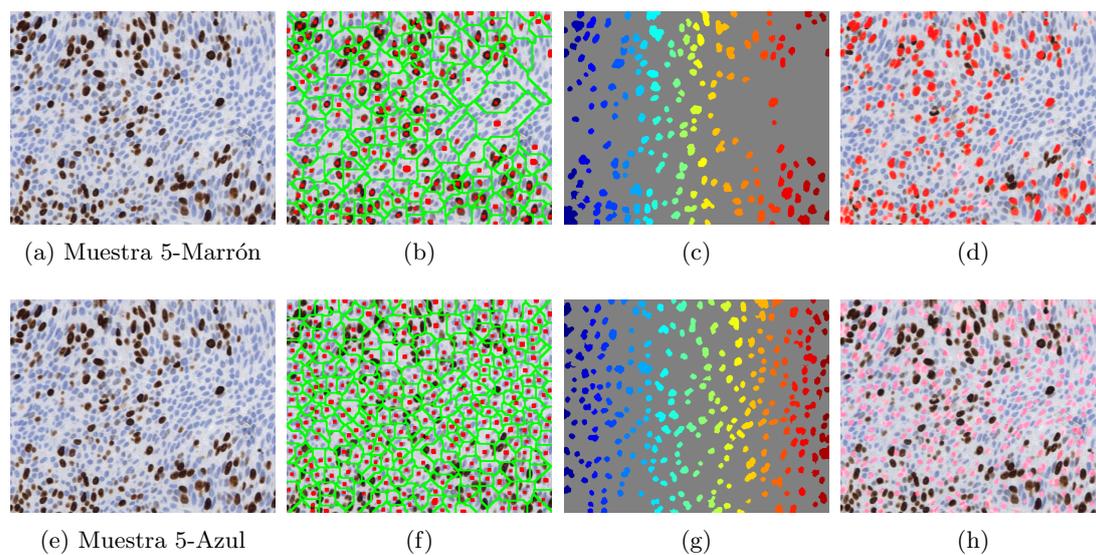


Figura 5.13: Resultados segmentación con el método de deconvolución manual de la muestra 5. De izquierda a derecha: original > marcadores > watershed > watershed sobre original.

### 5.4.2. Método VCA

El método VCA, como ya se ha mencionado anteriormente, trae varias problemáticas. La primera y más importante es su aleatoriedad. Dado que no hay *endmembers* característicos en la imagen, el algoritmo encuentra en cada ejecución unos *endmembers* distintos, lo que genera unos resultados diferentes. En ocasiones no es posible realizar la separación de las células azules y otras veces cambian completamente las imágenes aunque en su mayor parte siempre es posible segmentar con buenos resultados las marrones.

Para realizar la contabilización se han utilizado algunas de las mejores ejecuciones del código, en las que se visualizan más detecciones como se observa en las Figuras 5.14, 5.15, 5.16, 5.17 y 5.18, que siguen la misma estructura que las Figuras 5.9-5.13.

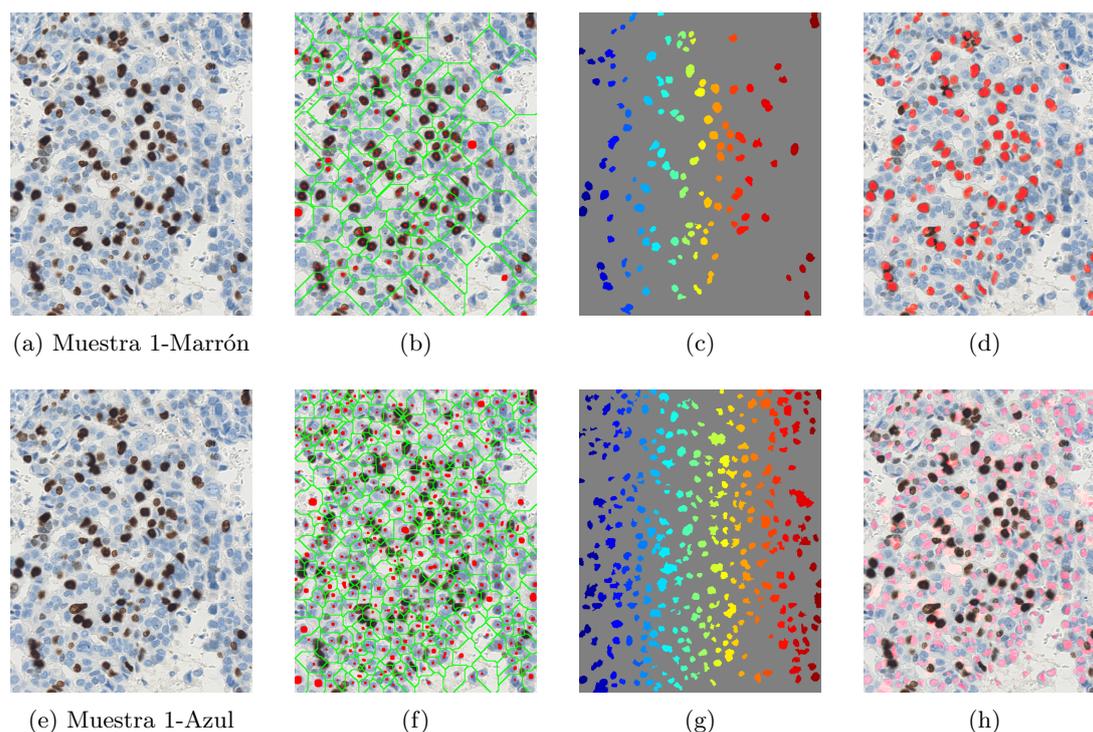


Figura 5.14: Resultados segmentación con el método VCA de la muestra 1. De izquierda a derecha: original > marcadores > watershed > watershed sobre original.

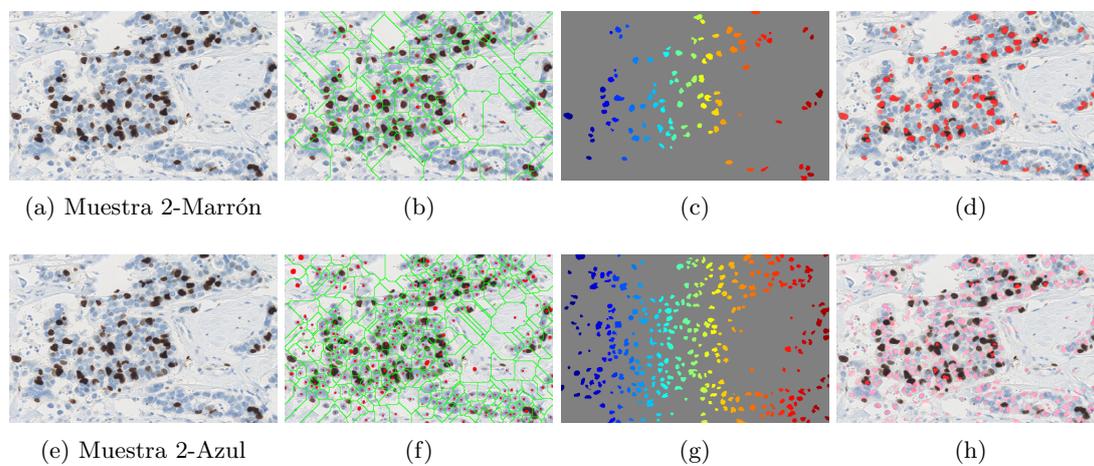


Figura 5.15: Resultados segmentación con el método VCA de la muestra 2. De izquierda a derecha: original > marcadores > watershed > watershed sobre original.

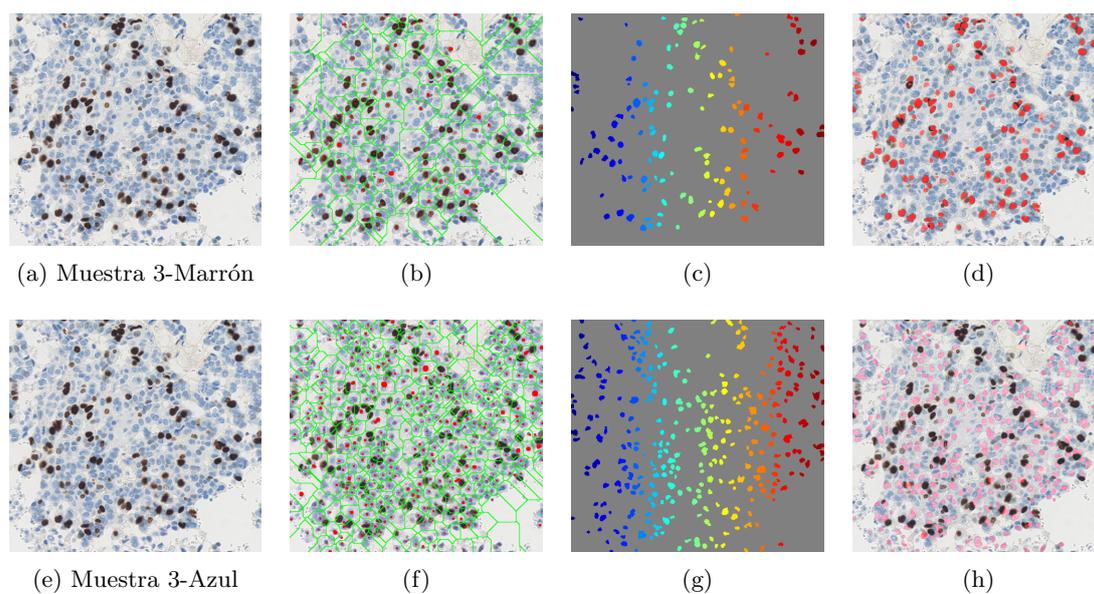


Figura 5.16: Resultados segmentación con el método VCA de la muestra 3. De izquierda a derecha: original > marcadores > watershed > watershed sobre original.

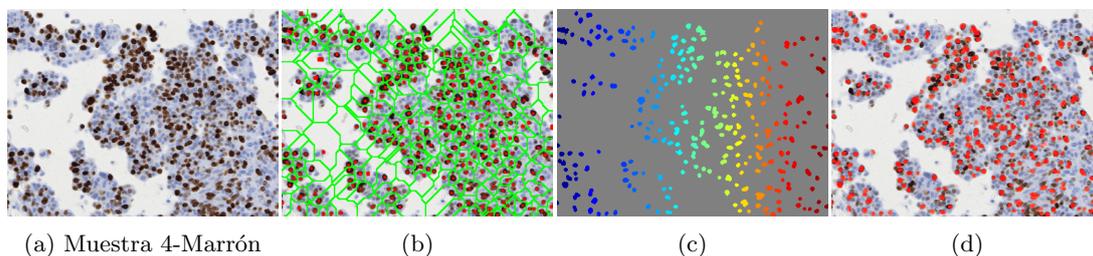


Figura 5.17: Resultados segmentación con el método VCA de la muestra 4. De izquierda a derecha: original > marcadores > watershed > watershed sobre original.

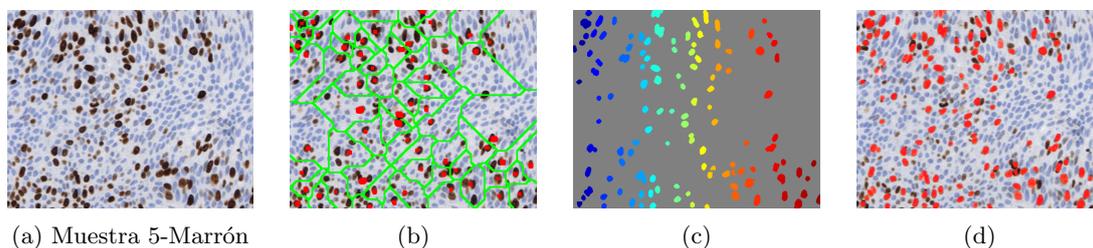


Figura 5.18: Resultados segmentación con el método VCA de la muestra 5. De izquierda a derecha: original > marcadores > watershed > watershed sobre original.

### 5.4.3. Contabilización

Para concluir, a continuación se presenta la contabilización de objetos segmentados tanto en las máscaras segmentadas a mano como en las obtenidas con la aplicación del programa creado en este TFG. Tras dichos datos, se introducirán los índices de evaluación de la segmentación y se expondrán los valores obtenidos para los resultados.

En las Tablas 5.1, 5.2 y 5.3 se muestra la contabilización de los objetos detectados para el método manual, la deconvolución manual y el método VCA. Se dividen en objetos detectados como células y la diferencia con los detectados en la segmentación manual. Además, se incluye el porcentaje de células marrones sobre el total de células detectadas y la diferencia con la segmentación manual.

	Marrones	Azules	Porcentaje
Muestra 1	123	459	21,13 %
Muestra 2	118	361	24,63 %
Muestra 3	142	585	19,53 %
Muestra 4	488	420	53,74 %
Muestra 5	220	536	29,10 %

Tabla 5.1: Número de células detectadas con la segmentación manual.

	Marrones	Dif.	Azules	Dif.	Porcentaje	Dif
Muestra 1	111	12	404	55	21,55 %	0.42 %
Muestra 2	95	23	368	7	20,52 %	4.12 %
Muestra 3	140	2	570	15	18,59 %	0.84 %
Muestra 4	384	104	332	88	53,63 %	0.11 %
Muestra 5	173	47	262	274	39,77 %	10.67 %

Tabla 5.2: Número de células detectadas con la segmentación de la deconvolución de color manual.

	Marrones	Dif.	Azules	Dif.	Porcentaje	Dif.
Muestra 1	119	4	301	158	28,33 %	7.2 %
Muestra 2	101	17	301	60	25,12 %	0.49 %
Muestra 3	131	11	299	244	30,47 %	8.22 %
Muestra 4	262	226	-	-	- %	-
Muestra 5	117	103	-	-	- %	-

Tabla 5.3: Número de células detectadas con la segmentación del método VCA.

A la vista de los resultados se pueden realizar varios comentarios. El primero es que el número de objetos segmentados en las muestras de  $40\times$  se aproxima mucho más al conteo manual que el de las muestras de  $20\times$  (donde la diferencia llega a ser superior a 100 objetos), por lo que se puede afirmar que la resolución de las imágenes obtenidas por el microscopio afecta a la separación celular. El segundo sería que, para el algoritmo aplicado, el método VCA en las mejores ejecuciones del algoritmo presenta mejores resultados en la segmentación de las células marrones que el método de deconvolución manual con coeficientes de separación personalizados. Este resultado se invierte para el caso de las células azules, donde la deconvolución manual es más efectiva. Por último, es necesario remarcar que aunque el conteo de células no sea preciso, la diferencia en el índice de proliferación (porcentaje de células marrones sobre el total de células) es muy baja. Es precisamente este parámetro el que interesa al patólogo para obtener una primera valoración del estado de la muestra en cuanto al avance del cáncer.

### Índices de evaluación

Para evaluar la calidad de los resultados se han elegido dos de los tres métodos sugeridos por la Universidad de Warwick [45] cuya implementación en MATLAB se encuentra disponible en su página web. Todos ellos utilizan como entrada una máscara binaria de etiquetas donde se encuentran los resultados verdaderos, en este caso la segmentación a mano, y una máscara binaria de etiquetas que en este caso sería el resultado de la aplicación de la función *watershed*. Por último, es necesario remarcar

que la segmentación a mano es totalmente subjetiva y se han segmentado las células en base a una opinión propia respecto a cuáles se pueden considerar azules o marrones. Esto es así debido a que algunas células tienen características de ambas coloraciones, haciendo imprecisa su naturaleza.

Los dos índices utilizados son los siguientes [46]:

- **F1-score.** Se utiliza como evaluador de la detección de células. Funciona comparando cada objeto detectado con su intersección con el objeto en la misma posición de la máscara de evaluación. En caso de superar el 50% de intersección se considerará como una detección positiva. En caso de no hacerlo o de que no haya correspondencia entre un objeto de la máscara de comprobación y la de segmentación, se dará como no detectado. Comprende un rango de 0 a 1 siendo uno el mejor valor que se puede obtener de la Expresión 5.1.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (5.1)$$

donde *Precision* y *Recall* se calculan con las expresiones 5.2 y 5.3.

$$Precision = \frac{TP}{TP + FP} \quad (5.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (5.3)$$

siendo *TP* el número de verdaderos positivos, *FP* el número de falsos positivos y *FN* el número de falsos negativos.

- **Índice Dice.** Sirve para evaluar la calidad de la segmentación. Compara cada objeto segmentado con el objeto de la máscara de comprobación con el que produzca mayor solape, calculando así la precisión de solape espacial. Comprende un rango de 0 a 1 siendo el mejor valor que se puede obtener de la Expresión 5.4.

$$Dice(G, S) = \frac{2 * |G \cap S|}{|G| + |S|} \quad (5.4)$$

siendo *G* los píxeles elegidos como máscara de segmentación de referencia y *S* los píxeles de la máscara de segmentación a evaluar.

Es importante tener en cuenta, tanto en la detección como en la segmentación, que los valores obtenidos para el método VCA corresponden a una de las mejores

---

ejecuciones del código. Esto quiere decir que, como se explica en la Sección 5.4.2, se ha escogido uno de los mejores resultados del análisis VCA ya que en cada aplicación del método los resultados eran distintos. Los valores obtenidos para ambos índices han sido los presentados en las Tablas 5.4 y 5.5.

	<i>F1-score Marrón</i>	<i>Dice Marrón</i>	<i>F1-score Azul</i>	<i>Dice Azul</i>
Muestra 1	0.8533	0.7506	0.7132	0.6373
Muestra 2	0.7742	0.6982	0.7133	0.6069
Muestra 3	0.7261	0.7442	0.6184	0.6428
Muestra 4	0.6942	0.4665	0.6112	0.4163
Muestra 5	0.7629	0.614	0.6054	0.4975

Tabla 5.4: Índices obtenidos para la deconvolución de color manual.

	<i>F1-score Marrón</i>	<i>Dice Marrón</i>	<i>F1-score Azul</i>	<i>Dice Azul</i>
Muestra 1	0.9068	0.7759	0.6323	0.5986
Muestra 2	0.7821	0.7041	0.6484	0.5781
Muestra 3	0.7458	0.7614	0.428	0.5331
Muestra 4	0.67	0.624	-	-
Muestra 5	0.6946	0.6789	-	-

Tabla 5.5: Índices obtenidos para el método VCA.

Por último, a la vista de estos resultados, las conclusiones son las siguientes:

- Detección de células marrones.** En este caso, el método que arroja mejores resultados (en términos de *F1-score*) es el VCA a pesar de que en la obtención de marcadores internos las células detectadas son en ambos casos prácticamente las mismas. Esto puede deberse a cómo afectan el preprocesado y el gradiente al resultado de la deconvolución del color. Al no realizarse con los coeficientes más adecuados, la imagen contiene mucho más ruido y elementos indeseados que la separación VCA por lo que es más difícil deshacerse de ellos y afectan al cálculo del gradiente. Un ejemplo de esta separación distinta se muestra en la Figura 5.19, donde se observa el resultado de la separación de células marrones para la deconvolución de color manual y la imagen complementaria del resultado para VCA.

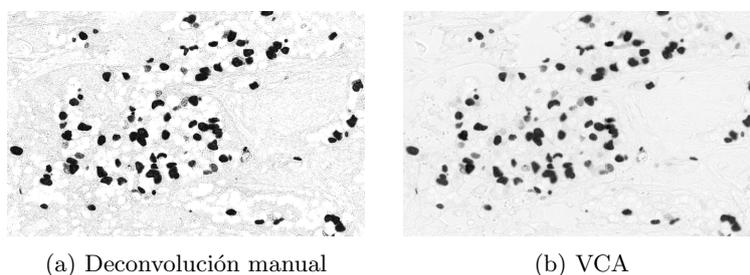


Figura 5.19: Separación de células marrones en deconvolución manual y VCA.

- **Detección de células azules.** En la evaluación de la detección (índice *F1-score*), el método de deconvolución manual es considerablemente más efectivo que el método VCA. Aunque dista de ser perfecto debido a la considerable cantidad de células que no son detectadas, es un claro indicativo de que el método de deconvolución de color manual es un buen camino a seguir en la detección de las células azules de las imágenes de Ki-67.
- **Segmentación de células marrones.** Al igual que en la detección, la conservación de la forma en las células marrones es mejor en el método VCA. Una vez más, podría mejorarse desde el preprocesado ya que afecta a todas las etapas posteriores, incluido el gradiente.
- **Segmentación de células azules.** Al tener una mejor detección, el método de deconvolución de color manual es el mejor candidato para realizar una buena segmentación como se comprueba por los valores obtenidos de *Dice* para todas las muestras. La conservación de la forma para mejorar el resultado respecto a la máscara puede mejorarse, partiendo de unos coeficientes óptimos para la deconvolución, desde el preprocesado y obtención de marcadores.

Por último, se comprueba que las muestras de  $20\times$  generan peores resultados ya que sus valores tanto del índice *Dice* como de *F1-score* son los más bajos para todas las muestras.

## Capítulo 6

# Conclusiones y trabajo futuro

### Índice de contenidos

---

<b>6.1. Introducción</b> . . . . .	<b>71</b>
<b>6.2. Métodos de separación de componentes de color</b> . . . . .	<b>71</b>
<b>6.3. Procesado morfológico</b> . . . . .	<b>72</b>
<b>6.4. Segmentación</b> . . . . .	<b>72</b>
6.4.1. Contabilización . . . . .	72
6.4.2. Discusión . . . . .	73
<b>6.5. Trabajo futuro</b> . . . . .	<b>75</b>

---

## 6.1. Introducción

En este capítulo se extraerán conclusiones de los resultados mostrados en el Capítulo 5, dividiendo dichas conclusiones en tres apartados. Posteriormente, se realizará una pequeña reflexión sobre la segmentación de las imágenes histológicas y, por último, se propondrán líneas futuras de trabajo.

## 6.2. Métodos de separación de componentes de color

Tras la discusión de los resultados puede decirse que los métodos más efectivos para la deconvolución de color son el VCA y el método de deconvolución manual.

El primero presenta una componente de aleatoriedad debido a que, al ser un método automático con una inicialización aleatoria, no siempre se alcanzan los mismos *endmembers* para una misma imagen. Esto produce una modificación del orden de presentación de los resultados o incluso los propios resultados, dificultando la segmentación de las células azules en algunas ejecuciones del código.

El segundo precisa de unos coeficientes específicos para el marcador celular Ki-67 los cuales se desconocen actualmente. Para su cálculo sería necesario poseer una muestra pura de la muestra azul y otra de la muestra marrón, por lo que queda en manos de los patólogos. La ventaja del método es que realiza una separación muy buena de las componentes, por lo que se recomienda profundizar más en el uso de la deconvolución de color manual pero su principal desventaja es su poca versatilidad. Esta es debida a que es muy difícil conseguir que las células se tiñan siempre de la misma manera. Además, se debe garantizar que los parámetros y la configuración del microscopio durante la adquisición sean siempre los mismos, lo cual no siempre es posible.

En cuanto a los otros métodos, sería interesante darle una oportunidad al método mencionado al final de la Sección 4.5.3, el SBICA [2], ya que utiliza el método ICA modificado para hallar la matriz de descomposición y después aplicarla en la deconvolución de color manual, prometiendo buenos resultados.

---

## 6.3. Procesado morfológico

El procesado escogido para la realización de este trabajo no es en absoluto definitivo. La utilización de los operadores morfológicos para la limpieza de la imagen puede ser muy agresiva si los métodos previos no han realizado una buena separación de las componentes. Es uno de los apartados más importantes ya que la segmentación *watershed* utilizará como máscara una combinación de los marcadores de segmentación y el gradiente de la imagen procesada, por lo que todos los cálculos se realizan a partir de este punto.

Dadas las características de las imágenes disponibles, se han escogido soluciones de compromiso para los tamaños de los elementos estructurantes pero no quiere decir que sean las mejores. Se ha implementado un método en el que el usuario elige el tamaño mínimo de célula que desea ser segmentada, lo que genera un radio mínimo de objetos que han de aparecer. El problema de la solución es que el tamaño es demasiado sensible (diferencias de un par de píxeles cambian totalmente los resultados) por lo que para la realización del trabajo se ha fijado a un valor umbral que conservaba la mayoría de elementos importantes.

## 6.4. Segmentación

Existen múltiples formas de obtener marcadores tanto internos como externos [40], por lo que en función del tipo de imagen y de las características que presenten sus células sería interesante cambiar el método. Los resultados obtenidos con el trabajo realizado son satisfactorios, por lo que se recomienda continuar por esta línea de trabajo buscando mejores ajustes de las variables.

### 6.4.1. Contabilización

La contabilización no es más que el conteo de objetos segmentados tras la aplicación del método *watershed*, por lo que su contabilización es precisa con el uso de *bwconncomp* en MATLAB.

Es importante recordar que el objetivo del uso del marcador Ki-67 es encontrar el índice de proliferación celular (porcentaje de células marrones sobre el número total de células) y tras los resultados del conteo se aprecia que, a pesar de no ser preciso, la diferencia entre el índice de proliferación de la segmentación y el de la referencia es

---

muy pequeña. Esto es un resultado importante a tener en cuenta para el patólogo y significa el cumplimiento de un objetivo fundamental del trabajo ya que se consigue implementar un sistema de cribado por regiones que muestren al patólogo las áreas más críticas de la muestra con mayor concentración de células malignas.

### Índices de evaluación

Los índices utilizados son una buena forma de obtener resultados objetivos acerca de la calidad de la segmentación. Una buena segmentación no es sólo una contabilización precisa ya que existen falsos positivos o negativos, por lo que la información de posición y tamaño respecto al original es esencial para saber qué se ha detectado. Es una tarea importante conseguir acercar esos índices a su máximo valor para poder decir que se ha obtenido una segmentación fiable para utilizarse en el campo médico.

#### 6.4.2. Discusión

Hay múltiples factores en toda la cadena, desde la adquisición de la muestra hasta la segmentación, que condicionan la obtención de un resultado adecuado.

En primer lugar, el tipo de tejido del que está compuesto la célula. La organización celular no es igual en muestras epiteliales que las muestras de zonas de la vejiga estudiadas durante la realización del TFG. En el primer caso, las células tienden a estar mejor distribuidas y los casos en los que aparecen agrupaciones de células que se solapan entre sí es menor, por lo que favorecerá una buena segmentación:

En segundo lugar, el marcador celular empleado. Existen diferentes estudios para marcadores como H&E o HDAB que proporcionan unos coeficientes de deconvolución de color manual óptimos. Estos coeficientes vienen incluidos, por ejemplo, en la herramienta de deconvolución de la aplicación *ImageJ* [35]. Con ellos, las imágenes obtenidas son más uniformes y con menos componentes de ruido, realizando una separación más precisa que no es realizable con el marcador Ki-67.

En tercer lugar, la resolución y calidad de las imágenes WSI. Como se ha apreciado durante el desarrollo del análisis de imagen, los resultados obtenidos para las imágenes de 40× son considerablemente mejores. Esto ha sido un grave inconveniente de por sí, ya que la variedad de muestras disponibles para la experimentación era muy escasa, siendo todas las proporcionadas por la patóloga de las mismas características.

En cuarto y último lugar, el análisis. Existen numerosos puntos críticos durante la

---

preparación de la imagen para su segmentación.

- El primero y más importante de ellos sería el preprocesado, en concreto el tamaño de los elementos estructurantes de los operadores morfológicos. Para cada muestra, además de otras características distintas, el tamaño de las células es variable. Esto quiere decir que es necesario ajustar todos los tamaños en función de las características de cada muestra, por lo que es un punto que necesita de la interacción del usuario para definir un valor adecuado. En caso contrario, se corre el riesgo de eliminar elementos de interés o de dejar otros que provoquen falsos positivos en pasos posteriores. Además, un mal preprocesado afectará al gradiente que se utilizará como máscara de segmentación y también afectará a la obtención de los marcadores internos y externos.
- El segundo sería la obtención de los marcadores para la aplicación del método *watershed*. Los resultados observados con la *Fast Radial Symmetry Transform* son bastante convincentes ya que, como se observa en la segunda columna de las Figuras 5.9-5.18, prácticamente la totalidad de células son detectadas como marcador interno. Aún así, éste método también depende de los tamaños escogidos en el apartado de preprocesado, por lo que es sensible a variaciones.
- El tercero sería el método de segmentación. Aunque las pruebas han sido realizadas con el método de segmentación *watershed* en vistas a separar las agrupaciones de células solapadas, quizás sería una buena opción explorar otras alternativas no contempladas en este trabajo.

El desarrollo de este TFG ha servido para obtener conocimientos sobre un campo del tratamiento de imagen que era totalmente desconocido antes de empezar. Además, ha servido para afianzar y ampliar los conocimientos obtenidos durante la carrera tanto de MATLAB como de tratamiento de imagen. Por último, tras la finalización de este TFG se ha conseguido implementar una aplicación totalmente funcional que es capaz de trabajar con imágenes de microscopía de gran tamaño, incluyendo su análisis y muestra de resultados. De esta forma se han podido cumplir todos los objetivos propuestos al inicio del estudio quedando para trabajo futuro las distintas mejoras propuestas en la Sección 6.5.

---

## 6.5. Trabajo futuro

Por distintos motivos expuestos durante el desarrollo de la memoria, han quedado múltiples opciones o características que, o bien no han podido ser implementadas o necesitan de mejoras. Es por ello que las líneas de trabajo propuestas son las siguientes:

- **Visualización en la interfaz.** Profundizar en el uso de archivos *rset* para lograr implementar un visionado adaptativo de la imagen en máxima resolución.
  - **Adaptación de la aplicación a otro lenguaje de programación.** En vistas a una aplicación de distribución más sencilla, sería aconsejable la migración del programa a otro lenguaje de programación como C++. El inconveniente puede ser la necesidad de programar algunos de los métodos matemáticos necesarios. Se recomienda el uso de las librerías *Cimg* y *OpenCV*.
  - **Análisis de imagen.** Profundizar en la vía del análisis a través de los métodos VCA y de la deconvolución de color manual, obteniendo mejores coeficientes de separación para este último. Además, se propone la búsqueda de un método de automatización del tamaño de los elementos estructurantes a partir de un tamaño mínimo elegido por el usuario.
-

# Bibliografía

- [1] Humayun Irshad ; Antoine Veillard ; Ludovic Roux ; Daniel Racoceanu,´. “Methods for Nuclei Detection, Segmentation, and Classification in Digital Histopathology: A Review - Current Status and Future Potential,” *IEEE REVIEWS IN BIOMEDICAL ENGINEERING* ,vol. 7,pp. 97 - 114 , 2014.
- [2] Najah Alsubaie ; Shan E Ahmed Raza ; Nasir Rajpoot´. “Stain deconvolution of histology images via independent component analysis in the wavelet domain,” *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)* ,pp. 803 - 806 , 2016.
- [3] Ventana Medical Systems, Inc, “Cancer testing: The journey of a tissue biopsy.” <http://www.ventana.com/journey/>, 2014. [Acceso: 20/08/2016.]
- [4] Michael T. McCann ; John A. Ozolek ; Carlos A. Castro ; Bahram Parvin ; Jelena Kovacevic. “Automated Histology Analysis. Opportunities for signal processing,” *IEEE Signal Processing Magazine*, vol. 32, pp. 78 - 87 , 2015.
- [5] Wikimedia Commons, the free media repository, “Tissue processing - Paraffin inclusion of a histological sample.”[https://commons.wikimedia.org/wiki/File:Tissue\\_processing\\_-\\_Paraffin\\_inclusion\\_of\\_a\\_histological\\_sample.jpg](https://commons.wikimedia.org/wiki/File:Tissue_processing_-_Paraffin_inclusion_of_a_histological_sample.jpg), 2009. [Acceso: 14/08/2016.]
- [6] Scholzen T ; Gerdes J. “The Ki-67 protein: from the known and the unknown,” *Journal of Cellular Physiology* , vol. 182, no. 3, pp. 311-322, 2000.
- [7] Geoffrey M Cooper, *The cell. A Molecular Approach*, 2nd Edition. Sinauer Associates; 2000.
- [8] Digital Pathology Association, “About Digital Pathology.” <https://digitalpathologyassociation.org/about-digital-pathology>, 2016. [Acceso: 1/08/2016.]

- 
- [9] Thomas J. ; Fuchsa,b ; Joachim M. Buhmann. “Computational pathology: Challenges and promises for tissue analysis,” *Computerized Medical Imaging and Graphics* , vol. 35, pp. 515-530, 2011.
- [10] Michael T.McCann ; John A.Ozolek ; Carlos A.Castro ; Bahram Parvin ; Jelena Kovacevic,´. “Automated Histology Analysis: Opportunities for Signal Processing,” *IEEE Signal Processing Magazine*,vol. 32, no. 1,pp. 78 - 87 , 2015.
- [11] Farahani N ; Parwani AV ; Pantanowitz L. “Whole slide imaging in pathology: advantages, limitations, and emerging perspectives,” *Pathology and Laboratory Medicine International*, vol. 7, pp. 23-33, 2015.
- [12] Toby C. Cornish. ‘An introduction to digital whole slide imaging and whole slide image analysis,’ *Johns Hopkins School of Medicine*, 2013.
- [13] Ventana Medical Systems, Inc, “VENTANA iScan HT.” <http://www.ventana.com/product/page?view=iscanht>, 2016. [Acceso: 15/08/2016.]
- [14] Wikipedia, the free encyclopedia, “Tagged Image File Format.” [https://en.wikipedia.org/wiki/Tagged\\_Image\\_File\\_Format](https://en.wikipedia.org/wiki/Tagged_Image_File_Format). [Acceso: 10/08/2016.]
- [15] OpenSlide <http://openslide.org/>, 2016. [Acceso: 1/03/2016.]
- [16] Michelle D Reid ; Pelin Bagci ; Nobuyuki Ohike ; Burcu Saka ; Ipek Erbarut Seven ; Nevra Dursun ; Serdar Balci ; Hasan Gucer ; Kee-Taek Jang ; Takuma Tajiri ; Olca Basturk1 ; So Yeon Kong ; Michael Goodman ; Gizem Akkas ; Volkan Adsay. “Calculation of the Ki67 index in pancreatic neuroendocrine tumors: a comparative analysis of four counting methodologies,” *Modern Pathology*, vol. 28, pp. 686-694, 2015.
- [17] ImageJ, “ImageJ. An open platform for scientific image analysis.” <http://imagej.net/Welcome>. [Acceso: 1/08/2016.]
- [18] MathWorks, “fordanic/openslide-matlab.” <https://es.mathworks.com/matlabcentral/fileexchange/48887-fordanic-openslide-matlab>, 2016. [Acceso: 1/03/2016.]
- [19] JGraph Ltd. “Draw.io - Diagram Environment ” <https://www.draw.io/>, 2015. [Acceso: 10/08/2016.]
-

- 
- [20] Steve Eddins, “Dealing with “Really Big” Images: Block Processing.” <http://blogs.mathworks.com/steve/2011/08/16/dealing-with-really-big-images-block-processing/>, 2011. [Acceso: 1/06/2016.]
- [21] Steve Eddins, “Dealing with “Really Big” Images: Image Adapters.” <http://blogs.mathworks.com/steve/2011/09/23/dealing-with-really-big-images-image-adapters/>, 2011. [Acceso: 1/06/2016.]
- [22] Steve Eddins, “Dealing with “Really Big” Images: Viewing.” <http://blogs.mathworks.com/steve/2011/08/05/dealing-with-really-big-images-viewing/>, 2011. [Acceso: 1/06/2016.]
- [23] R.Gonzalez ; R. Woods, and S. Eddins, *Digital Image Processing Using MATLAB*. Gatesmark Publishing, 2009.
- [24] MathWorks, “Block Size and Performance.” <http://es.mathworks.com/help/images/block-size-and-performance.html?refresh=true>, 2016. [Acceso: 1/06/2016.]
- [25] R. Bro ; A.K. Smilde, “Principal component analysis,” *Analytical Methods*, vol. 6, no. 9, pp. 2812–2831, 2014.
- [26] Computer vision for dummies, “Feature extraction using PCA.” <http://www.visiondummy.com/2014/05/feature-extraction-using-pca/>. [Acceso: 16/08/2016.]
- [27] J.M.P. Nascimento ; J.M.B. Dias. “ Vertex component analysis: a fast algorithm to unmix hyperspectral data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 898 - 910 , 2005.
- [28] OpenCV, “Affine Transformations.” [http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/warp\\_affine/warp\\_affine.html](http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/warp_affine/warp_affine.html). [Acceso: 1/07/2016.]
- [29] Weisstein, Eric W, “Simplex.” *From MathWorld—A Wolfram Web Resource* <http://mathworld.wolfram.com/Simplex.html>. [Acceso: 5/07/2016.]
-

- 
- [30] Weisstein, Eric W, “Convex Hull.” *From MathWorld—A Wolfram Web Resource* <http://mathworld.wolfram.com/ConvexHull.html>. [Acceso: 5/07/2016.]
- [31] J.M.P. Nascimento ; J.M.B. Dias. “Vertex component analysis MATLAB Code,” <http://www.lx.it.pt/~bioucas/code.htm>, 2004. [Acceso: 10/07/2016.]
- [32] Hyvärinen A ; Oja E. “Independent component analysis: algorithms and applications,” *Neural Networks*, vol. 13, pp. 411-430, 2000.
- [33] Brian Moore, “PCA and ICA Package.” <http://es.mathworks.com/matlabcentral/fileexchange/38300-pca-and-ica-package>. [Acceso: 1/07/2016.]
- [34] C. Ruifrok ; Dennis A. Johnston. “Quantification of histochemical staining by color deconvolution,” *Analytical and Quantitative Cytology and Histology*, vol. 23, no. 4, pp. 291 - 299 , 2001.
- [35] G. Landini, “Colour Deconvolution..” <http://www.mecourse.com/landinig/software/cdeconv/cdeconv.html>, 2015. [Acceso: 20/05/2016.]
- [36] Department of Computer Science, University of Warwick, “Stain Normalisation Toolbox.” <https://www2.warwick.ac.uk/fac/sci/dcs/research/combi/research/bic/software/sntoolbox/>, 2015. [Acceso: 30/06/2016.]
- [37] Center of Mathematical Morphology, “Basics in Mathematical Morphology.” <http://cmm.enscm.fr/~serra/cours/index.htm>, 2000. [Acceso: 06/07/2016].
- [38] Antonio Albiol Colomer. “Tema 8: Morfología,” *Tratamiento Digital de la Imagen*, Universidad Politécnica de Valencia, 2015.
- [39] MathWorks, “Marker-Controlled Watershed Segmentation.” <http://es.mathworks.com/help/images/examples/marker-controlled-watershed-segmentation.html>, 2016. [Acceso: 22/06/2016.]
- [40] M. Veta ; A. Huisman ; M. A. Viergever ; P. J. van Diest ; J. P. W. Pluim. “Marker-controlled watershed segmentation of nuclei in H&E stained breast cancer biopsy images,” *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 618 - 621 , 2011.
-

- 
- [41] Steve Eddins, “ The Watershed Transform: Strategies for Image Segmentation.” <http://es.mathworks.com/company/newsletters/articles/the-watershed-transform-strategies-for-image-segmentation.html>, 2002. [Acceso: 22/06/2016.]
- [42] Gareth Loy ; Alexander Zelinsky. “Fast Radial Symmetry for Detecting Points of Interest,” *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, vol. 25, no. 8, pp. 959 - 973 , 2003.
- [43] Gareth Loy. “Fast Radial Symmetry Transform,” [http://www.nada.kth.se/~gareth/homepage/local\\_site/code.htm](http://www.nada.kth.se/~gareth/homepage/local_site/code.htm), 2005. [Acceso: 15/07/2016.]
- [44] Milan Gavrilovic ; Jimmy C. Azar ; Joakim Lindblad ; Carolina Wählby, Ewert Bengtsson ; Christer Busch ; Ingrid B. Carlbom, Member. “Blind Color Decomposition of Histological Images,” *IEEE TRANSACTIONS ON MEDICAL IMAGING*, vol. 32, no. 6, pp. 983 - 994, 2013.
- [45] Korsuk Sirinukunwattana. “Evaluation of BioImage Analysis. Department of Computer Science, University of Warwick” <http://www2.warwick.ac.uk/fac/sci/dcs/research/combi/research/bic/glascontest/evaluation/>, 2015. [Acceso: 18/08/2016.]
- [46] Kolawole Oluwole Babalola ; Brian Patenaude ; Paul Aljabar ; Julia Schnabel ; David Kennedy ; William Crum ; Stephen Smith ; Tim Cootes ; Mark Jenkinson ; Daniel Rueckert. “An evaluation of four automatic methods of segmenting the subcortical structures in the brain,” *Neuroimage*, vol. 47, pp. 1435 - 1447, 2009.
-