



UNSAM
UNIVERSIDAD
NACIONAL DE
SAN MARTÍN

**Implementación de una herramienta de análisis de imágenes
dinámicas en PET-CT**

por

Lucía Alejandra Dugón

Tesis de Licenciatura en Diagnóstico por Imágenes

**Escuela de Ciencia y Tecnología
Universidad Nacional De San Martín**

Agosto de 2019

Tema: Cuantificación de imágenes dinámicas de PET-CT

Alumna: Lucia Alejandra Dugón

LU n° : CPU7293CyT-3802

Lugar de trabajo: Escuela de Ciencia y Tecnología - Universidad Nacional de San Martín

Director del trabajo: Alejandro Valda

Fecha de iniciación: 7 de junio de 2017

Fecha de finalización: 31 de Agosto de 2019

Fecha de examen: 16 de Septiembre de 2019

Jurado: Lic. Pablo Sanabria

Dra. Patricia Parma

Lic. Marcelo Ruffolo

Trabajo final de Tesis aprobado por:

.....
Autora

.....
Director

.....
Director de la Comisión de Tesis de Licenciatura

Resumen

Los equipos de PET-CT (Tomografía por emisión de positrones – Tomografía Computada) vienen provistos de software diseñado para llevar a cabo distintos análisis cuantitativos de concentración de actividad del radiofármaco entre los cuales se encuentran la determinación de SUV (por *standardized uptake value*, valor de captación estandarizado) sobre las ROIs (regiones de interés) trazadas sobre las imágenes. Esta posibilidad se aplica también a series de imágenes temporales correspondientes a estudios dinámicos. Sin embargo, en la amplia mayoría de los casos, tales programas se encuentran instalados sólo en estaciones de trabajo que generalmente están contiguas a la consola de adquisición del equipo PET-CT en los servicios de medicina nuclear, lo que implica grandes restricciones de uso.

Objetivo: desarrollar una herramienta informática basada en software libre que permita realizar cuantificaciones basadas en SUV –y magnitudes asociadas al mismo– sobre imágenes de adquisiciones dinámicas de PET-CT y que pueda ejecutarse en diversos sistemas operativos (multiplataforma).

Método: Para el desarrollo de la herramienta se utilizó el lenguaje de macros y el entorno provisto por el programa de visualización, análisis y procesamiento de imágenes Fiji y de uno de sus módulos adicionales llamado PET-CT Viewer. La herramienta se aplicó al análisis de tres tipos de estudios dinámicos de PET-CT realizados en un equipo Philips Gemini TF TOF 64 en el Centro de Educación Médica e Investigaciones clínicas Norberto Quirno (CEMIC) para evaluar la capacidad de la herramienta desarrollada. Estos mismos estudios fueron analizados utilizando el programa PMOD ampliamente validado en el campo de las imágenes biomédicas. Las magnitudes comparadas fueron el SUV máximo y el SUV promedio obtenidos sobre diversas regiones volumétricas de interés. Otras magnitudes, variantes del SUV no brindadas por PMOD, también fueron exploradas.

Resultados: se trazaron gráficos comparativos obtenidos tanto por el código de programación creado como por PMOD de las magnitudes SUV_{max} y SUV_{medio} sobre los tres grupos de estudio obteniéndose una muy buena concordancia entre los mismos. Se trazaron gráficos contrastando las variables mencionadas anteriormente añadiendo dos nuevas tendencias a cosechar mayor información sobre las ROIs delimitadas.

Conclusiones: se ha concluido que la herramienta creada obtiene correctamente los resultados obtenidos con referencia a PMOD para estudios dinámicos de PET-CT. Además, se han incorporado dos magnitudes de utilidad relacionadas al SUV (calculadas por PET-CT Viewer) cuya finalidad consiste en adicionar mayor información sobre las ROIs diseñadas. Considerando que el desempeño del código ha sido satisfactorio, que resulta de fácil implementación en diversos sistemas operativos y que se ejecuta dentro de un sólido programa destinado al análisis y procesamiento de imágenes en general, se concluye que la herramienta desarrollada podrá brindar un valioso apoyo en el procesamiento de estudios cinéticos de PET-CT.

Agradecimientos

Quiero agradecer a la Universidad de San Martín y al Instituto CEMIC que hizo posible la realización de este trabajo de tesis.

A mi director de tesis Alejandro Valda que, sin su dedicación y paciencia no hubiera logrado llegar a concluir con este trabajo transmitiéndome sus conocimientos día tras día.

A mis padres, por darme la vida y guiarme a formarme constantemente en mi profesión.

Tabla de contenidos

1	INTRODUCCIÓN	7
2	IMAGEJ	9
2.1	Antecedentes, generalidades y utilidad	9
2.2	Entorno de trabajo en ImageJ	10
2.3	Programación en ImageJ	11
3	FIJI: GENERALIDADES	13
4	PET-CT VIEWER	15
4.1	Descarga y utilización	15
4.2	Archivos separados por comas (CSV): definición y estructura según PET-CT Viewer	20
5	SUV (VALOR ESTANDARIZADO DE CAPTACIÓN)	23
5.1	SUV máximo y medio (<i>SUV_{max}</i> y <i>SUV_{medio}</i>)	23
6	HERRAMIENTA DE ANÁLISIS DESARROLLADA EN ESTE TRABAJO	25
6.1	Utilización del archivo CSV en el código creado	25
6.2	Formato estándar DICOM	25
6.3	Descripción del código	26
7	VALIDACIÓN DE LA HERRAMIENTA	29
7.1	Materiales y Métodos	29
7.1.1	Primer caso: estudio con distribución de actividad homogénea	29
7.1.2	Segundo caso: estudio clínico con distribución de actividad simple (próstata)	30
7.1.3	Tercer caso: estudio animal con distribución compleja	31
8	RESULTADOS	32
8.1	Validación de la herramienta desarrollada utilizando PMOD	32
8.2	Análisis de variables relacionadas al SUV generadas por PET-CT Viewer	40

9	DISCUSIÓN	46
10	CONCLUSIÓN	48
11	REFERENCIAS BIBLIOGRÁFICAS	49
12	ANEXO	50
12.1	Código de programación creado en Fiji para cuantificar Imágenes dinámicas en PET-CT.....	50

1 Introducción

En la actualidad, la tomografía por emisión de positrones asociada a la tomografía de rayos X (abreviada como PET-CT, por *positron emission tomography – computed tomography*) se considera una modalidad diagnóstica poderosa para la detección, entre otras, de lesiones cancerosas facilitando la planificación de la terapia a seguir así como también la evaluación del tratamiento comenzado hasta la valoración de un pronóstico frente a la enfermedad (Lodge 2017). Gracias a la doble modalidad de imagen, la PET-CT otorga, por un lado, resultados cuantitativos en el plano metabólico-funcional asociados a valores referidos a la concentración del radiofármaco captado por los diversos tejidos (PET) y, por otro lado, resultados cualitativos basados en la visualización de la imagen PET en conjunción con la imagen estructural dada por la CT (ver por ejemplo Cherry et al. 2012). Existen diversos abordajes para la extracción de datos cuantitativos en PET-CT, muchos de ellos están basados sobre el análisis de conjuntos de vóxeles determinados a partir del trazado de regiones llamadas regiones de interés (o ROI por *region of interest*). Éstas pueden ser bi o tridimensionales, en cuyo caso suelen también denominarse VOI (*volume of interest*). En el caso de un estudio realizado como una serie temporal de adquisiciones (también denominado estudio cinético o estudio dinámico), es relevante realizar el proceso de cuantificación sobre una determinada ROI (o VOI), idéntica en cada una de las adquisiciones. De esta manera se puede estudiar, por ejemplo, la evolución de la concentración del radiofármaco a lo largo del tiempo.

El valor de captación estandarizado o SUV (abreviatura en inglés por *standardized Uptake value*) es una magnitud muy utilizada en PET-CT con la que se busca establecer una medida de concentración relativa del radiofármaco. Existen también otras magnitudes relacionadas con el SUV que tienden a corregir diversas imprecisiones o sesgos de éste. La magnitud llamada SUL (abreviatura en inglés por *standardized Uptake value normalized to lean body mass*), por ejemplo, es otro valor de concentración pero normalizado sobre la masa corporal magra –en lugar de la masa corporal total que se utiliza en el cálculo del SUV- y que generalmente se tiene en cuenta en el avance de una lesión (Wahl et al. 2009).

Es habitual que en los servicios de PET-CT, los médicos especialistas en medicina nuclear que elaboran los informes realicen las cuantificaciones de los estudios empleando programas diseñados por las distintas marcas que comercializan los equipos. Sin embargo, dichos programas poseen algunas limitaciones para desarrollar de forma completa la tarea de cuantificar debido a la presencia limitada de herramientas para llevar a cabo diversos análisis, particularmente cinéticos, de la concentración del radiofármaco y a la acotada posibilidad en el acceso a las mismas por toda la comunidad institucional, dependiendo de licencias para su utilización. Es aquí que nace la necesidad de desarrollar un código que facilite la ejecución de análisis cuantitativos sobre ROIs trazadas en un tejido utilizando series dinámicas en PET-CT. Para desarrollar dicho código se eligió el programa de dominio público ImageJ empleado actualmente para el procesamiento y análisis de diversos tipos de imágenes. ImageJ no sólo contiene una interfaz con una amplia variedad de comandos y funciones para un trabajo interactivo sino que también posee un lenguaje simple de programación de macrocomandos. El programa puede descargarse y ejecutarse en cualquier computadora con cualquier sistema operativo en donde pueda ejecutarse programas escritos en Java. El empleo de ImageJ resulta así de baja complejidad y sin la necesidad de renovación de licencias para su uso.

Ahora bien, en la actualidad existen numerosas herramientas computacionales de gran ayuda para llevar a cabo las mencionadas cuantificaciones en las imágenes médicas, una de ellas es el programa PMOD pero requiere de una licencia comercial para su utilización. El sitio web del Centro PET de la Universidad de Pittsburgh de EEUU (University of Pittsburgh, PET Research

Center) posee una variada lista de programas y recursos de software para PET¹, tanto gratuitos como comerciales. Entre ellos se encuentran, por ejemplo, OSIRIX², SPM³, MIPAV⁴. OSIRIX presenta una versión gratuita que está limitada principalmente por la cantidad de imágenes que pueden ser visualizadas y procesadas, su uso no es compatible para el campo del diagnóstico médico sino solamente para fines académicos. Si se desea trabajar con un gran flujo de datos debe optarse por la versión paga de OSIRIX. Por otra parte, SLICER (Fedorov et al. 2012) es otra opción que permite procesar imágenes médicas, escrita en lenguaje JAVA al igual que ImageJ y de libre acceso, pero hay que destacar que SLICER contiene un lenguaje y un manejo del software con un grado de complejidad mayor en comparación a la herramienta elegida para el desarrollo de este trabajo.

ImageJ incluye una alta compatibilidad para cargar y procesar imágenes en diversos formatos como TIFF, JPEG, GIF y DICOM, entre otros. DICOM es el formato estándar utilizado en imágenes médicas a nivel mundial, que establece, en particular, cómo deben organizarse los datos de los estudios médicos mediante ficheros. Estos ficheros contienen por un lado, un encabezado con datos del estudio, del paciente, de la institución donde se obtuvieron las imágenes, datos del equipo, de la modalidad de imágenes utilizada, entre otras y, por otro lado, contiene un cuerpo con los datos de las imágenes propiamente dichas (datos de cada pixel). Así, para el desarrollo de la herramienta de cuantificación creada durante este trabajo se precisó de la extracción de datos específicos del encabezado que se detallarán más adelante.

El presente documento se inicia con la descripción, funcionalidad y generalidades de ImageJ y de una de sus variantes llamada Fiji, que es sobre la cual se desarrolló la herramienta de análisis objeto de este trabajo. Asimismo se presenta la funcionalidad y los aspectos que resultan aquí relevantes de una extensión de programa (*plugin*) de Fiji, sobre la que se apoya nuestra herramienta, llamada PET-CT Viewer (Tai and Kanoun 2019). Dado que los datos extraídos por PET-CT Viewer están basados en el SUV y algunas de sus variantes, se introduce luego el concepto de SUV. A continuación se plantean los pasos realizados para desarrollar nuestra herramienta de análisis y el modo de funcionamiento de la misma. La evaluación del código se lleva a cabo mediante el análisis de tres estudios cinéticos que presentan diversos grados de complejidad. Los resultados obtenidos se contrastan con los obtenidos por un programa independiente ampliamente validado. Los tres estudios anteriores se utilizan también para mostrar la utilidad de otros resultados arrojados por el módulo PET-CT Viewer relacionados con el SUV. Una discusión de los resultados y una conclusión del trabajo cierran el cuerpo principal de este documento. Finalmente, en un anexo se incluye el código fuente de la herramienta desarrollada en este trabajo de tesis.

¹PET: Software Resources | Department of Radiology; <https://www.radiology.pitt.edu/pet-software-resources.html>, página visitada el 29 de agosto de 2019.

²<https://www.osirix-viewer.com/>

³<https://www.fil.ion.ucl.ac.uk/spm/>

⁴<https://mipav.cit.nih.gov/>

2 ImageJ

2.1 Antecedentes, generalidades y utilidad

ImageJ es un programa de acceso libre creado por Wayne Rasband en 1987 (Rasband 2019, Schneider et al. 2012) que permite el procesamiento de imágenes y puede descargarse en cualquier sistema operativo ya sea Windows, Mac OS, Mac OS X o Linux con una versión Java de al menos 1.4 debido a que el programa está escrito en este lenguaje. ImageJ ha sufrido actualizaciones con el paso del tiempo donde se ha mejorado principalmente sus distintos macros y plugins enfocados a diversas ramas biomédicas, los macros son instrucciones complejas que se ejecutan de manera secuencial facilitando la automatización de tareas repetitivas y los plugins son programas independientes que realizan tareas específicas agregando funcionalidad nueva al programa principal. En la figura 1, puede observarse la diversidad de programas constituidos en relación a ImageJ con el paso del tiempo. Se inicia con NIH Image en el año 1987 llegando, hasta el momento⁵, a la versión denominada ImageJFX originada en 2015.

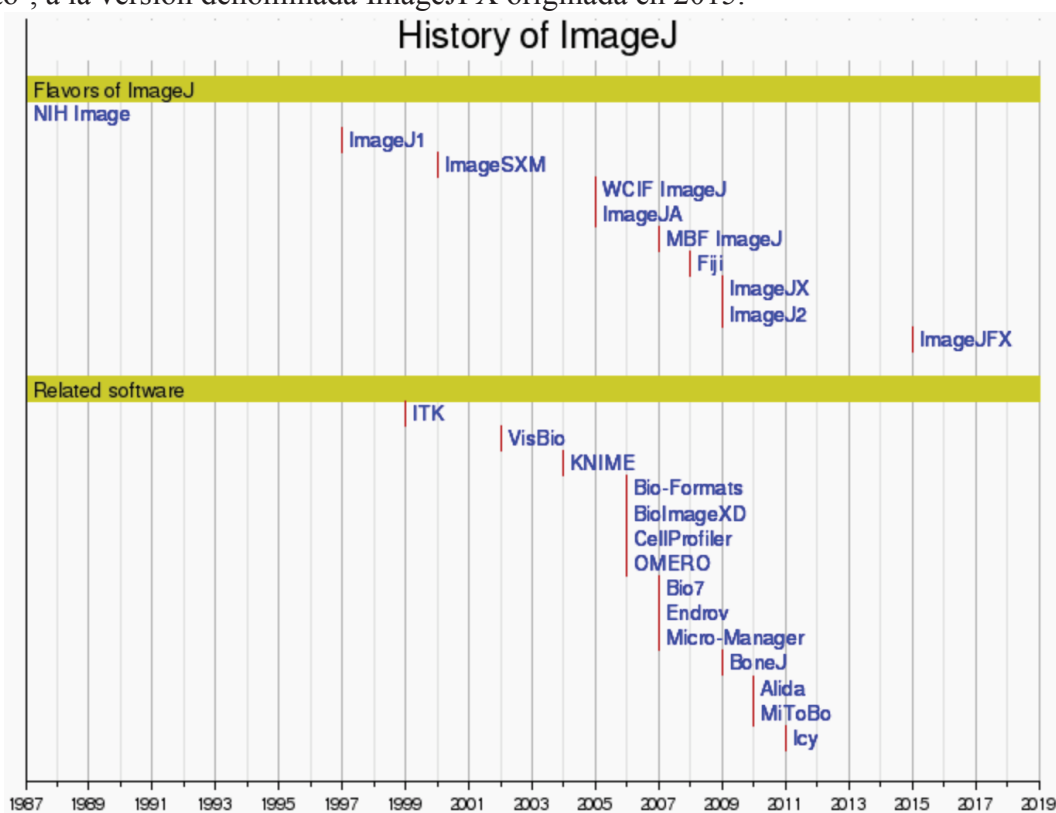


Figura 1. Línea de tiempo que demuestra el avance en la creación de los diversos softwares iniciando por NIH Image en 1987 (ver nota al pie de esta página).

ImageJ contiene un foro en su página web donde usuarios y programadores de todo el mundo pueden sugerir cambios o transmitir inconsistencias encontradas en el código provocando críticas constructivas tendientes a resolver problemas en las sucesivas actualizaciones que se lleven a cabo en el mismo. Las respuestas a conflictos pueden solucionar situaciones no sólo a la persona que consulta sino también a toda la comunidad que participa activamente del foro.

ImageJ presenta multiplicidad de herramientas para interactuar con una o diversas imágenes a la vez pudiendo crear o visualizar series de imágenes o descomponer esas series en imágenes

⁵ <https://imagej.net/ImageJ>, página actualizada el 10 de julio de 2019, a las 06:37 hs, según el acceso realizado el 27 de agosto de 2019.

individuales (figura 2). El programa incorpora funciones básicas de tratamiento de imágenes, como el trazado de perfiles de intensidad, visualización y modificación de histogramas de intensidades, cálculo de áreas, medición de ángulos y la aplicación de filtros para provocar efectos concretos sobre las imágenes, ya sea, por ejemplo, realce de bordes o difuminados, entre otros.

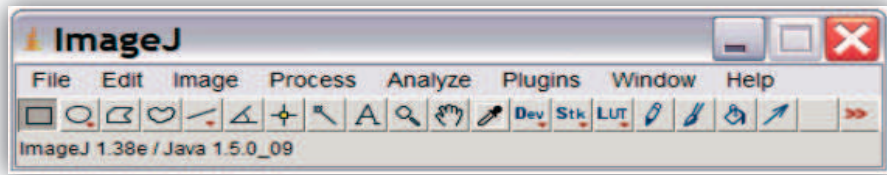


Figura 2. Barra de menú de ImageJ con los distintos submenús a seleccionar dependiendo de las tareas a realizar y diversas herramientas a ejecutar.

También se puede apreciar en la ventana principal de ImageJ la barra de menús en la fila superior y herramientas para trabajar con alguna imagen en la fila del medio. Mientras que en la fila inferior, por su parte, se localiza una barra de estado, que indica la versión actual de ImageJ y de Java con el porcentaje de memoria en uso. En el caso de estar trabajando con una imagen, en la barra de estado también es posible visualizar las coordenadas y el valor del pixel sobre la posición en la que se encuentra el cursor y las coordenadas y las dimensiones de las regiones que se tracen de manera interactiva.

Este programa es capaz de abrir, editar, analizar, procesar, guardar y visualizar imágenes de 8, 16, 32 y hasta 64 bits con formatos de imágenes diversos como ser BMP, JPEG, GIF, TIFF, DICOM y FITS.

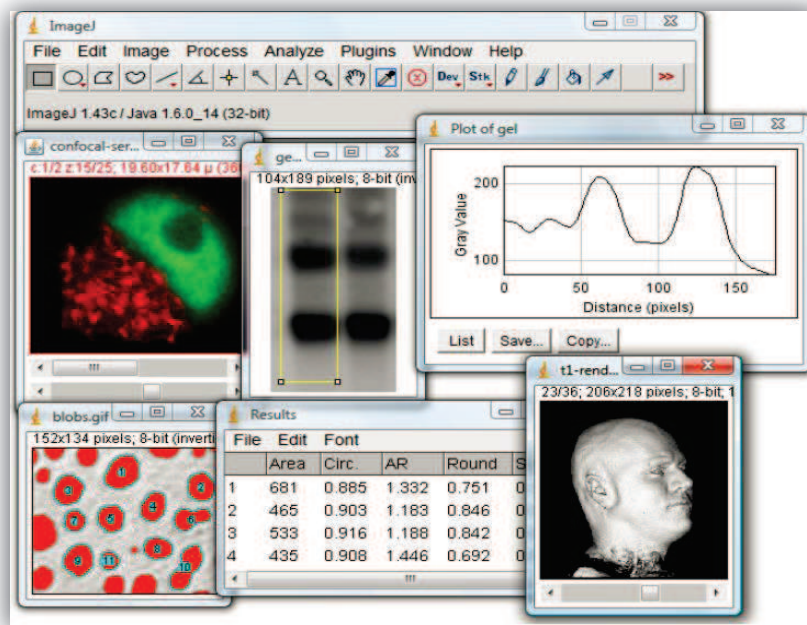


Figura 3. Ejemplo de visualización de las diversas ventanas en ImageJ que consumen parte de la memoria en uso del programa. Desde el menú principal es posible elegir la visualización de todas las ventanas o sólo la ventana principal.

2.2 Entorno de trabajo en ImageJ

La pantalla principal de ImageJ consta de una ventana en pequeño tamaño con la finalidad principal de lograr practicidad de visualización a la hora de manipular varias imágenes, tablas y

gráficas a la vez (figura 3). Sin embargo, toda esa cantidad de información con la que se trabajará se encuentra limitada por la cantidad de memoria RAM asignada al programa. Inicialmente, ImageJ está configurado para utilizar 256 MB pero puede ampliarse fácilmente según el volumen de información a utilizar (menú *Edit > Options > Memory & Threads...*). No obstante, ImageJ aconseja no exceder el 75% de la memoria RAM del sistema.

2.3 Programación en ImageJ

La interacción del usuario con ImageJ se realiza mediante menús y botones, como se pudo visualizar previamente. Para desarrollar tareas repetitivas o complejas cabe la posibilidad de utilizar las funciones de ImageJ como líneas de comando siendo el usuario el que escriba programas para ejecutarlas. Existen dos tipos de programas: macros y plugins. Los macros son programas que ejecutan líneas de comando asociadas a los menús y a funciones y rutinas propias de ImageJ mediante un intérprete de comandos incluido en ImageJ. En este contexto el usuario puede asimismo escribir funciones propias que pueden ser llamadas por macros u otras funciones. La condición para esto es que tanto el código fuente de las funciones del usuario como el conjunto de macros que las utilicen se encuentren en un mismo archivo. Los macros permiten implementar operaciones relativamente sofisticadas utilizando una sintaxis simple. Esta tarea se ve notablemente simplificada para los usuarios principiantes mediante la utilización de una herramienta de ImageJ que va escribiendo en tiempo real –en una ventana especial y en el lenguaje de macros– las operaciones que el usuario va realizando de manera interactiva en ImageJ (comando *Record...* del submenú *Macros* del menú *Plugins*). El sitio web de ImageJ proporciona una muy buena documentación relativa a la programación de macros y a la descripción y uso de las funciones nativas disponibles.

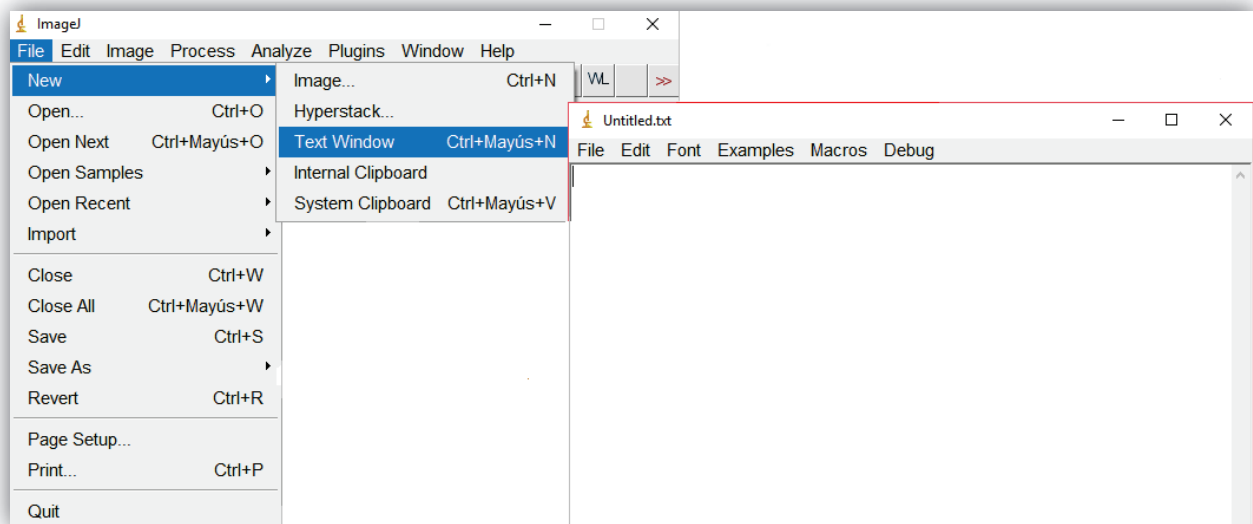


Figura 4. Ventana del editor (derecha) donde se escribe el código de programación y la forma de abrirla en ImageJ (izquierda).

Si bien los macros dan la posibilidad de programar tareas muy diversas de manera simple, operaciones con gran demanda computacional pueden estar penalizadas ya que ImageJ debe ir interpretando cada una de las líneas de comando. Los plugins son programas más poderosos y rápidos que los macros. Su lenguaje de programación es Java y pueden integrarse sin dificultad a ImageJ ya que éste es de código abierto. La gran mayoría de los comandos existentes en el menú del programa son plugins.

Cabe destacar que ImageJ incluye la posibilidad de abrir o crear macros en diversos lenguajes de programación promoviendo un beneficio para la utilización de aquellos documentos creados en otros programas de procesamiento de imágenes gracias a la presencia de un interpretador que establece comandos en común para decodificar esos macros grabados en lenguajes varios.

La escritura de macros puede ser realizada internamente mediante una ventana de edición específica en ImageJ. Esta ventana, el editor, contiene el espacio donde se escriben las líneas del código de programación a partir de funciones básicas que contiene el programa o de funciones más o menos complejas que bien puede crear el usuario (figura 4). Como se dijo previamente, estas funciones propias del software se encuentran detalladas en la página web de ImageJ así como también la explicación en profundidad de uso del programa.

En la sintaxis de programación en ImageJ se admiten variables locales (reconocidas sólo al interior de macros o funciones definidas por el usuario) o variables globales que son compartidas por el conjunto de macros y funciones definidos en un mismo archivo de programa. Una variable es un espacio de memoria que puede alojar tanto un valor numérico, una cadena de caracteres o una matriz de datos. La asignación interactiva de valores a las variables puede programarse a través de funciones intrínsecas que generan, de manera simple, útiles ventanas de diálogo con el usuario. Las variables locales pueden ser creadas tanto de manera explícita como de manera implícita a partir de una simple asignación de valor. Es decir que el tipo de una variable (tipo numérico, de cadena de caracteres, etc.) está determinado por la sentencia de asignación y, por lo tanto, el tipo puede cambiar dinámicamente a lo largo del programa. Las variables globales, por su parte, son variables que deben ser declaradas antes de la definición de los macros y funciones que las utilicen.

Una vez guardado el archivo que contiene el código fuente, la forma de tener disponible el conjunto de macros definidos en el mismo, es mediante un proceso de instalación en ImageJ. Esta operación se lleva a cabo a través del comando *Install...* del submenú *Macros* del menú *Plugins* ubicado en la ventana principal de ImageJ (figura 5).

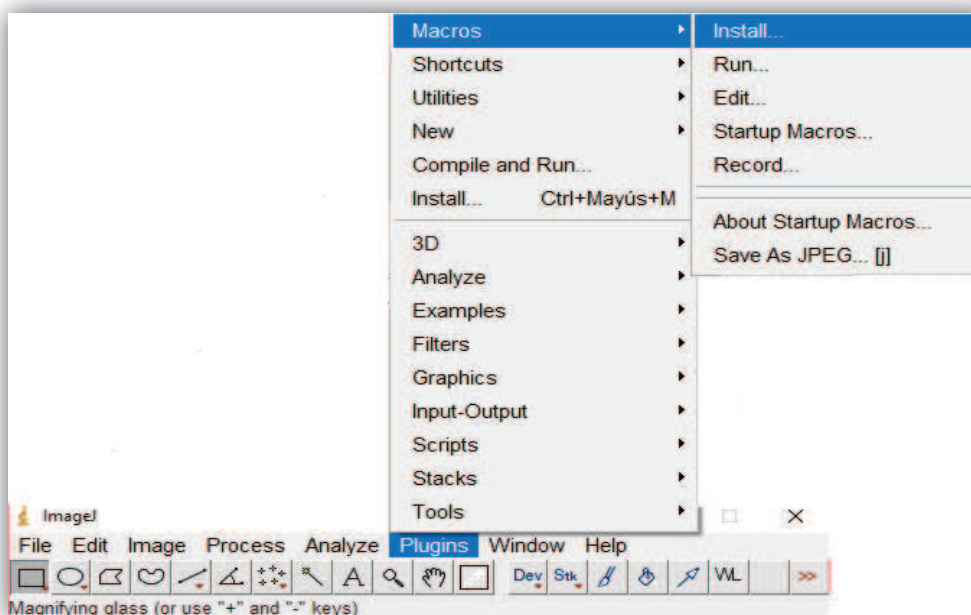


Figura 5. Pasos para la instalación de un macro desde la ventana principal de ImageJ.

3 FIJI: generalidades

FIJI (acrónimo inglés que indica que Fiji es simplemente ImageJ, *Fiji is just ImageJ*) es un programa de análisis de imágenes de código abierto basado en ImageJ pero con otro grado de sofisticación conteniendo multiplicidad de aplicaciones enfocadas a análisis de imágenes científicas en general. Nace en el año 2008 de la mano de Johannes Schindelin y Albert Cardona y se construyó básicamente para mejorar herramientas en el campo de las ciencias de la vida procesando y analizando información (Schindelin et al. 2012).

Al igual que ImageJ su descarga se lleva a cabo desde la página web de FIJI pudiendo utilizarse en cualquier sistema operativo. Fiji permite abrir y/o guardar los macros creados en una multiplicidad de lenguajes, incluso muchos más de los que permite la versión original de ImageJ. Algunos de ellos son GROOVY, PYTHON, JAVASCRIPT, RUBY y MATLAB.

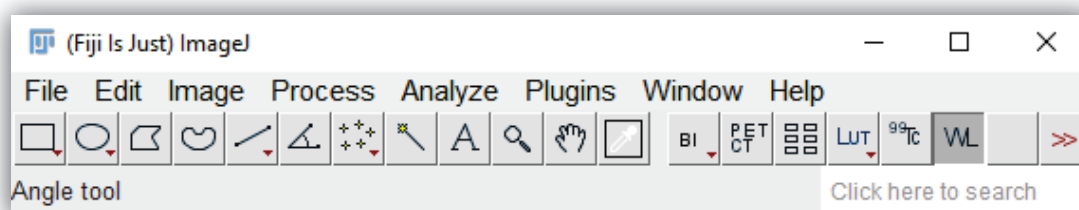


Figura 6. Ventana principal de Fiji.

Desde un punto de vista estético, la visibilidad de la ventana principal de Fiji es idéntica a la de ImageJ, salvo por la presencia de diversos plugins instalados y ubicados al final de la barra de herramientas previa actualización del programa (figura 6). En Fiji se mantienen muchas de las funciones que se realizaban en ImageJ, principalmente las funciones de apertura y manipulación de las imágenes pero con el adicional de nuevos macros y plugins otorgando herramientas nuevas para llevar a cabo análisis de imágenes científicas en general en varios campos como ser la biología, la genética, las ciencias de materiales, etc. Así, FIJI logra desarrollar múltiples componentes sin necesidad que el usuario tenga que descargar programas adicionales para ejecutar el análisis de la información.

Como mejoría en el programa se encuentra un actualizador automático, que se descubre al iniciar FIJI⁶, de forma manual o cuando existe alguna versión más moderna del programa. De esta manera, actualiza macros, plugins y componentes de plugins, entre otros. El programa cuenta con más de doscientos de ellos y, en caso de necesitarse un plugin determinado puede descargarse durante esta etapa de actualización (figura 7 y figura 8). Luego de la instalación de algún plugin en particular, debe reiniciarse FIJI para efectivamente terminar la etapa de incorporación del programa en el menú principal de FIJI y comenzar a trabajar con él.

⁶ <https://fiji.sc/>

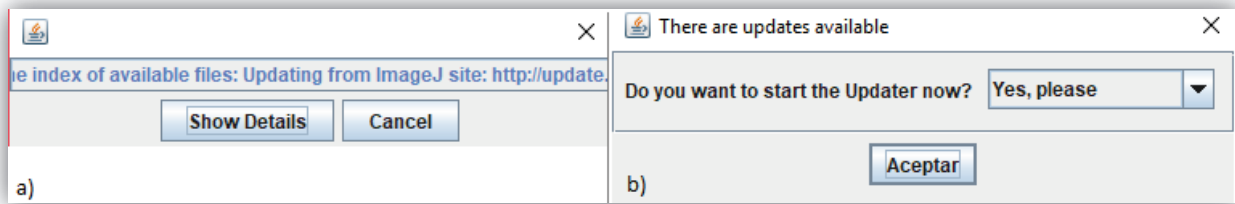


Figura 7. Ventana con el proceso de actualización del programa (a) y ventana de actualización que propone FIJI al iniciar el programa (b).

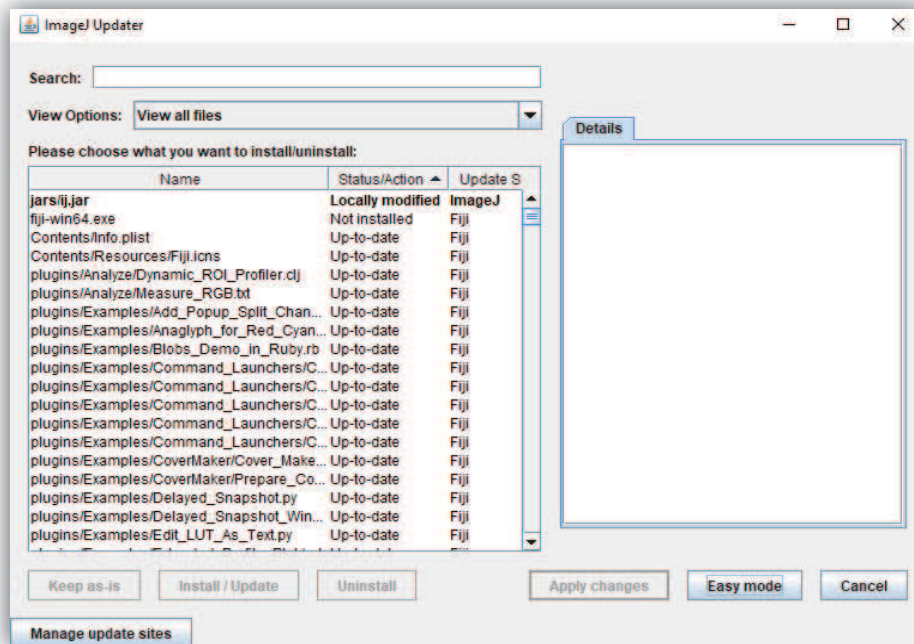


Figura 8. Ventana de FIJI que contiene un buscador donde se facilita la localización de algún plugin en particular, accediendo desde la solapa *Help>Update...>Advanced mode*.

Dentro de Fiji existen varios plugins creados para funciones diversas, por ejemplo, en el ambiente astronómico y biocientífico. Dentro de ellos, se encuentra un plugin denominado PET-CT Viewer, creado por Llan Tal y Salim Kanoun, que fue fundamental para el desarrollo de este trabajo de tesis y que describimos a continuación.

4 PET-CT Viewer

4.1 Descarga y utilización

PET-CT Viewer (Tai and Kanoun 2019) es un programa desarrollado en el hospital Beth Israel en Boston para leer y procesar principalmente imágenes en la modalidad diagnóstica de PET-CT (figura 9) aunque también acepta imágenes de tomografía por emisión de fotones (o SPECT por *single-photon emission tomography*) y de resonancia magnética (o MRI *magnetic resonance imaging*). La creación de esta herramienta radica, según sus autores, en tres realidades que se viven en la actualidad en los servicios de PET-CT, primariamente, una cuestión de tiempo en uso del software comercial para uso clínico que limita su utilización por toda la comunidad profesional que trabaje en la institución. Por otro lado, lo costosa que es la herramienta habitualmente empleada y, por último, la realmente escasa cantidad de información cuantitativa que puede obtenerse al analizar las imágenes.

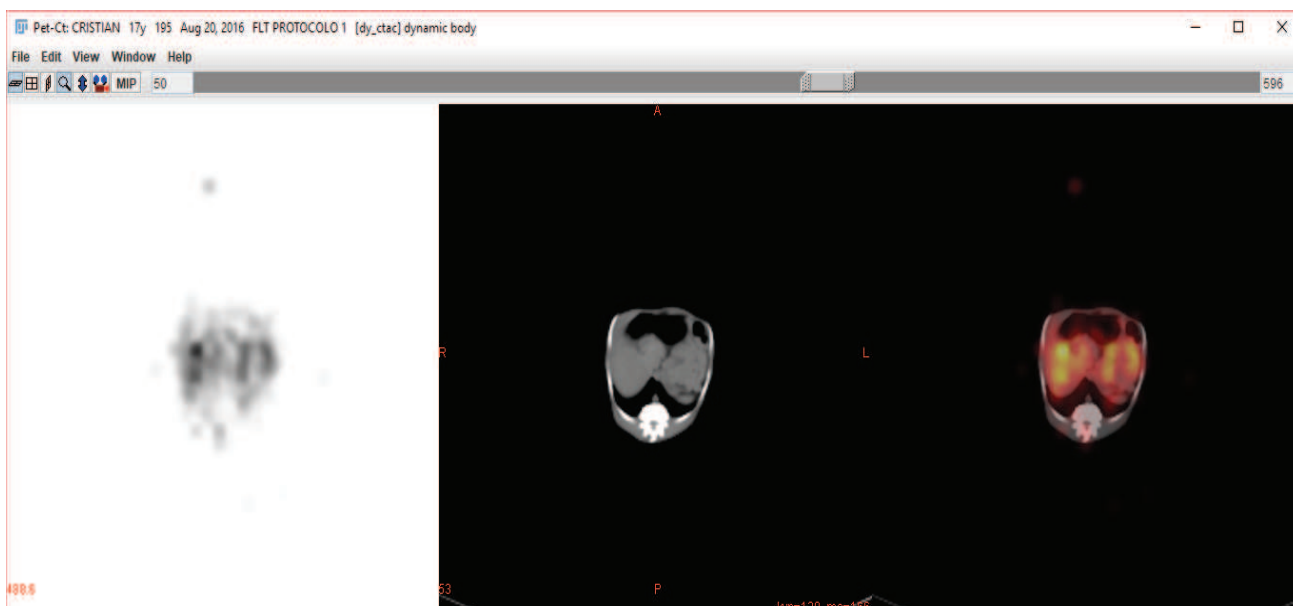


Figura 9. Pantalla principal del plugin PET-CT Viewer, donde se visualiza las imágenes de los estudios PET y TC por separado y la fusión correspondiente.

Dicho programa contiene múltiples funciones tendientes a conseguir datos relevantes acerca de las imágenes de PET-CT. Sus funciones tratan desde la obtención de valores de SUV a nivel de vóxel, determinación de distancias entre puntos, trazado de regiones de interés con su posterior análisis cuantitativo de varias variables importantes hasta la posibilidad de documentar esa información.

En primera instancia debe descargarse la herramienta desde la ventana principal de FIJI para proceder a la visualización de la fusión del estudio PET con la CT, a través de la opción de actualización en el menú de ayuda (figura 10). Una vez descargado el plugin, debe reiniciarse FIJI para actualizar también el menú principal del mismo donde se visualizará en uno de los iconos, el que corresponde al de PET-CT, y así, se permite automáticamente el acceso al mismo cuando se decida trabajar con él.

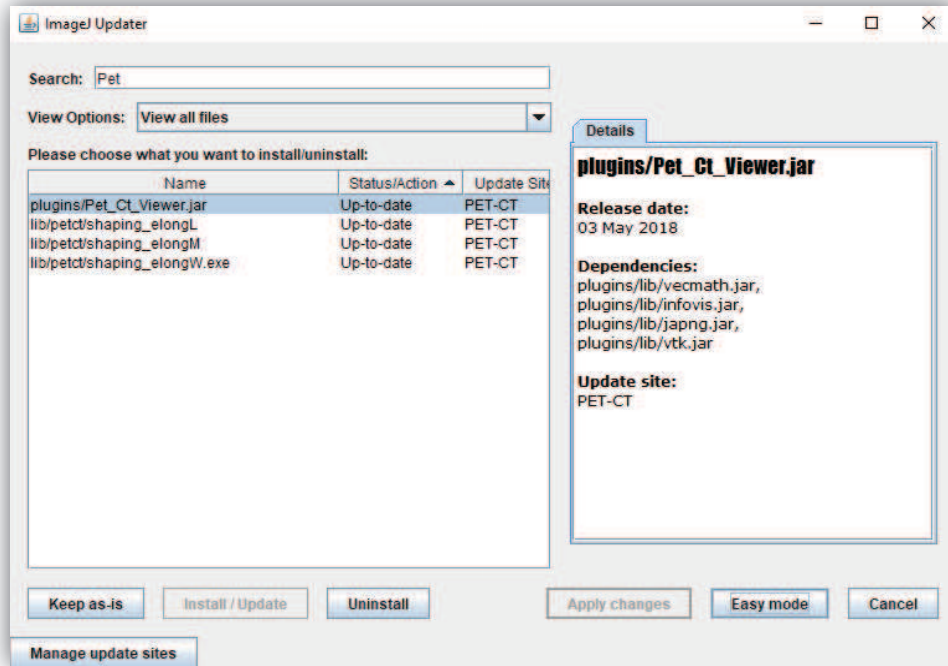


Figura 10. Ventana principal de actualización del software donde se inicia la búsqueda de algún plugin en particular, en este caso, de PET-CT Viewer.

Por otro lado, deben seleccionarse los archivos del estudio de PET-CT con los cuales se va a trabajar. Esto se realiza desde una opción presente en el menú contiguo al icono de PET-CT (figura 11).

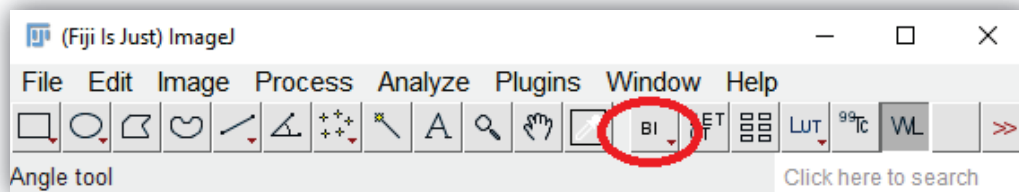


Figura 11. El icono BI posibilita la selección de los archivos del estudio de PET-CT.

Bastará con elegir por única vez los archivos a analizar para lograr visualizar la ventana de BI siempre que se inicie el programa Fiji, sin necesidad de acceder al botón BI en dicho menú. Dicha ventana muestra la última serie de archivos con los que se ha trabajado previamente en forma de carpetas, aquí se le pide al usuario que seleccione la carpeta con el estudio DICOM de PET-CT con el que se va a trabajar. Cabe destacar que desde la solapa *setup* se puede configurar hasta doce carpetas desde donde leer automáticamente archivos DICOM de los estudios a analizar (figura 12).

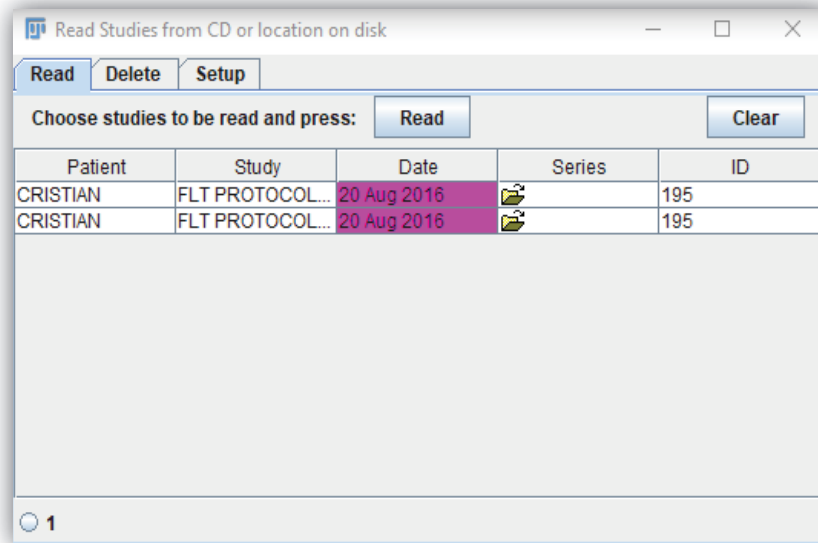


Figura 12. Ventana de BI donde se seleccionará el archivo PET-CT con el cual se trabajará.

Una vez seleccionada una de las carpetas, se inicia la apertura de todas las series de imágenes que incluyen a dicho estudio, finalizando con otra ventana que lista dichas series. De aquí se procede a seleccionar aquellas que se desea mostrar de manera fusionada, en nuestro caso, las imágenes PET y las imágenes de la TC (figura 13).

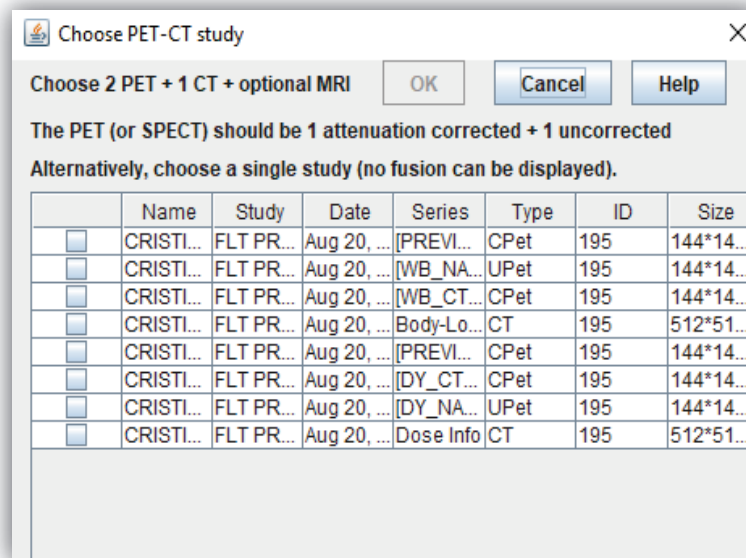


Figura 13. Ventana de selección para fusionar las imágenes de PET-CT.

Una vez seleccionados los archivos se inicia un visualizador general del plugin dividido en tres espacios pudiendo observar las imágenes PET, las imágenes tomográficas de rayos X y la fusión de ambas en la misma pantalla. Aquí, se puede modificar contraste, tamaño de las imágenes, plano de corte a visualizar, guardar alguna imagen relevante y hasta marcar ROIs para realizar un análisis completo relacionado a diversos valores SUV en dichas zonas. Es preciso aclarar que PET-CT Viewer detalla las regiones de interés que se deseen trazar por medio de ROIs donde el operador puede otorgarle nombres a cada una de ellas. Si se requiere delimitar VOIs (volúmenes

de interés) surgidos del trazo de varias ROIs, el programa engloba los resultados dentro de un ítem denominado *sum* (ver figura 17). Así, a la hora de graficar diversas magnitudes puede optarse por alguna ROI en particular o directamente sobre los VOIs a través del ítem *total (axial)*, que se visualizarán a la hora de graficar los resultados.

Para llevar a cabo el análisis de información en ROIs desde el menú principal se selecciona *Edit>Brown fat, ROIs*; de esta manera se abre la ventana *Brown fat volume* la cual contiene varias solapas (figura 14). A continuación detallaremos la solapa ROI ya que es la más relevante para este trabajo.

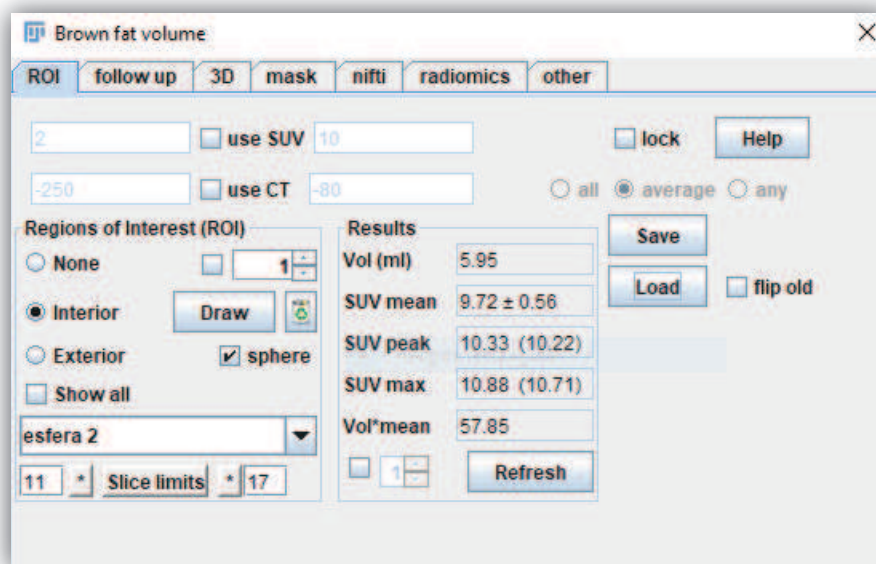


Figura 14. Ventana interactiva para el trazado de las ROIs.

La solapa ROI de la ventana *Brown fat volume* se utiliza principalmente para delimitar ROIs, cuyos trazos dependen de las intenciones del usuario. Existen diversas maneras de definir una ROI, por un lado, accediendo al botón *sphere* se dibuja una esfera (más estrictamente se trata de un elipsoide) constituida por el trazado de varias elipses donde el usuario solamente interviene en la delimitación de la elipse central dibujando únicamente dos puntos sobre determinado corte y visualizando las demás elipses de manera automática por parte del programa (figura 15). Aquí, es posible elegir límites de cortes sobre las imágenes que definirán el tamaño y cantidad de las elipses que componen al elipsoide. Cabe mencionar que la elipse trazada presenta el mayor diámetro y el tamaño de las elipses contiguas disminuye desde el centro hasta alcanzar los cortes extremos. En este caso, cada una de las elipses constituye una ROI y la esfera completa el VOI. La otra oportunidad que presenta el software es delimitar tantos puntos como sean necesarios para conformar la ROI deseada por el operador (ROI poligonal) donde la separación mínima entre puntos depende del tamaño del pixel de la imagen (figura 16).

Adicionalmente a la delimitación geométrica de las ROIs es posible aplicar, a los vóxeles que las componen, reglas adicionales de selección para excluirlos del análisis. Estas reglas están basadas en la definición independiente de un intervalo de aceptación para valores de SUV y de un intervalo de aceptación de valores Hounsfield (HU). De esta manera es posible aplicar un criterio de medicina nuclear y un criterio anatómico a la selección de vóxeles que están incluidos en una dada ROI.

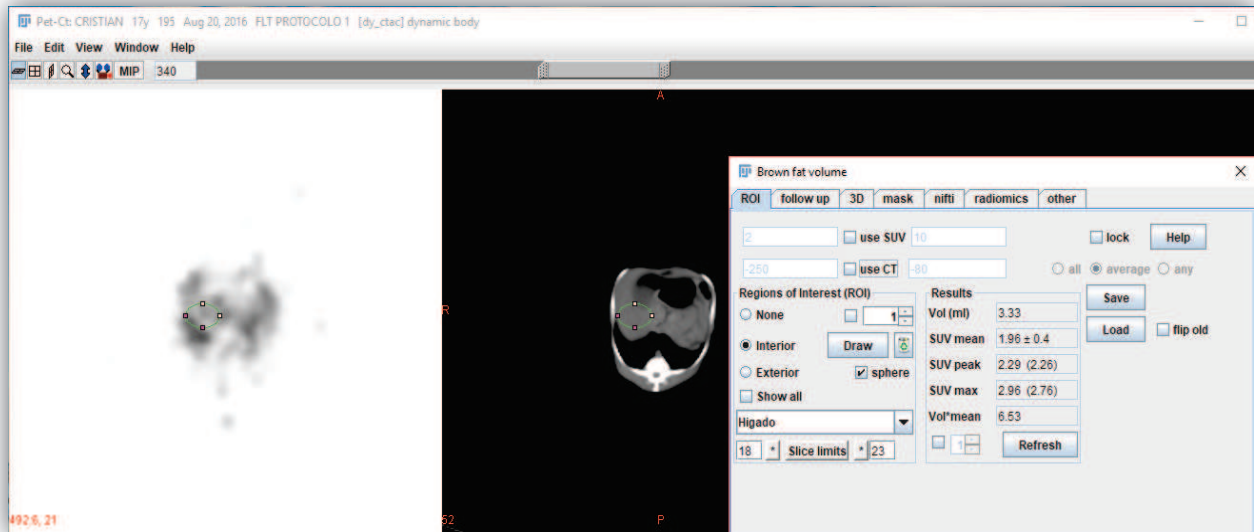


Figura 15. Visualización de imágenes y delimitación de una ROI en forma de elipse.

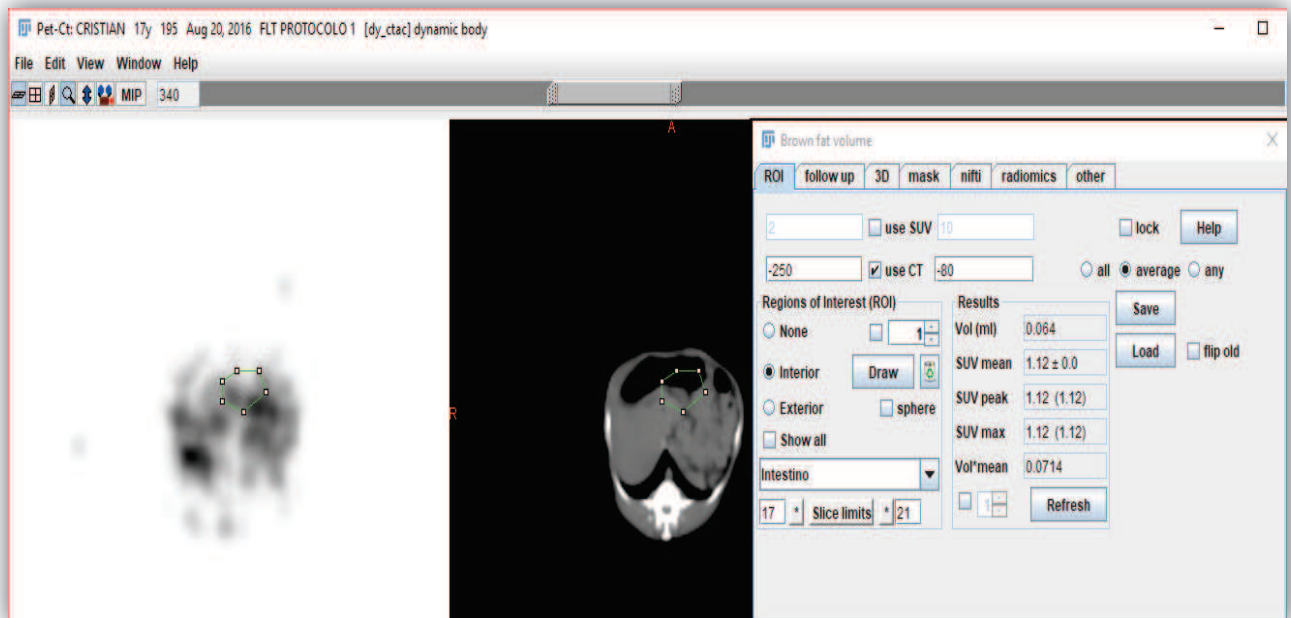


Figura 16. Visualización de imágenes y delimitación de una región poligonal.

En la ventana hay una tabla que, al finalizar el trazado de una ROI, muestra diversos resultados obtenidos sobre la misma. Esta tabla resumen da a conocer algunos valores del análisis para ciertas magnitudes asociadas al SUV. La tabla presenta los resultados para una ROI determinada, sin embargo, el *SUV Peak* y *SUV max* contienen un par de resultados cada uno. El primer valor corresponde al valor general de la ROI, la cual puede ser en realidad un VOI en el caso de estar constituida por varios cortes tomográficos. El segundo valor, el que se encuentra entre paréntesis, indica un promedio sobre los valores obtenidos sobre cada uno de los cortes del VOI.

Este análisis se desarrollará posteriormente en detalle incluyendo distintas variables adicionales que complementan el estudio de las zonas delimitadas. Los resultados se pueden guardar a través del botón *Save*. Por otra parte, puede cargarse archivos grabados por este programa visualizando

en pantalla las coordenadas de esas ROIs trazadas junto a los datos cuantitativos obtenidos de ese análisis utilizando el botón *Load*.

4.2 Archivos separados por comas (CSV): definición y estructura según PET-CT Viewer

PET-CT Viewer optó por archivos CSV para alojar aquella información obtenida del análisis de los trazados de interés de relevancia para el usuario, este tipo de archivo almacena los datos en forma de texto plano que puede asimilarse fácilmente a una estructura de tabla donde la coma sirve como delimitador para separar datos desde una columna a la otra. Las filas se separan por un carácter de salto de línea o de retorno de carro. Las tablas benefician la visualización de la extensa información que es obtenida. Al abrirse este tipo de archivos en programas para el manejo de hojas de cálculo, es posible llevar a cabo fácilmente operaciones entre los datos.

En el caso de estudios dinámicos (es decir correspondientes a series temporales) los datos se organizan en bloques sucesivos, uno para cada adquisición. La separación entre bloques se realiza mediante una línea que sólo contiene el texto “*Phase n*”, donde *n* es un número que identifica secuencialmente a la adquisición. En la figura 17 se muestra un fragmento del contenido de este tipo de archivos donde se visualizan magnitudes consideradas en PET-CT Viewer.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Label	ROI	type	Vol(ml)	Vol*mean	SUVMean	SD	SUVPeak	MeanWahl	SUVqPeak	MeanQpet	SUVMax	MeanMax	HU
2	Prostata1	1	1	2.18	0.05	0.02	0.02	0.06	0.06	0.06	0.06	0.07	0.07	0.0±0.0
3	Prostata2	2	1	2.56	0.05	0.02	0.01	0.03	0.03	0.03	0.03	0.06	0.06	0.0±0.0
4	Prostata3	3	1	1.47	0.04	0.03	0.01	0.03	0.03	0.04	0.04	0.05	0.05	0.0±0.0
5	Prostata4	4	1	1.73	0.08	0.05	0.03	0.08	0.08	0.09	0.09	0.1	0.1	0.0±0.0
6	Prostata5	5	1	3.26	0.1	0.03	0.02	0.05	0.05	0.06	0.06	0.07	0.07	0.0±0.0
7	Prostata6	6	1	4.54	0.2	0.04	0.04	0.15	0.15	0.17	0.17	0.21	0.21	0.0±0.0
8	Prostata7	7	1	4.61	0.14	0.03	0.03	0.09	0.09	0.12	0.12	0.13	0.13	0.0±0.0
9	Prostata8	8	1	4.74	0.12	0.03	0.02	0.07	0.07	0.07	0.07	0.09	0.09	0.0±0.0
10	Prostata9	9	1	2.37	0.09	0.04	0.02	0.08	0.08	0.08	0.08	0.11	0.11	0.0±0.0
11	axial	sum	1	27.46	0.88	0.03	0.03	0.13	0.07	0.17	0.08	0.21	0.1	0.0±0.0
12	Phase 2													
13	Prostata1	1	1	2.18	0.89	0.41	0.1	0.59	0.59	0.59	0.59	0.68	0.68	0.0±0.0
14	Prostata2	2	1	2.56	1.35	0.53	0.19	0.81	0.81	0.89	0.89	1.02	1.02	0.0±0.0
15	Prostata3	3	1	1.47	0.87	0.59	0.19	0.79	0.79	0.84	0.84	0.94	0.94	0.0±0.0
16	Prostata4	4	1	1.73	1.2	0.69	0.18	0.91	0.91	0.93	0.93	1.08	1.08	0.0±0.0
17	Prostata5	5	1	3.26	2.55	0.78	0.3	1.13	1.13	1.25	1.25	1.5	1.5	0.0±0.0
18	Prostata6	6	1	4.54	4.59	1.01	0.32	1.53	1.53	1.63	1.63	1.75	1.75	0.0±0.0
19	Prostata7	7	1	4.61	4.8	1.04	0.3	1.56	1.56	1.62	1.62	1.72	1.72	0.0±0.0
20	Prostata8	8	1	4.74	5.44	1.15	0.36	1.65	1.65	1.71	1.71	1.85	1.85	0.0±0.0
21	Prostata9	9	1	2.37	2.59	1.09	0.46	1.79	1.79	2.01	2.01	2.09	2.09	0.0±0.0
22	axial	sum	1	27.46	24.27	0.88	0.39	1.73	1.2	2.01	1.27	2.09	1.4	0.0±0.0

Figura 17. Archivo CSV visualizado en una planilla de cálculo donde figuran datos de todas las variables calculadas por PET-CT Viewer para cada ROI y para cada instante de tiempo.

La primera columna corresponde a *Label* (etiqueta), donde se sitúan los nombres de cada una de las ROIs que se trazaron. Hay ciertos nombres dados automáticamente por el programa: *axial*, *coronal* y *sagital*. Con ellos, el programa indica el nombre del tipo de plano utilizado para definir la última ROI trazada y la fila correspondiente contiene información global acerca de todas las ROIs que se trazaron sobre dicho tipo de plano.

La segunda columna corresponde a las *ROIs* donde se las enumera de forma ascendente respetando el orden de trazado de las mismas y en el renglón siguiente a la numeración de la última ROI trazada se encuentra el elemento *sum*, para indicar, como se dijo precedentemente, que

en el análisis que se realiza en el resto de las columnas considera al conjunto de las ROIs definidas según el plano mencionado en la primera columna.

La tercera columna se relaciona al *type* (tipo) pudiéndose completar con valores del 1 al 3, axial, coronal o sagital respectivamente, según cuál haya sido el plano de corte elegido para el trazado.

A partir de la cuarta columna se encuentran aquellos datos referidos a las diversas variables de información elocuente para cada ROI.

Las magnitudes calculadas en PET-CT Viewer son:

- ✓ *Volume* (volumen): corresponde al volumen para cada ROI trazada expresado en ml.
- ✓ *Volume*Mean* (volumen*promedio): surge del producto entre el volumen y el SUV promedio.
- ✓ *SUV Mean* (SUV promedio): corresponde al promedio del valor SUV de los pixeles hallados en cada ROI (ver expresión (3) más abajo).
- ✓ *SD* (desviación estándar): constituye el desvío estándar del SUV promedio.
- ✓ *SUV Peak* (*SUV pico*): corresponde al promedio de SUV tomado alrededor de 1 cm^3 de volumen alrededor del pixel que contiene el máximo valor del SUV (ver más abajo *SUVmax*) dentro de la ROI considerada.
- ✓ *Mean Wahl* (media de Wahl) constituye el valor promedio de los *SUV pico* obtenidos en cada corte que componen el VOI.
- ✓ *SUV Q Peak* (*SUV q pico*): corresponde al promedio del SUV máximo junto a otros 3 valores máximos alrededor de un 1 cm^3 de volumen para cada ROI trazada.
- ✓ *Mean Q PET* (Q PET medio) consiste en el promedio de los *SUV q pico* en todos los cortes que incluyen al VOI delimitado.
- ✓ *SUV max* (SUV máximo): corresponde al valor SUV más alto sobre la ROI delimitada (ver expresión (2) más abajo).
- ✓ *Mean max* (máximo medio): corresponde al promedio de todos los valores SUV máximo de cada ROI para todo el VOI.
- ✓ *HU*: en el caso que el usuario haya decidido filtrar los vóxeles a analizar mediante un rango de valores de Hounsfield (ver figura 14), esta columna muestra el promedio de dichos valores (que también hayan pasado el filtro para el rango de SUV) con su respectivo desvío estándar.

Al final del archivo CSV se puede apreciar otro grupo de datos, entre los que se destacan nombre del paciente, número de identificación del mismo, cantidad de ROIs trazadas, sus nombres y coordenadas del trazado, además de la cantidad de puntos utilizados para la delimitación de las ROIs, valores máximos y mínimos de SUV y HU utilizados para filtrar los datos (ver figura 14) y si se utilizaron o no para la obtención de los datos en este archivo (figura 18). De esta manera, el almacenamiento en el archivo de las coordenadas de las ROIs trazadas facilita la ubicación de las mismas en las imágenes de PET-CT en PET-CT Viewer, por si se desea cargar ROIs que se encuentran ya trazadas.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
479	intestino	3	1	2.88	6.72	2.33	0.44	2.81	2.59	3.21	2.79	3.41	3.16	27.53 ± 18.36
480	axial	sum		4.03	8.59	2.13	0.62	2.81	2.2	3.21	2.34	3.41	2.7	30.11 ± 16.22
481	Phase 97													
482	corazon	1	1	0.26	0.18	0.7	0.05	0.72	0.72	0.72	0.72	0.75	0.75	32.75 ± 3.77
483	higado	2	1	0.9	2.01	2.25	1.05	3.49	3.49	3.62	3.62	4.56	4.56	37.64 ± 5.0
484	intestino	3	1	2.88	7.25	2.52	0.44	2.85	2.72	3.26	2.96	3.46	3.28	27.53 ± 18.36
485	axial	sum		4.03	9.44	2.34	0.75	3.49	2.47	3.62	2.64	4.56	3.03	30.11 ± 16.22
486	Phase 98													
487	corazon	1	1	0.26	0.18	0.69	0.06	0.69	0.69	0.69	0.69	0.75	0.75	32.75 ± 3.77
488	higado	2	1	0.9	2.2	2.46	1.04	3.62	3.62	3.81	3.81	4.73	4.73	37.64 ± 5.0
489	intestino	3	1	2.88	6.94	2.41	0.43	2.92	2.69	3.26	2.88	3.38	3.14	27.53 ± 18.36
490	axial	sum		4.03	9.32	2.31	0.73	3.62	2.47	3.81	2.63	4.73	2.98	30.11 ± 16.22
491	Phase 99													
492	corazon	1	1	0.26	0.17	0.68	0.04	0.7	0.7	0.7	0.7	0.72	0.72	32.75 ± 3.77
493	higado	2	1	0.9	2.48	2.77	1.4	4.36	4.36	4.58	4.58	5.84	5.84	37.64 ± 5.0
494	intestino	3	1	2.88	7.38	2.56	0.58	3.24	3.04	3.55	3.31	3.66	3.55	27.53 ± 18.36
495	axial	sum		4.03	10.04	2.49	0.94	4.36	2.84	4.58	3.04	5.84	3.44	30.11 ± 16.22
496	CRISTIAN	Aug 20_2016	SUL =	1.16546	time(sec) =	254							patient ID =	195
497														
498	Number of ROIs = 3													
499	corazon	1	36	36	num points	71 68	72 70	74 67						
500	higado	1	25	25	num points	74 72	68 67	73 67						
501	intestino	11	12	16	num points	72 70	75 70	72 68						
502	SUVlo	SUVhi	CTlo	CThi	useSUV	useCT	CtRadio							
503	2	10	-100	80	0	1	1							

Figura 18. Tramo final del archivo CSV que muestra otros datos de las regiones trazadas.

5 SUV (valor estandarizado de captación)

El SUV es una magnitud de utilidad que apunta a proveer una cuantificación de imágenes de PET-CT asociado al metabolismo del ^{18}F -FDG con el objetivo de diferenciar entre tejido tumoral y sano. Esta medición indica una relación entre la concentración de la actividad del radiofármaco y la actividad que se inyectó de acuerdo al peso del paciente.

$$SUV = \frac{\text{Concentración de actividad dentro del tejido [MBq/ml]}}{\frac{\text{Actividad inyectada corregida por decaimiento [MBq]}}{\text{Peso del paciente [g]}}} \quad (1)$$

Así definido, el SUV puede también aplicarse como una medida de captación relativa de otros radiofármacos más allá de ^{18}F -FDG perdiendo en este caso, por supuesto, su vínculo con el metabolismo de la glucosa. En la expresión (1) se puede apreciar que el SUV contiene unidades de g/ml. pero teniendo en cuenta que el cuerpo humano tiene un contenido mayoritariamente de agua y la densidad del agua es de 1 g/ml equivaldría a dejar a la magnitud sin unidades.

Sin embargo, para considerar la exactitud de la información cuantitativa obtenida de las regiones trazadas debe prestarse especial atención a diversas condiciones, tanto al inicio del estudio de PET-CT como una vez reconstruidas las imágenes y comenzada la etapa de delimitación de las ROIs. Los factores, corresponden a los parámetros técnicos, físicos y biológicos durante la ejecución del estudio (ver por ejemplo Ferreyra 2011 o Spinelli 2018). Los técnicos y físicos corresponden a parámetros propios de adquisición del estudio de PET-CT, es decir, tamaño de la matriz de adquisición, grosor de corte elegido, calibración del equipo de PET-CT así como también la aplicación de algoritmos de corrección y desarrollo de la normalización de los datos previo a la etapa de reconstrucción de las imágenes. Existe un efecto originado por la resolución espacial del sistema conocido como volumen parcial que promueve la degradación de las imágenes alterando los valores obtenidos. La magnitud de tal degradación depende de factores tales como el tamaño de la lesión relativo a la resolución espacial y a la relación de concentración entre la lesión y el tejido circundante para el trazo de las ROIs. Por otra parte, dentro de los factores biológicos, se encuentran el peso y contextura del paciente, la concentración de glucosa en sangre, que pudiese competir con la concentración de ^{18}F -FDG inyectada y movimientos por parte del paciente durante la etapa de adquisición de las imágenes.

5.1 SUV máximo y medio (*SUVmax* y *SUVmedio*)

Los médicos especialistas en medicina nuclear informan los estudios de PET-CT no sólo desde la componente visual de las imágenes dando información cualitativa de la captación sino también mencionando valores relevantes cosechados al trazar y analizar ROIs sobre zonas tisulares de relevancia. Dichas variables corresponden habitualmente a valores de *SUVmax* y *SUVmedio* reflejados en las expresiones (2) y (3):

$$SUV_{max} = \frac{\text{Valor máximo entre todos los píxeles que forman parte de la ROI [MBq/ml]}}{\frac{\text{Actividad corregida por decaimiento al comienzo del estudio PET [MBq]}}{\text{Peso del paciente [g]}}} \quad (2)$$

$$SUV_{medio} = \frac{\text{Valor promedio de todos los píxeles que forman parte de la ROI [MBq/ml]}}{\frac{\text{Actividad corregida por decaimiento al comienzo del estudio PET [MBq]}}{\text{Peso del paciente [g]}}} \quad (3)$$

El *SUV_{max}* corresponde al valor máximo dentro de todos los píxeles que abarcan la ROI delimitada siendo susceptible a las fluctuaciones estadísticas pero sin depender fuertemente de la precisión con la que se definan los límites de la ROI.

Por otra parte, el *SUV_{medio}* constituye un promedio de todos los valores de los píxeles intervinientes en la ROI. Así, esta medida disminuye las fluctuaciones estadísticas al considerar varios píxeles provocando un suavizado del ruido. No obstante, se considera una medición dependiente del operador al variar su valor según la ubicación de la ROI mayoritariamente por contribución del efecto de volumen parcial.

6 Herramienta de análisis desarrollada en este trabajo

Antes de proceder a detallar la manera en la que se desarrolló la herramienta vamos a especificar, en primer lugar, la estructura final del archivo de resultados originado por PET-CT Viewer debido a que funciona como dato de entrada en el programa desarrollado. Dicho archivo es el que se describió previamente en la sección 4.2. Como el archivo CSV no contiene toda la información necesaria para realizar un análisis temporal de la captación, fue necesario extraer estos datos directamente de los archivos DICOM del estudio. Por lo tanto, se describen también las etiquetas DICOM (ver NEMA 2018) consideradas en el presente trabajo.

6.1 Utilización del archivo CSV en el código creado

Los archivos CSV generados por PET-CT Viewer contienen información que no es necesaria para la herramienta que hemos desarrollado. Asimismo contienen información redundante o agrupada en bloques tales que, si bien estos bloques resultan útiles para su visualización en una planilla de cálculo, no son cómodos para una lectura automática desde un programa. Con el fin de sortear estos inconvenientes, aprovechando lo mejor posible las funciones incluidas en el lenguaje de programación de macros de Fiji para así optimizar el código, una parte del mismo está destinada al manejo y reorganización de los datos del archivo mediante la manipulación de cadenas de caracteres (recordemos que los archivos CSV son archivos de texto). De esta manera, se procedió a la eliminación de datos repetidos y a situar valores de acuerdo a las diversas magnitudes en cada una de las fases temporales. Asimismo se procedió a desglosar los datos presentes en la columna HU; se generó una variable asociada al HU promedio y otra a su desvío estándar aprovechando para ello la presencia del carácter \pm en el archivo (ver figura 17). La información de la sección final del archivo (figura 18) no resulta de relevancia para el desarrollo de la herramienta de análisis creada por lo que se decidió no procesarla.

6.2 Formato estándar DICOM

DICOM (ver NEMA 2018) es el formato estándar de almacenamiento, impresión y transmisión entre sistemas de información en el campo de las imágenes médicas. Respecto del almacenamiento de los datos, establece la estructura y organización de los archivos y carpetas que contienen la información relacionada al estudio. Estos archivos no sólo cuentan con los datos de las imágenes obtenidas sino también con información referida al paciente, al equipo formador de las imágenes así como también a parámetros de adquisición, entre otros. De esta manera, podemos decir que un archivo DICOM se compone de un cuerpo con los valores asociados a cada píxel de la imagen y de un encabezado que contiene el resto de los datos que se identifican mediante etiquetas (*tags*) predefinidas.

Con el pasar de los años el estándar ha sufrido actualizaciones tendientes a mejorar principalmente los sistemas de comunicación y transferencia de información. Estas modificaciones son llevadas a cabo por la asociación NEMA a través de normas de calidad y reglas para efectivizar un desarrollo continuo.

Attribute Name	Tag	Type	Attribute Description
Acquisition Date	(0008,0022)	2	The date the acquisition of data that resulted in this image started. See C.8.9.4.1.4 for specialization.
Acquisition Time	(0008,0032)	2	The time the acquisition of data that resulted in this image started. See C.8.9.4.1.4 for specialization.
Actual Frame Duration	(0018,1242)	2	Elapsed time of the data acquisition for this image, in msec. See C.8.9.4.1.6 for explanation.
Nominal Interval	(0018,1062)	3	Average duration of accepted beats, in msec, of the R-R interval.
Frame Reference Time	(0054,1300)	1	The time that the pixel values in the image occurred. Frame Reference Time is the offset, in msec, from the Series reference time. See explanation in C.8.9.4.1.5.
Trigger Time	(0018,1060)	1C	Time interval, in msec, from the start of the trigger to the beginning of data acquisition for this image. Required if Series Type (0054,1000), Value 1 is GATED.
Frame Time	(0018,1063)	1C	Nominal duration per individual frame, in msec. Required if Series Type (0054,1000), Value 1 is GATED. See C.8.9.4.1.3 for explanation.

Figura 19. Ejemplos de algunos campos DICOM, con sus respectivas etiquetas (*tags*) y una breve descripción de los atributos DICOM. Los que fueron considerados en este trabajo en relación a variables temporales se encuentran resaltados en color rojo (fragmento extraído de NEMA 2018, parte 3).

Para la finalidad de este trabajo, se ha estudiado exhaustivamente los campos DICOM que fueron de utilidad para el desarrollo del código de cálculo. El campo *frame reference time* (tiempo de referencia de los grupos de imágenes) fue el elegido como variable para el trazado de la cinética de captación porque, según establece DICOM, es el instante de referencia donde se produce la corrección de las imágenes por decaimiento en el cual se realiza una compensación de la actividad dependiendo de la vida media del radioisótopo utilizado, dicha compensación se relaciona con la mitad de la duración de los grupos de imágenes (*Actual frame duration*). En la figura 19 se muestran en detalle atributos, *tags* y descripción de diversos campos del estándar DICOM resaltando aquellos utilizados en la concreción de la herramienta creada.

6.3 Descripción del código

Con la finalidad de lograr practicidad se divide el código en dos partes mediante sendos macros que funcionan de manera independiente pero interrelacionados. El primer macro tiene como objetivo facilitar al usuario la exploración de carpetas DICOM con la finalidad de hallar el estudio de interés. Para esto el macro genera una tabla que permite visualizar el nombre de la carpeta y el nombre de la serie asociada, a medida que el usuario selecciona una imagen cualquiera de cada una de las carpetas que componen el estudio de PET-CT. Así, la acción se repite hasta que el usuario reconoce el archivo en cuestión.

Cabe destacar que la tabla no se elimina al finalizar el macro sino que continua activa permitiendo al usuario visualizar las carpetas que se han seleccionado y, en particular, la carpeta requerida que se utilizará en el próximo macro.

El siguiente diagrama de flujo esquematiza los diversos pasos que se suceden en este macro que finaliza con la obtención del nombre de la carpeta que engloba la serie de imágenes dinámicas (figura 20).

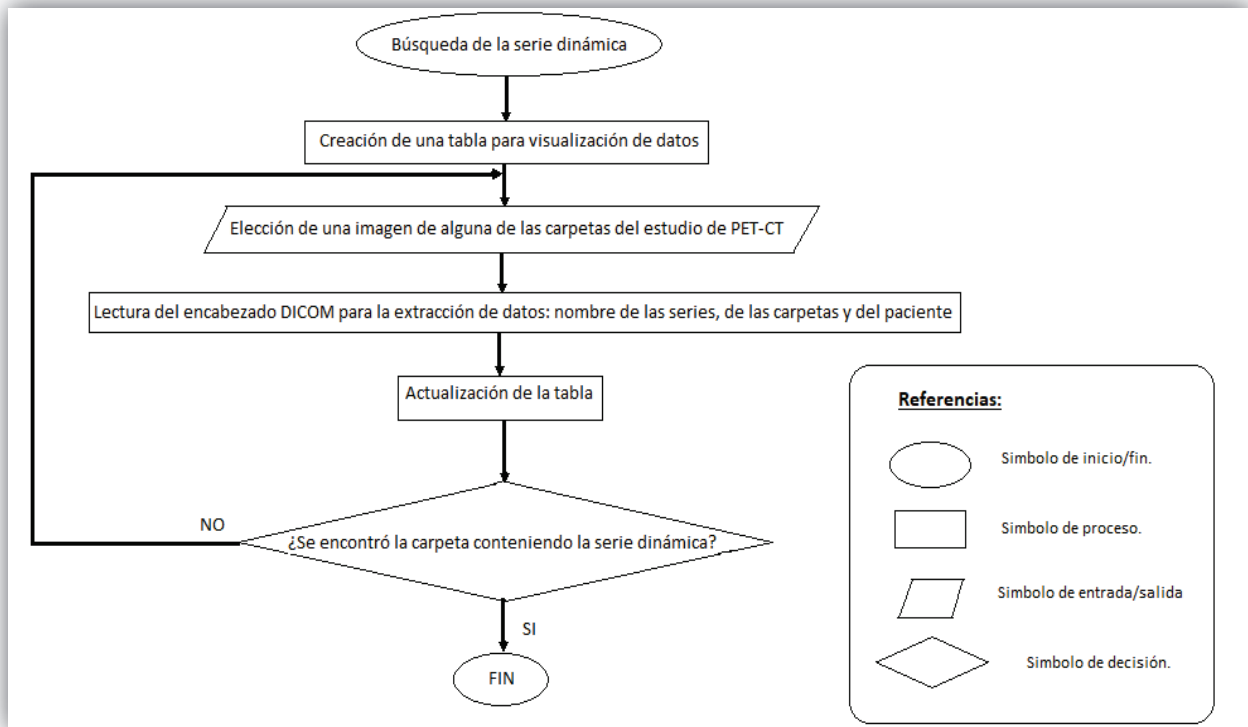


Figura 20. Esquema del diagrama de flujo para el desarrollo del macro 1.

El segundo macro requiere principalmente de dos acciones por parte del usuario. Por un lado, se debe seleccionar una de las imágenes de la carpeta de interés (en particular, la encontrada en el macro anterior) y, por otro lado, el operador debe seleccionar el archivo CSV guardado previamente luego de la utilización de PET-CT Viewer. Recordemos que este archivo contiene la información resultante del análisis de ROIs trazadas. Inmediatamente, el usuario elige variable y familias de ROIs que desea graficar, creándose un gráfico por cada variable pudiendo comparar una o más ROIs en el mismo gráfico. Del mismo modo, se podrá graficar el VOI correspondiente formado con todas las ROIs trazadas seleccionando la opción *total (axial)* desde el menú de elección a graficar. Al mismo tiempo, se visualiza en pantalla una ventana de texto conteniendo información adicional de las ROIs delimitadas.

Se irán visualizando las gráficas que el usuario desee evaluar hasta que éste decida no graficar ninguna otra; de esta manera, finaliza la ejecución del macro. No obstante, las ventanas generadas durante la ejecución del mismo continúan abiertas a no ser que el operador decida cerrarlas.

Las ventanas que contienen las curvas son interactivas pudiéndose modificar colores, formas y trazos de las mismas. Es posible además almacenar los datos utilizados en las gráficas para un análisis *a posteriori* en profundidad con programas externos o, simplemente, conservar la información resultante. La figura 21 indica los pasos desarrollados en el macro 2 mediante un diagrama de flujo de tareas.

Por otra parte, es importante mencionar que existe un atajo para la ejecución de ambos macros una vez instalados en FIJI, presionando en el teclado numérico 1 y 2 para ejecución del primero o segundo macro, respectivamente, teniendo activa la ventana principal de FIJI. Todo el código de programación redactado para lograr graficar variables SUV en series dinámicas de PET-CT se detalla minuciosamente en la sección Anexo.

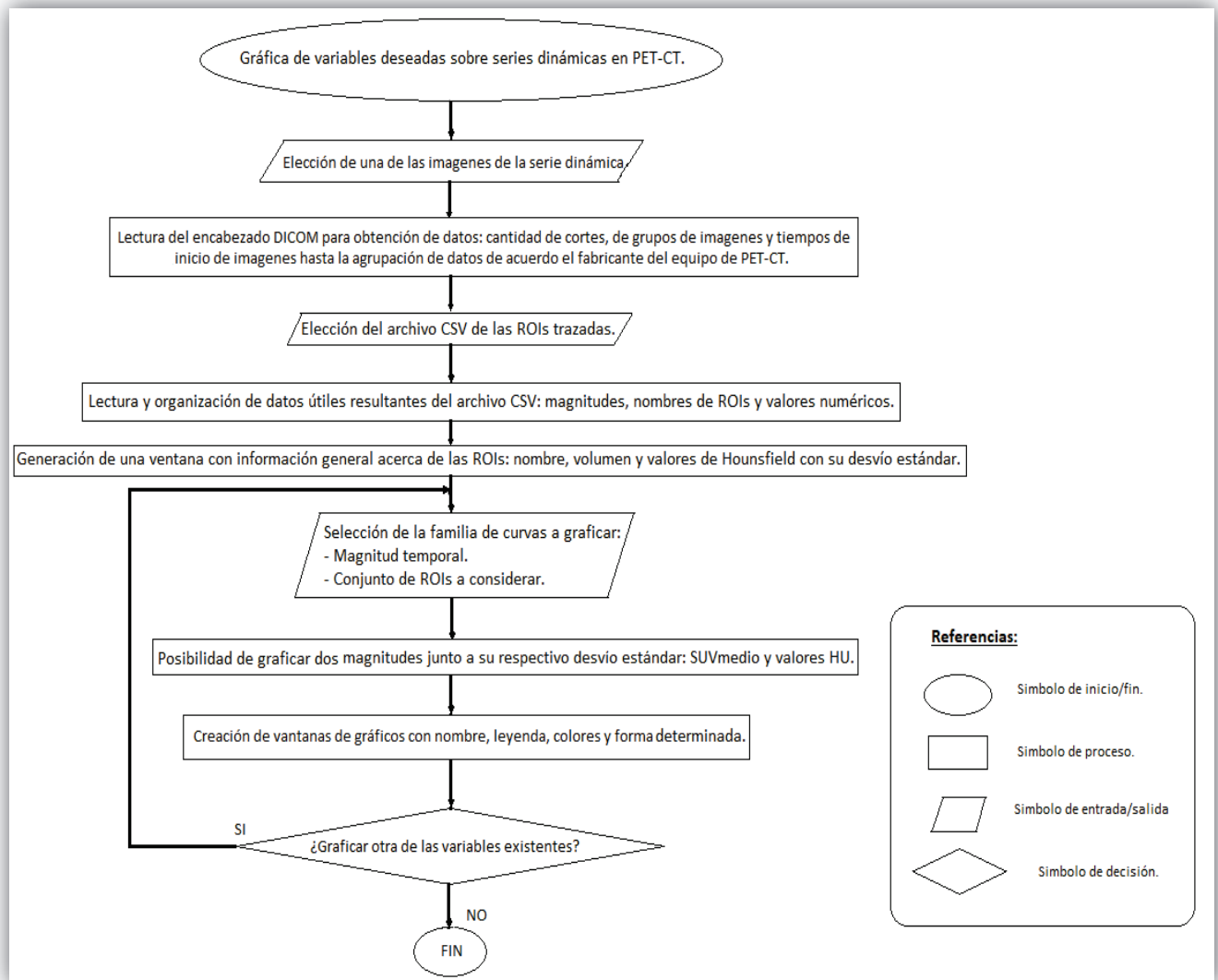


Figura 21. Esquema del diagrama de flujo que explica detalladamente los pasos del macro 2.

7 Validación de la herramienta

La evaluación de la herramienta fue hecha por medio de tres estudios cinéticos donde cada uno presenta particularidades bien notorias que ponen a prueba diversos aspectos de la misma. El primer estudio corresponde a un fantoma con distribución homogénea de actividad conocida. El segundo se basa en un estudio clínico de próstata que tiene la particularidad de corresponder a una distribución simple de actividad. Y por último, se optó por un estudio animal en un primate de pequeño tamaño con una distribución más compleja y con un número de adquisiciones considerable.

7.1 Materiales y Métodos

A fin de evaluar la herramienta de cuantificación de imágenes dinámicas desarrollada en este trabajo se ha optado por trabajar con tres tipos de estudios con características diversas en la modalidad de PET-CT. Los estudios fueron adquiridos en un equipo Philips Gemini TF TOF 64 en la Sección de Imágenes Moleculares y Terapia Metabólica del Centro de Educación Médica e Investigaciones Clínicas Norberto Quirno (CEMIC).

Para cada estudio se confeccionó un conjunto de curvas de diversas magnitudes relacionadas al SUV en función del tiempo. Con el fin de valorar los resultados generados por nuestra herramienta, se compararon los valores SUV_{max} y SUV_{medio} obtenidos con dicha herramienta con los valores obtenidos mediante el software PMOD⁷. Este es un software de carácter comercial ampliamente validado en aplicaciones de cuantificación de imágenes biomédicas, tanto en humanos como en animales. El análisis se realizó mediante el trazado de los mismos VOIs en ambos programas cotejando luego los resultados de SUV_{max} y SUV_{medio} , ya que son las variables que PMOD permite analizar. Finalmente, dado que el software PET-CT Viewer permite obtener otros valores de SUV, como el SUV_{pico} y $SUV \ q \ pico$, para cada estudio se trazaron también curvas cinéticas de dichas magnitudes.

7.1.1 Primer caso: estudio con distribución de actividad homogénea

Como primer estudio, se optó por un caso simple de un fantoma conteniendo una solución radiactiva distribuida uniformemente. La evolución temporal de la actividad del fantoma corresponde sólo al decaimiento natural del radioisótopo. El fantoma consistió en un sachet de solución fisiológica conteniendo 15,875 MBq de ¹⁸F-FDG. La planificación temporal de la serie dinámica se constituyó de la siguiente manera: los primeros cuatro *frames*⁸ duraron 3 segundos cada uno, los siguientes cuatro duraron 10 segundos, los siguientes cinco duraron 30 segundos, los siguientes cinco duraron 60 segundos y los últimos dos 300 segundos, adquiridos de manera contigua. Se totalizó así una serie de 20 adquisiciones.

Se han trazado ROIs circulares en el plano axial desde el corte número 11 al 17 inclusive y de determinados tamaños logrando la delimitación de un VOI en forma de esfera donde la ROI del corte central corresponde al trazo con diámetro más amplio disminuyendo a medida que alcanza a ambos cortes límites (figura 22).

⁷ PMOD Technologies LLC, Suiza, <http://www.pmod.com>.

⁸ Del inglés, en la jerga se denomina *frame* a cada una de las adquisiciones de una serie temporal.

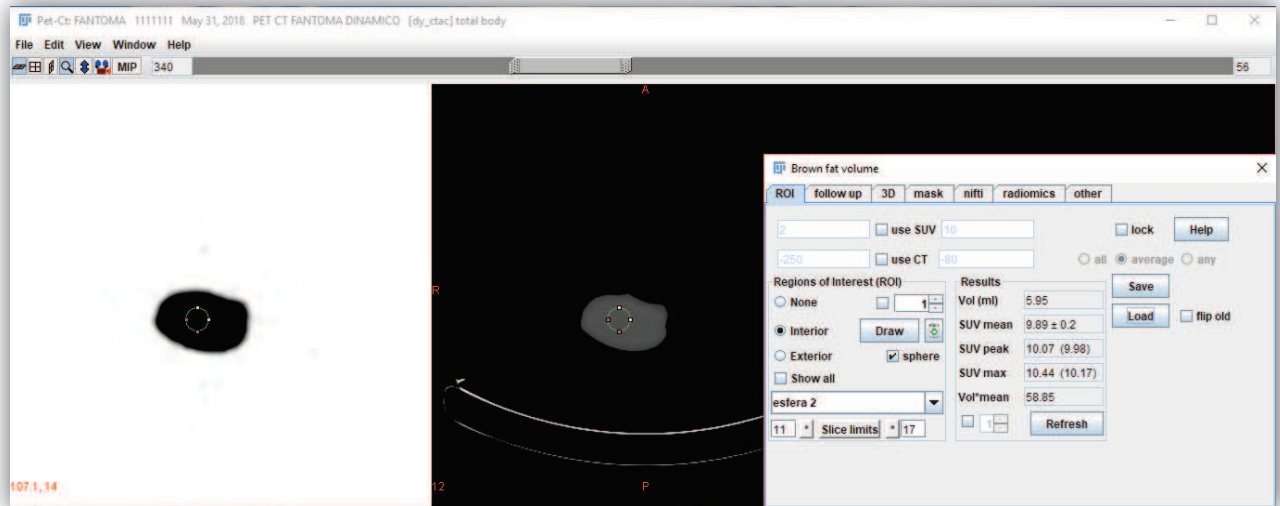


Figura 22. Trazado a modo de ejemplo de una de las ROIs esféricas para el caso del fantoma.

7.1.2 Segundo caso: estudio clínico con distribución de actividad simple (próstata)

Como segundo caso, se utilizó un estudio realizado sobre la próstata con una actividad administrada al paciente de 370 MBq de ^{18}F -colina. La serie dinámica se compone de 10 grupos de imágenes obtenidos en una franja temporal fija de 65 segundos cada uno de manera consecutiva.

Sobre cortes axiales se delimitó la próstata mediante ROIs poligonales, una por corte, abarcando todo el tejido prostático. Se definió así un VOI con 11 cortes (desde el corte número 12 al corte 22) evitando el trazado sobre tejido circundante a la próstata que pudiese distorsionar los valores de captación de la misma (figura 23).

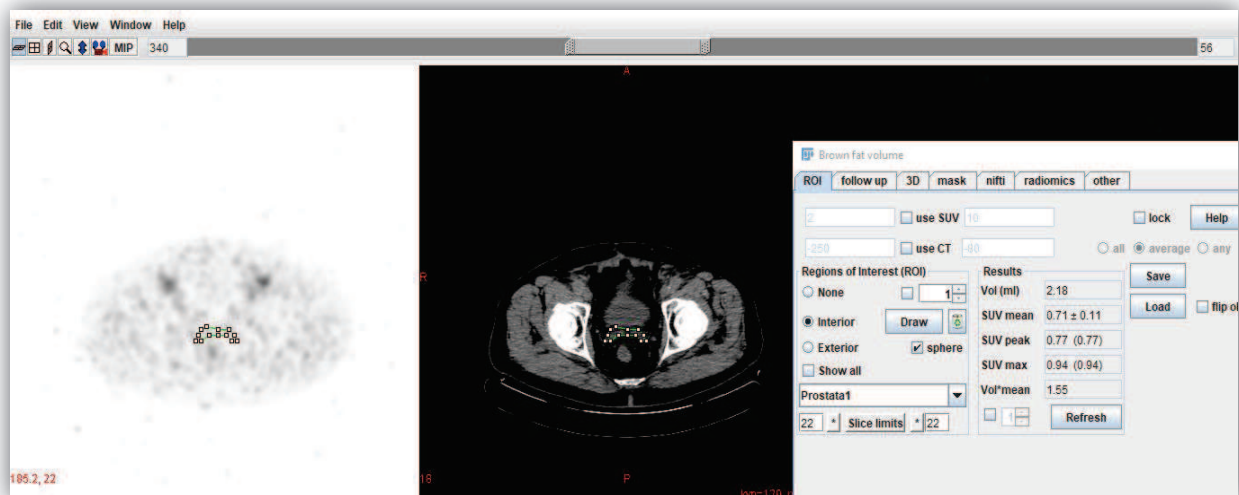


Figura 23. Trazo de una de las ROIs en el plano axial (corte 22) sobre la próstata a modo de ejemplo.

7.1.3 Tercer caso: estudio animal con distribución compleja

Como tercer caso, se emplea un estudio realizado a un primate al cual se le inyectó una actividad de 15,429 MBq de ^{18}F -timidina (^{18}F -FLT). La serie dinámica se compone de 99 grupos de imágenes obtenidos de la siguiente manera: los primeros cuarenta y un grupos de 3 segundos cada uno, los veinticuatro siguientes de 5 segundos, los siguientes veinticuatro de 15 segundos, los siguientes cinco de 60 segundos y los últimos cinco de 300 segundos de manera contigua. Se trazaron ROIs poligonales en varios cortes axiales para delimitar VOIs en diversos órganos (ver tabla 1 y figura 24).

Numeración de VOIs	Órgano delimitado	Cortes del trazado
1	Corazón (ventrículo izquierdo)	35 al 40
2	Hígado	18 al 28
3	Pulmón derecho	19 al 45
4	Pulmón izquierdo	19 al 45
5	Riñón derecho	8 al 18
6	Riñón izquierdo	6 al 16

Tabla 1. Lista de los VOIs trazados en el caso del primate dando extensión de los mismos.

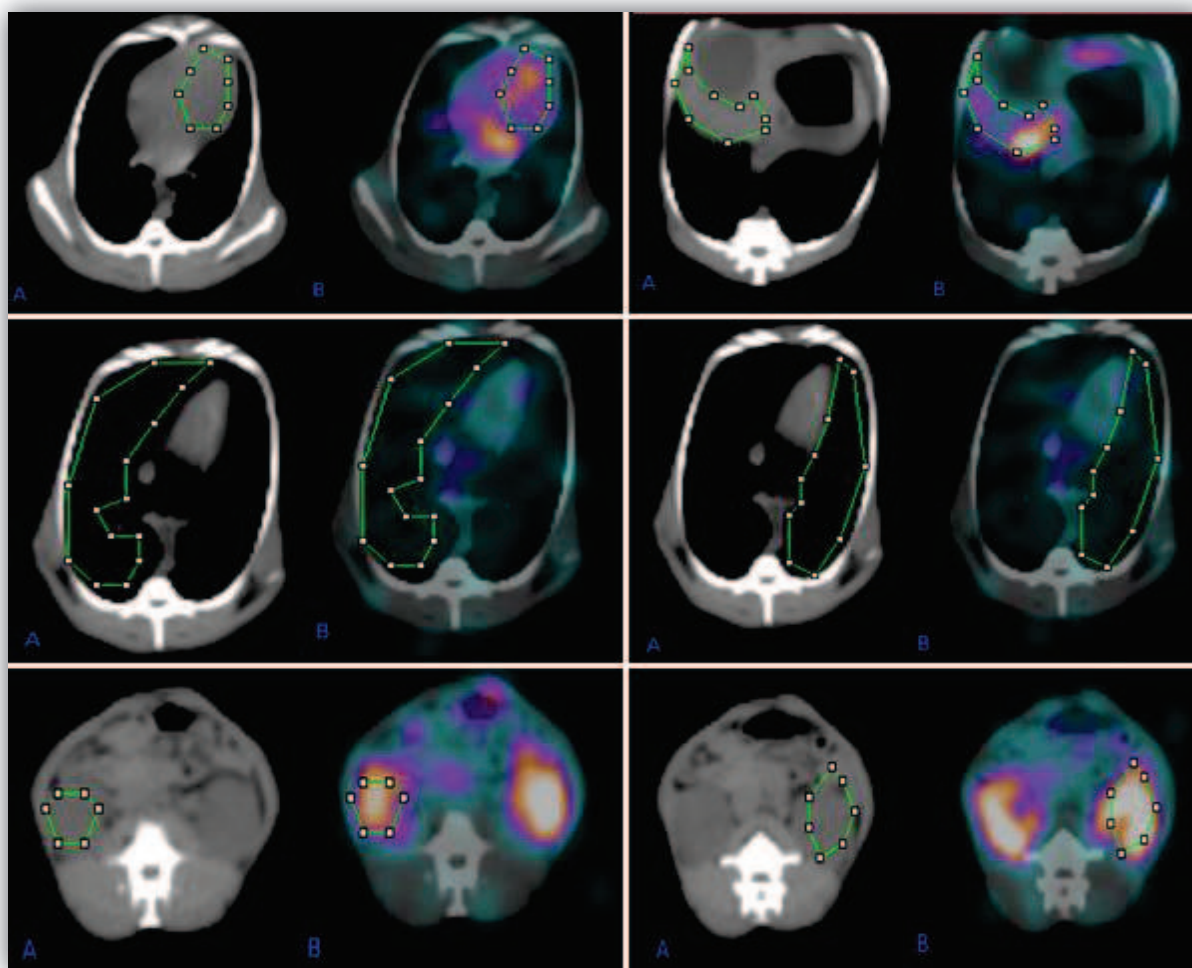


Figura 24. Ilustración general de algunas ROIs delimitadas en cortes representativos en el tercer caso estudiado. Se visualizan ROIs trazadas en varios órganos: corazón, hígado, ambos pulmones y ambos riñones (ver tabla 1).

8 Resultados

8.1 Validación de la herramienta desarrollada utilizando PMOD

Luego de trazadas los ROIs en los tres casos particulares, se procedió a obtener el análisis cuantitativo de los mismos para dos magnitudes relacionadas al SUV: SUV_{max} y SUV_{medio} . Para ello, se utilizó el software PMOD como evaluador del funcionamiento de la herramienta creada comparando los trazos de la concentración del radiofármaco con el transcurso del tiempo.

Si bien en cada caso los valores medidos son de carácter discreto, se unieron los puntos mediante trazos rectos con el fin de facilitar la apreciación visual de los resultados.

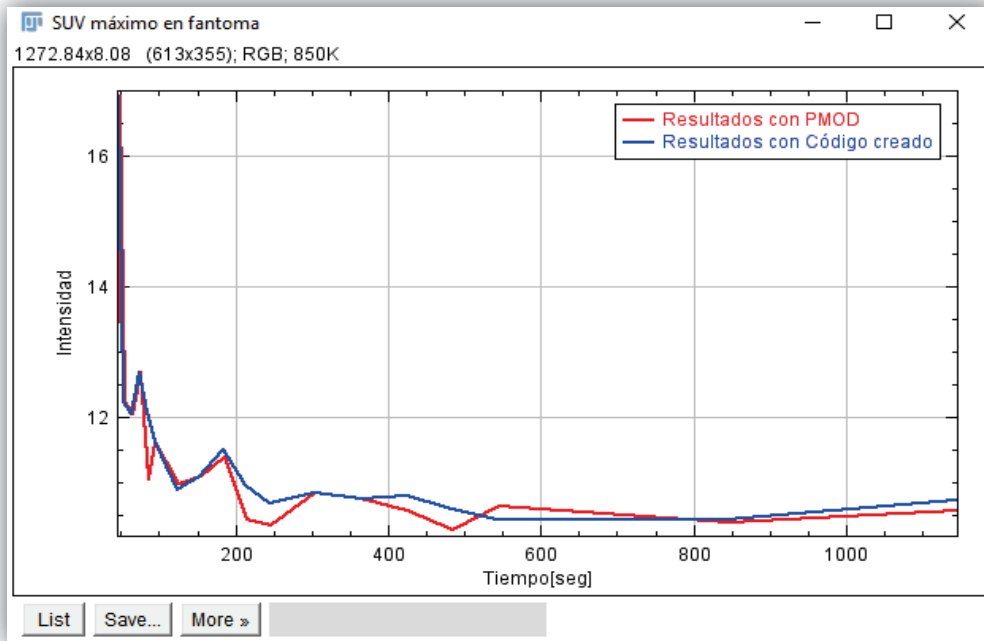


Figura 25. Curvas correspondiente al SUV_{max} sobre el fantoma comparando el código creado y PMOD.

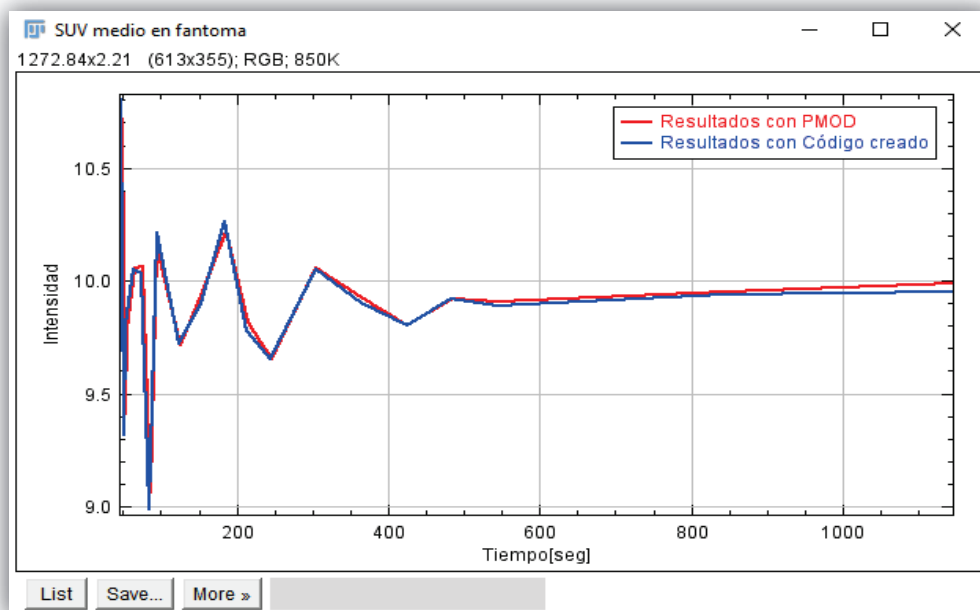


Figura 26. Curvas correspondiente al SUV_{medio} sobre el fantoma comparando el código creado y PMOD.

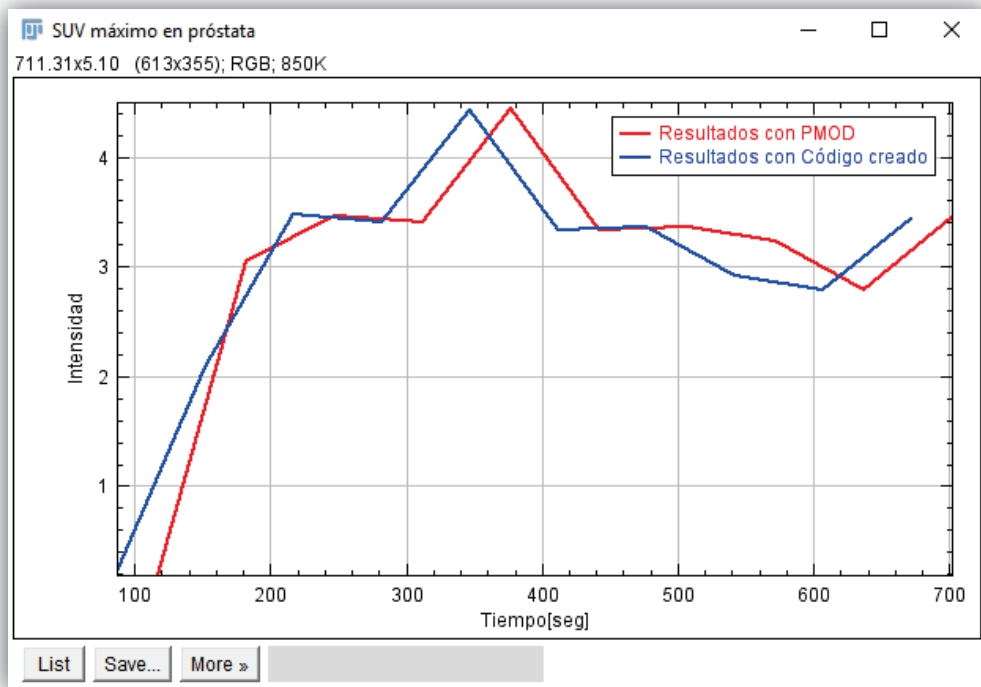


Figura 27. Curvas de SUV_{max} en próstata comparando el código creado y PMOD.

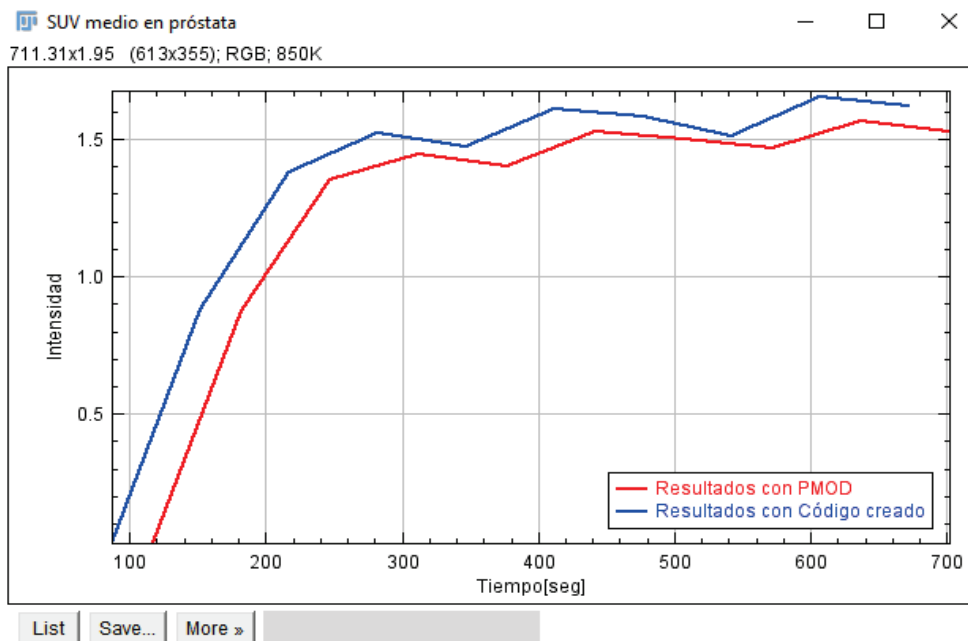


Figura 28. Curvas de SUV_{medio} en próstata comparando el código creado y PMOD.

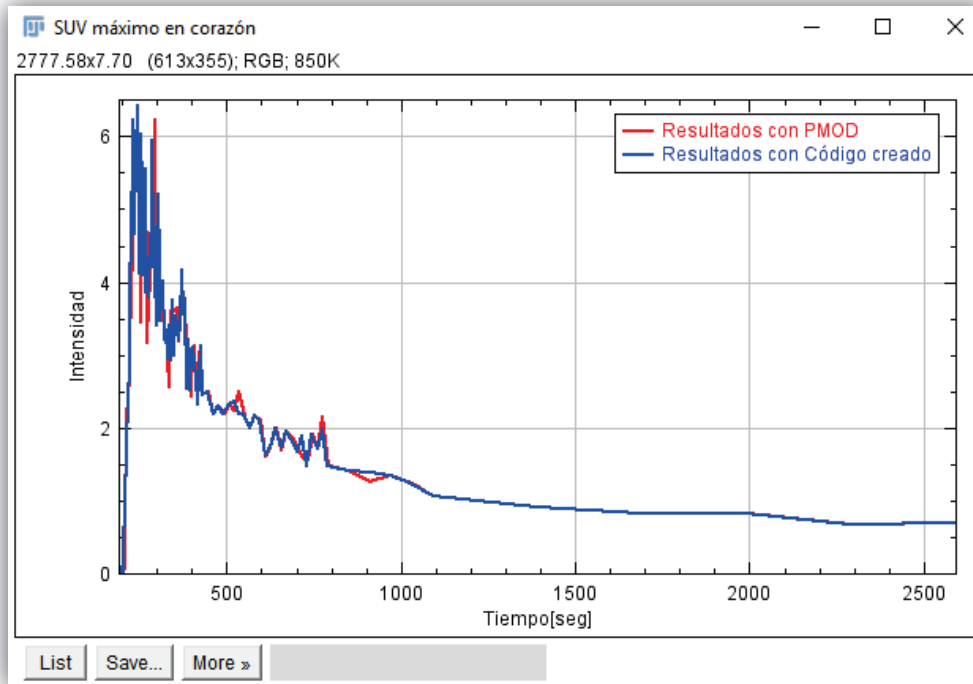


Figura 29. Curvas de SUV_{max} en corazón comparando el código creado y PMOD.

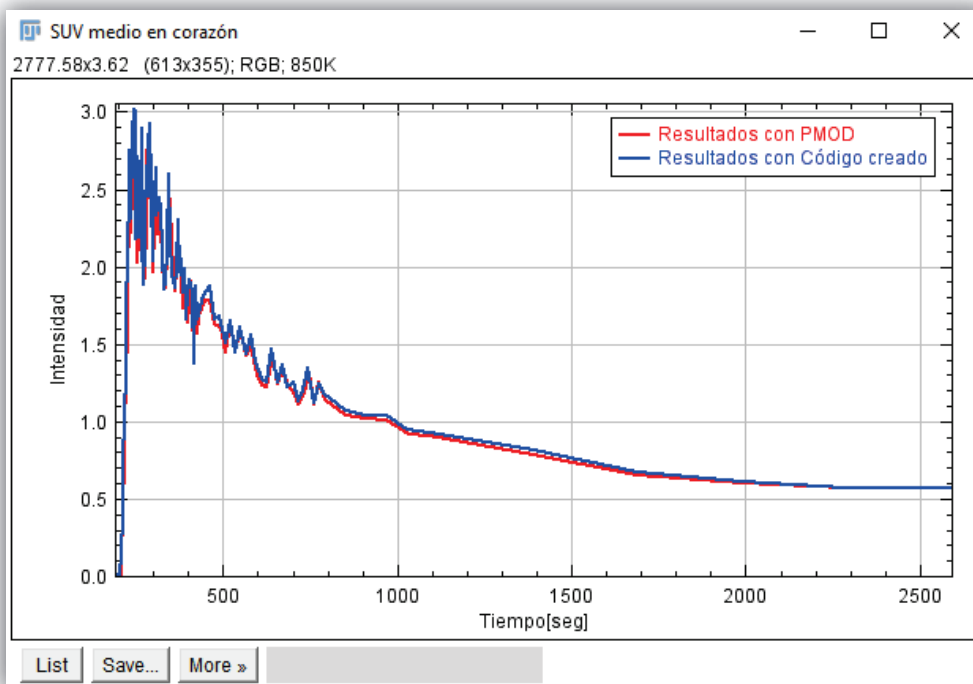


Figura 30. Curvas comparativas para SUV_{medio} en corazón en el código creado y PMOD.

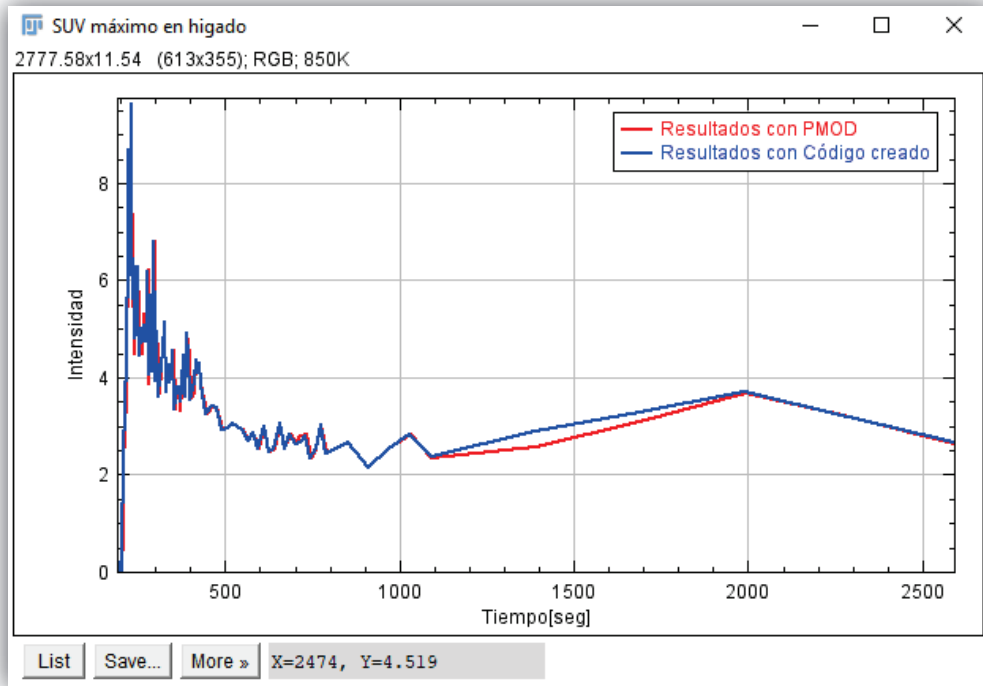


Figura 31. Curvas comparativas para SUV_{max} en hígado trazadas sobre el código creado y PMOD.

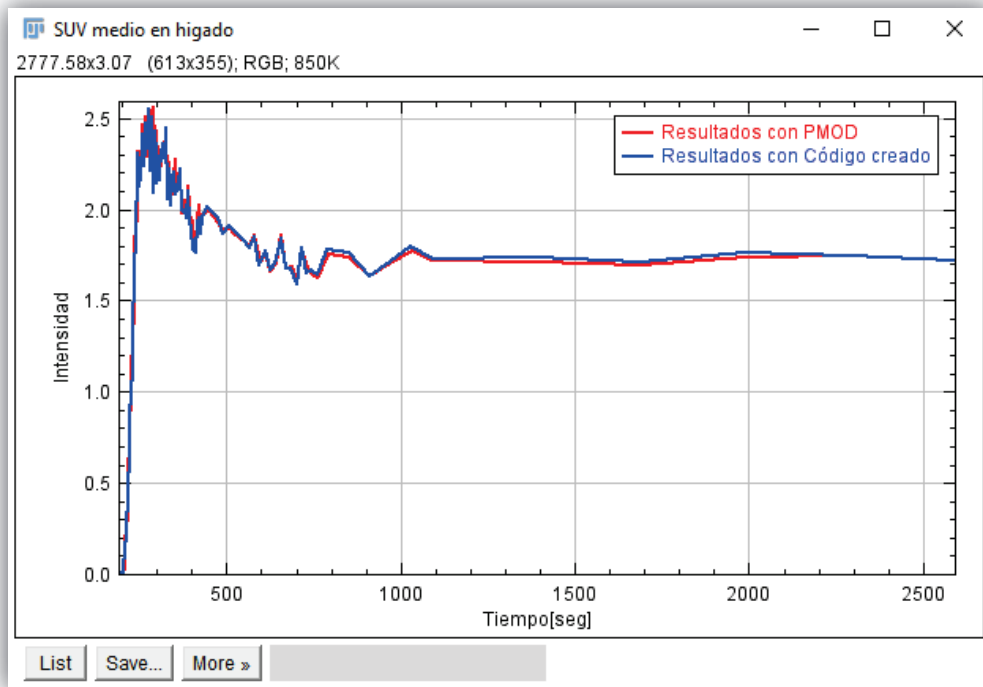


Figura 32. Curvas de SUV_{medio} en hígado sobre el código creado y PMOD.

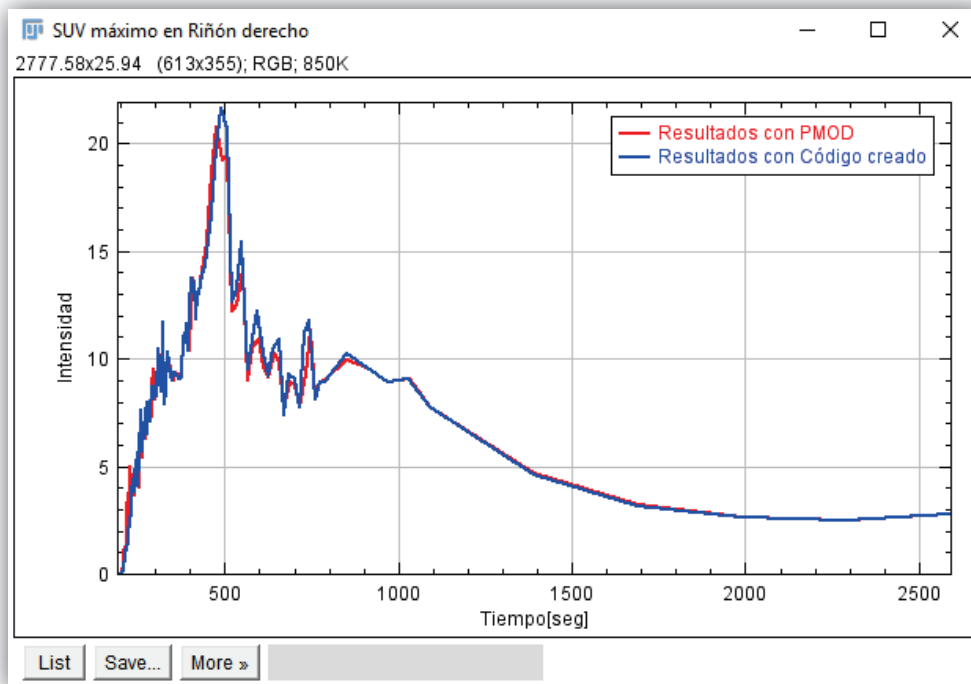


Figura 33. Curvas comparativas de SUV_{max} en riñón derecho sobre el código creado y PMOD.

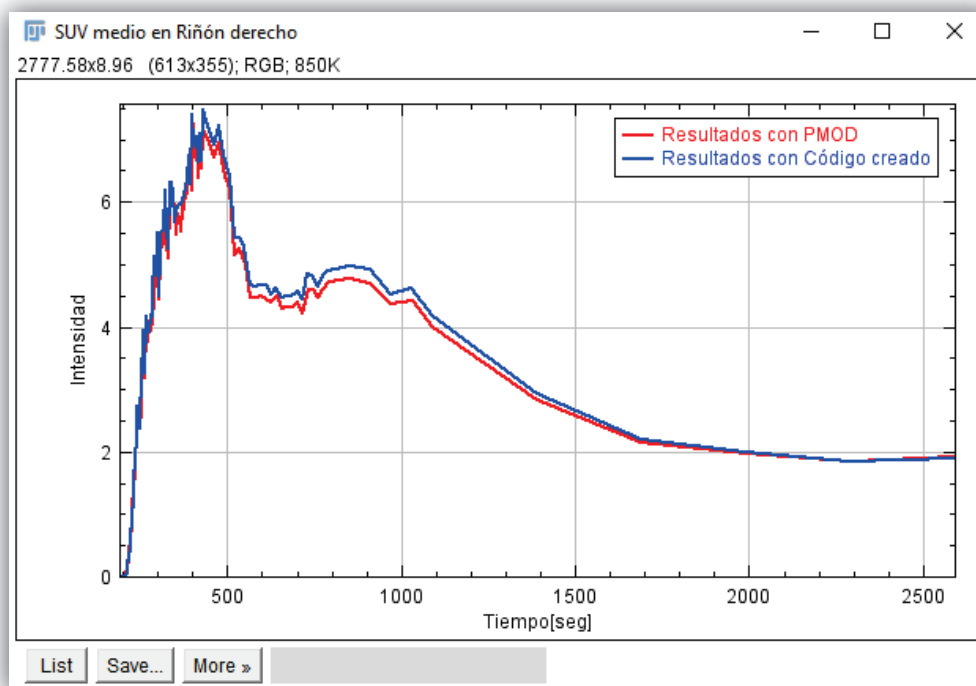


Figura 34. Curvas comparativas resultantes para SUV_{medio} en riñón derecho sobre el código creado y PMOD.

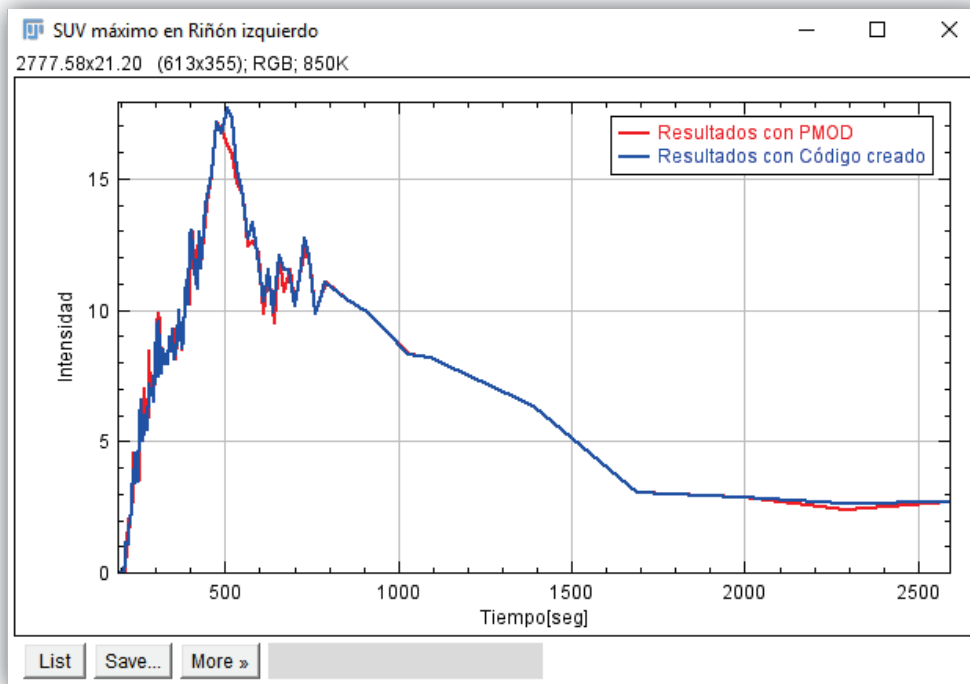


Figura 35. Curvas de SUV_{max} en riñón izquierdo comparando resultados en el código creado y PMOD.

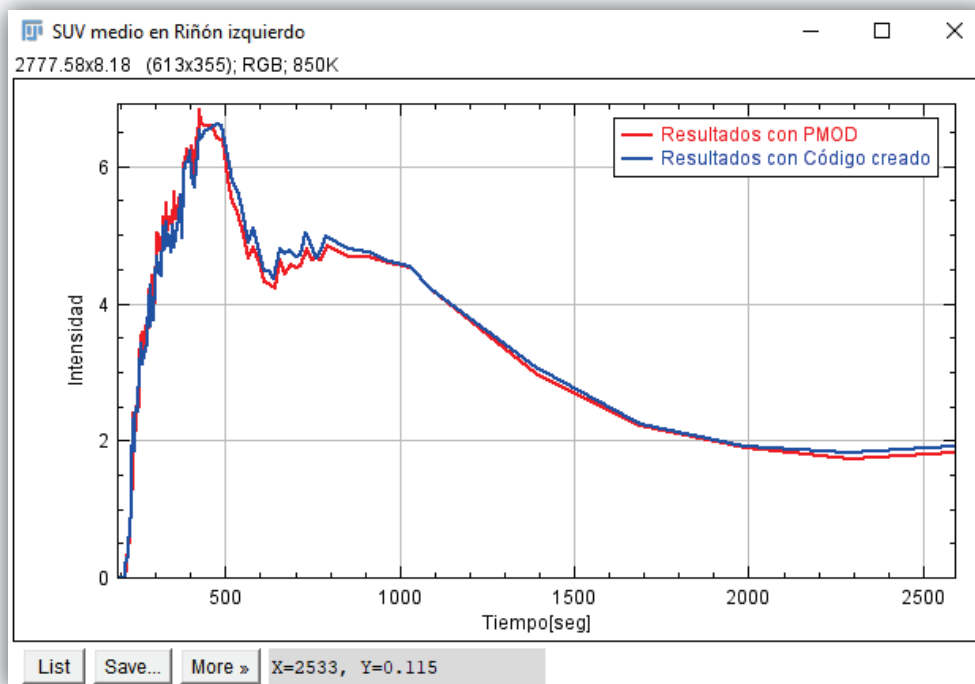


Figura 36. Curvas obtenidas de SUV_{medio} en riñón izquierdo comparando resultados en el código creado y PMOD.

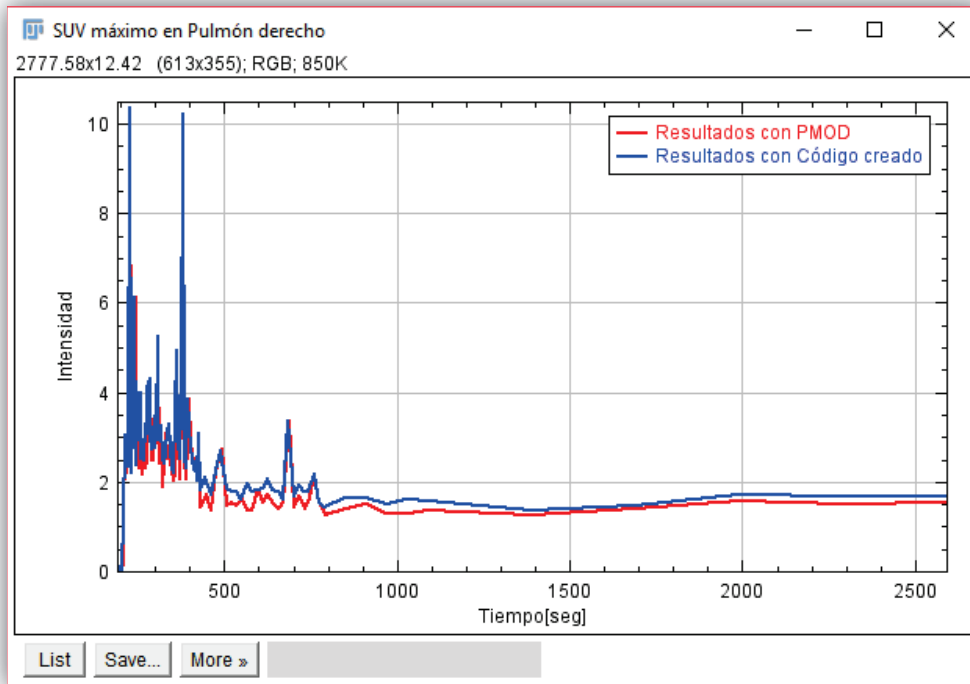


Figura 37. Curvas resultantes de SUV_{max} en pulmón derecho comparando el código creado y PMOD.

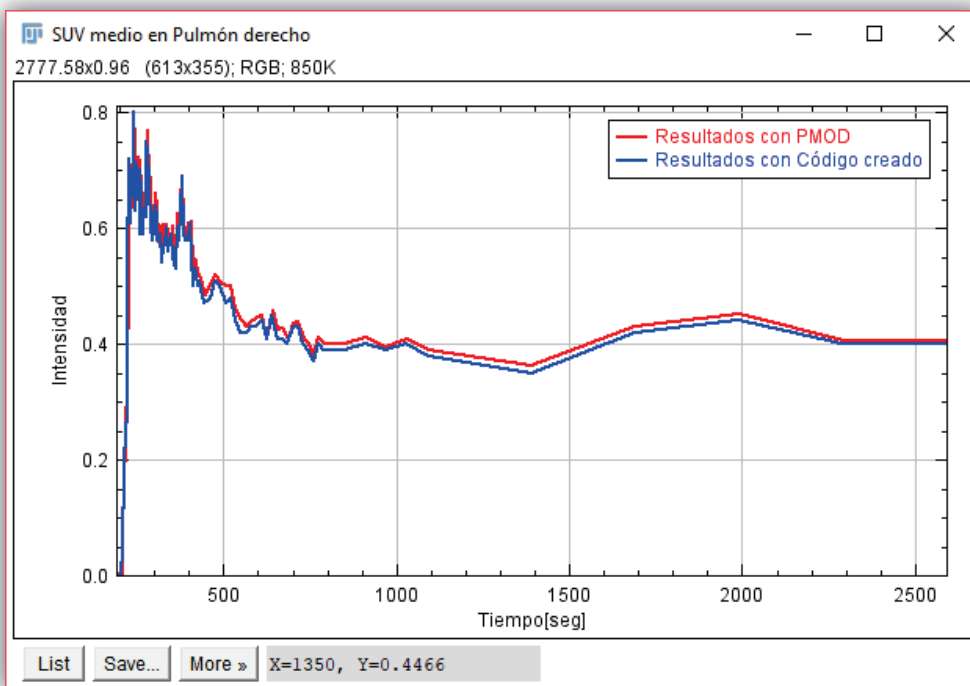


Figura 38. Curvas obtenidas de SUV_{medio} en pulmón derecho comparando resultados en el código creado y PMOD.

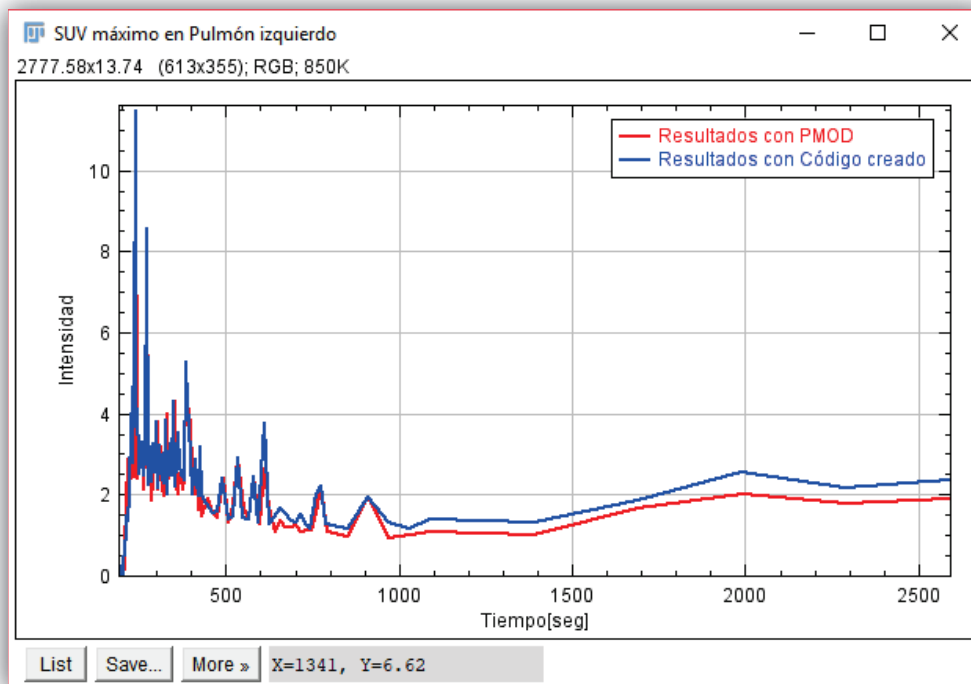


Figura 39. Curvas comparativas de SUV_{max} en pulmón izquierdo para el código creado y PMOD.

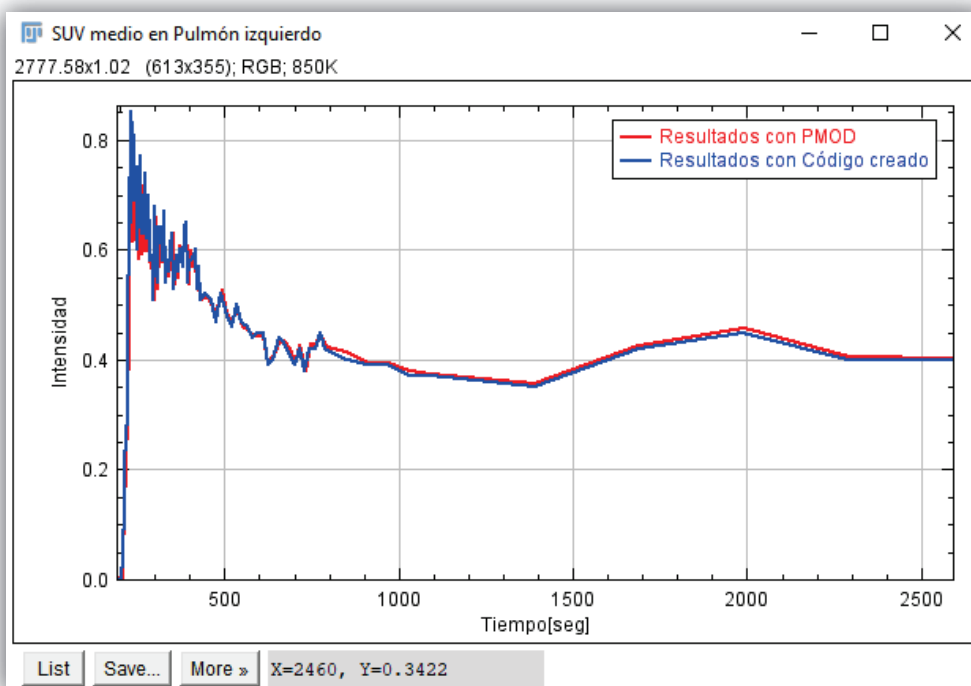


Figura 40. Curvas resultantes de SUV_{medio} en pulmón izquierdo comparando el código creado y PMOD.

Desde las curvas obtenidas y comparadas contra PMOD se nota principalmente leves y casi imperceptibles desfases sobre el eje horizontal relacionados con la elección de la variable temporal considerada para la determinación del SUV con el transcurso del tiempo. Como se ha mencionado anteriormente, para evaluar la herramienta creada se utilizó objetivamente el tiempo de referencia para cada grupo de imágenes pero PMOD utiliza un criterio diferente, puede

deducirse que el desfase de las curvas se encuentra representado aproximadamente por la mitad de la duración del primer grupo de imágenes para cada estudio de PET-CT. De esta manera, por ejemplo, en el primer caso tenemos un desfase de 1,5 s porque el primer grupo de imágenes dura 3 s pero esta situación es casi imperceptible en los trazados debido a presentar un valor de corrimiento muy pequeño en comparación a la extensión temporal del estudio. Sin embargo, el desfase se torna notorio cuando la duración de los grupos de imágenes aumenta como puede verse en las gráficas correspondientes al caso clínico de la próstata debido a que el primer grupo de imágenes dura 65 s obteniéndose un desfase de 30 s (figuras 27 y 28).

Los valores de SUV obtenidos por medio de PMOD y PET-CT Viewer son altamente coincidentes. Atribuimos las pequeñas diferencias observadas a reproducciones inexactas del trazado de las ROIs en ambos programas. A este respecto es importante mencionar que PET-CT Viewer establece las ROIs sobre la base de vóxeles enteros mientras que PMOD establece por omisión ignorar vóxeles cuyo volumen no se integre con un mínimo del 50% dentro de la ROI. Aunque el usuario puede variar este porcentaje, no se utilizó esta opción en el presente análisis. Es por lo tanto esperable que en algunos casos existan diferencias en los vóxeles considerados en cada ROI y, por lo tanto, diferencias en los cálculos de SUVs. Por ejemplo, en el estudio del primate, el hecho de integrar o no algún vóxel periférico daría lugar a las pequeñas diferencias encontradas para SUV_{max} en los pulmones dado que son regiones contiguas a otras con altos valores de concentración como son el hígado, sobre el pulmón derecho, y el corazón, sobre el izquierdo (figuras 37 y 39).

8.2 Análisis de variables relacionadas al SUV generadas por PET-CT Viewer

Luego de validar la herramienta desarrollada con referencia a PMOD, se procedió a analizar en conjunto magnitudes relacionadas al SUV cuya finalidad es recolectar mayor información sobre las ROIs trazadas.

De las curvas obtenidas podemos observar fluctuaciones destacadas sobre los primeros grupos de imágenes debido al corto tiempo de duración en la adquisición de información para la constitución de las imágenes promoviendo la presencia de ruido estadístico.

Tal como era de esperar, también se observa que el trazo correspondiente al SUV_{max} se encuentra por encima de las demás curvas y que la curva más baja corresponde al SUV_{medio} marcado por el promedio de los valores dentro de cada ROI.

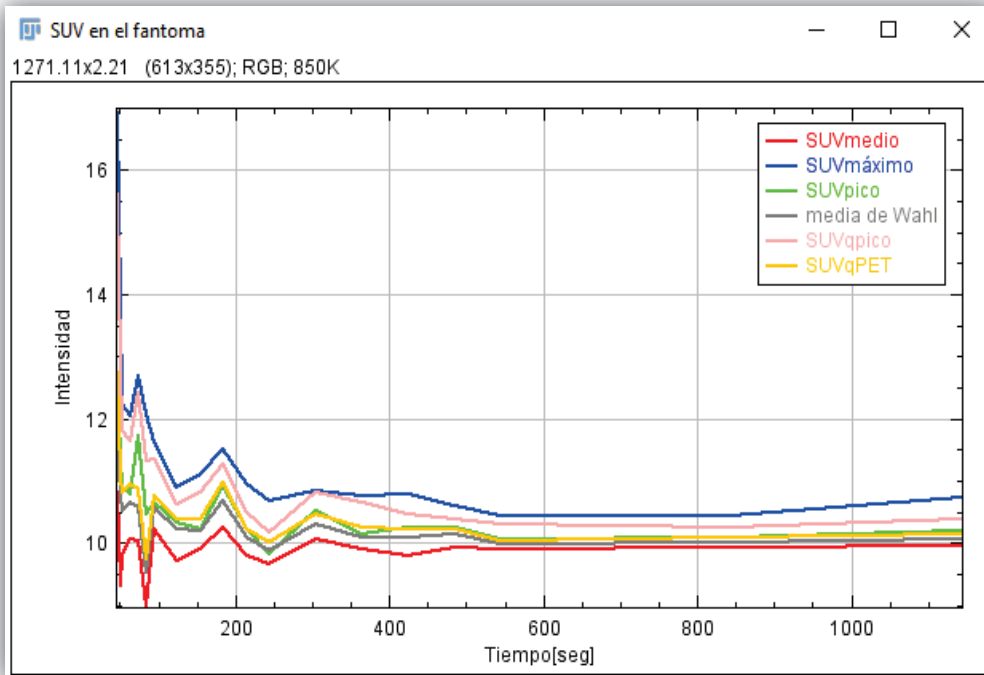


Figura 41. Curvas comparativas de distintas determinaciones de SUV obtenidas sobre el fantoma.

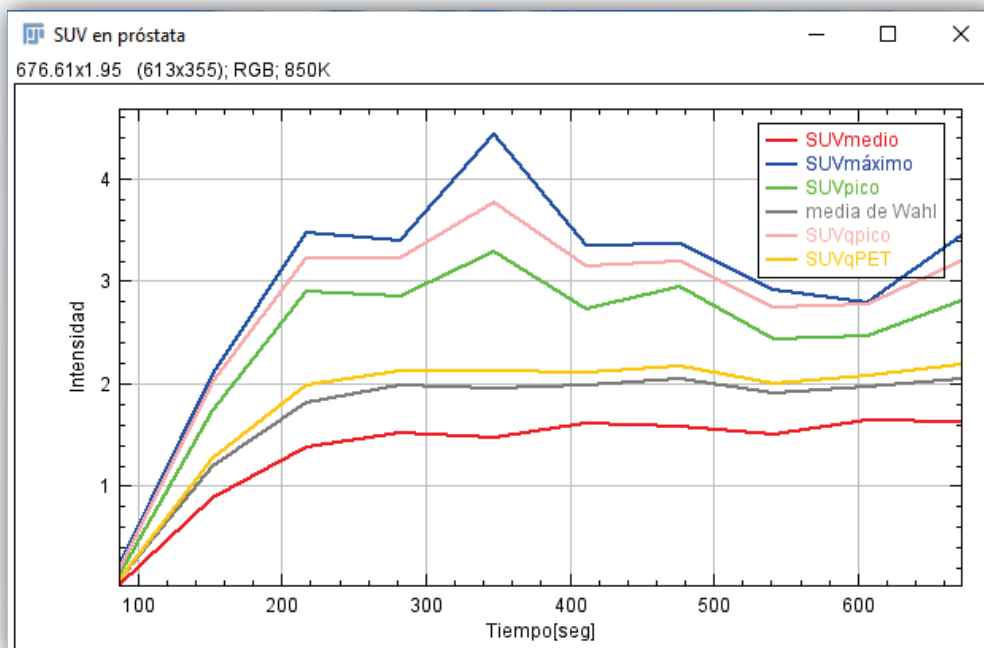


Figura 42. Curvas comparativas de distintas determinaciones de SUV obtenidas sobre la próstata.

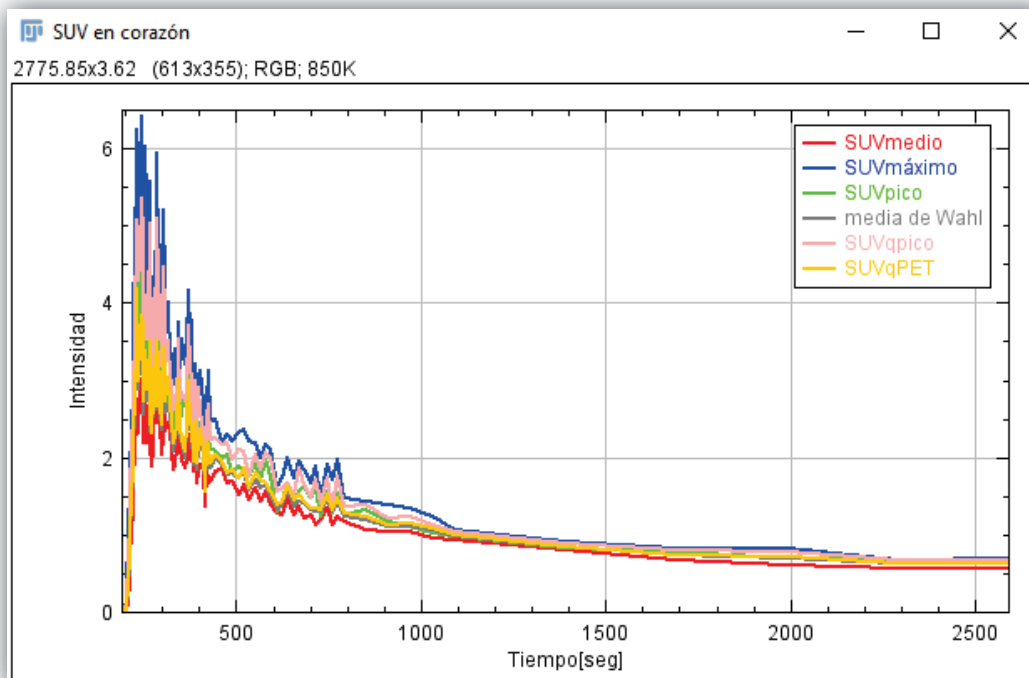


Figura 43. Curvas comparativas de distintas determinaciones de SUV obtenidas sobre el corazón.

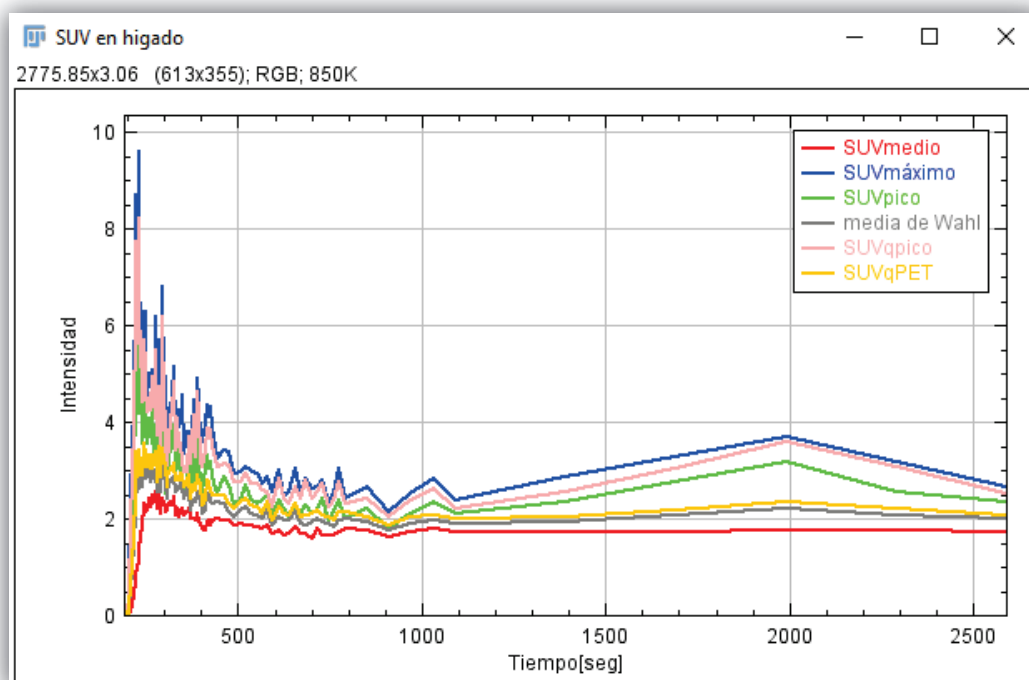


Figura 44. Curvas comparativas de distintas determinaciones de SUV obtenidas sobre el hígado.

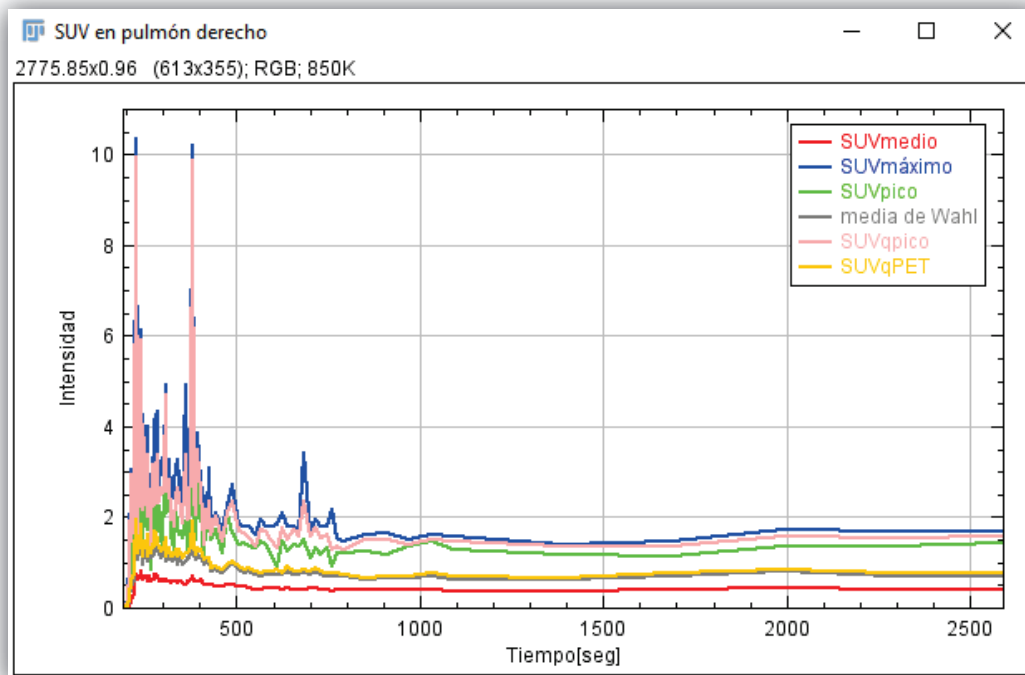


Figura 45. Curvas comparativas de distintas determinaciones de SUV obtenidas sobre el pulmón derecho.

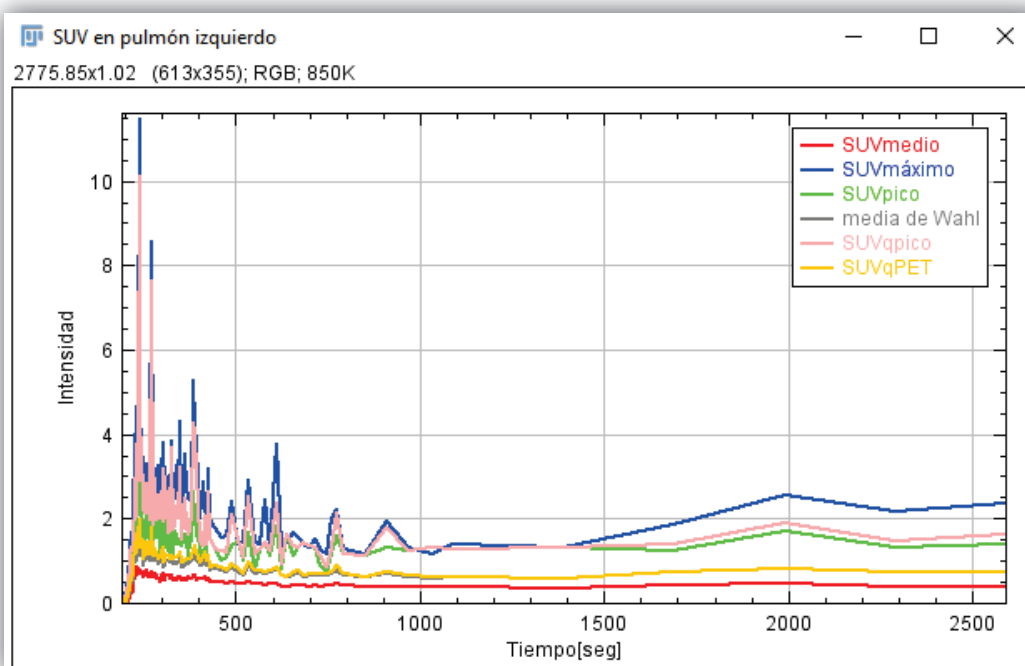


Figura 46. Curvas comparativas de distintas determinaciones de SUV obtenidas sobre el pulmón izquierdo.

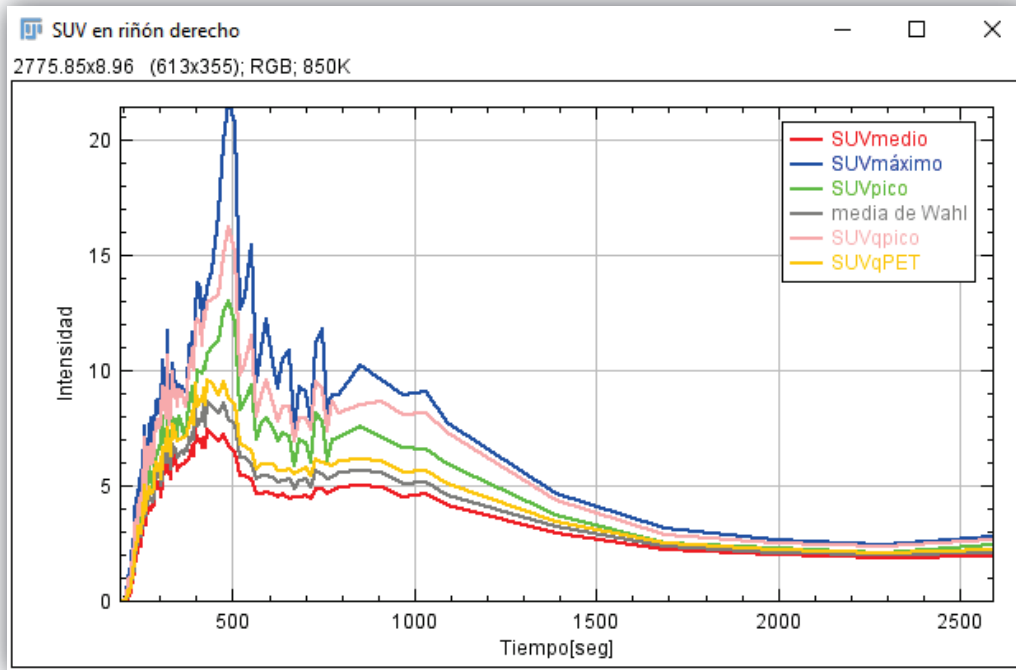


Figura 47. Curvas comparativas de distintas determinaciones de SUV obtenidas sobre el riñón derecho.

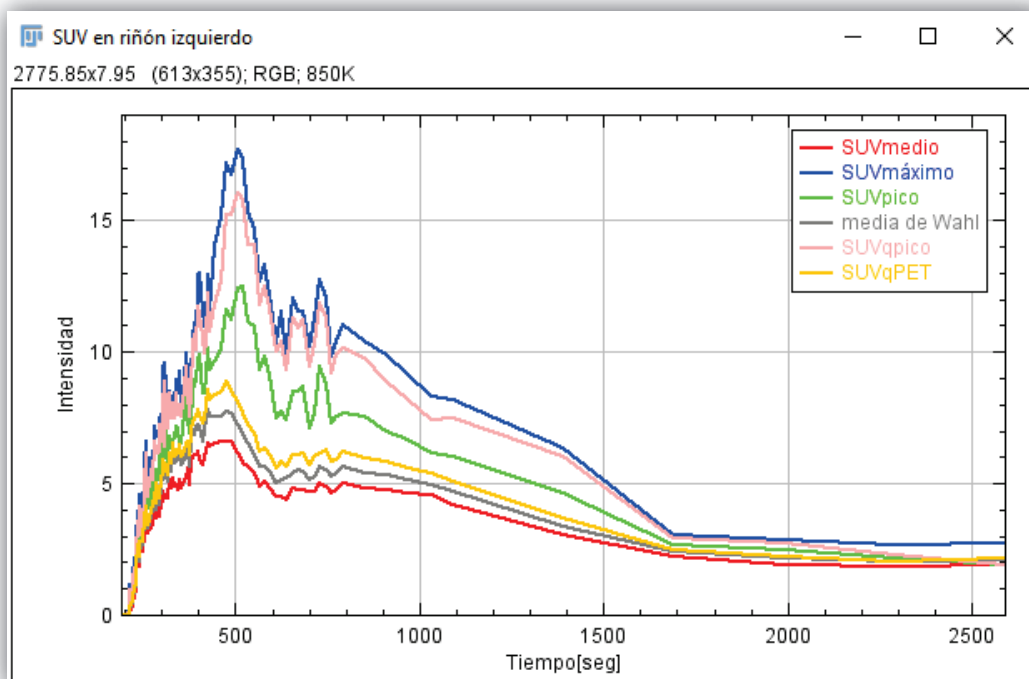


Figura 48. Curvas comparativas de distintas determinaciones de SUV obtenidas sobre el riñón izquierdo.

Los valores $SUV\ pico$ y $SUV\ q\ pico$ son valores relacionados al $SUV\ max$ pero con criterios diversos por tal motivo tienen estilos similares pero con valores disminuidos considerando la vecindad de ese pixel central que contiene el conteo máximo.

Mientras que el valor $SUV\ pico$ corresponde al promedio de las intensidades de los pixeles, el valor $SUV\ q\ pico$ corresponde el promedio del valor máximo junto a otros tres valores que le siguen en

intensidad pero tomados, en ambos casos, dentro de 1 cm^3 de volumen alrededor del voxel en donde se ubica el SUV_{max} . Así, para cada ROI trazada habrá tantas magnitudes SUV_{pico} y $SUV_{q \text{ pico}}$ como cortes estén incluidos en el volumen tisular a analizar. No obstante, se unifican esos valores mediante promedios dando como resultado magnitudes denominadas media de Wahl y Q_{PET} , respectivamente. El valor $SUV_{q \text{ PET}}$ es un marcador de consideración al evaluar concentración residual de la lesión, cuyo valor se utiliza como parámetro dentro de un criterio denominado Deauville que expresa la estadificación de la enfermedad o definición del tratamiento en pacientes pediátricos con diagnóstico de linfoma (Hasenclever et al. 2014).

9 Discusión

Como se ha mencionado durante el transcurso de este trabajo la finalidad de la herramienta computacional desarrollada radica en la necesidad de cosechar información detallada sobre imágenes dinámicas en la modalidad de PET-CT. Se buscó construirla de manera simple y accesible en cualquier sistema operativo, sin necesidad de renovar licencias para su uso y que posibilite el traslado de la misma a cualquier computadora. El código creado presenta la posibilidad de cuantificación simple analizando no solamente SUV_{max} y SUV_{medio} sino que añade dos variables adicionales como son el SUV_{pico} y $SUV_{q\ pico}$ tendientes a suavizar dos problemas sensibles en esta modalidad, como son el ruido y el efecto por volumen parcial para obtener valores más precisos de la concentración del radiofármaco en el tejido estudiado.

Análisis cualitativos y cuantitativos de imágenes en PET-CT al inicio y en el transcurso de la terapia ayudan a determinar la eficacia del tratamiento analizando claramente tamaño del tumor, concentración de ^{18}F -FDG mediante la intensidad del SUV y progreso de la lesión (Wahl et al. 2009).

El SUV_{max} se compone como el valor preferido para determinar la captación del radiofármaco que resiste frente al efecto de volumen parcial pero es verdaderamente susceptible a la calidad estadística de las imágenes y el tamaño del pixel (Nahmias y Wahl 2008). Por otra parte, el abordaje médico para describir en los informes la evolución de la patología depende fuertemente de varios factores, como son, la experiencia clínica, conocimiento de parámetros considerados normales o no, presencia de artefactos y conocimiento de los patrones de avance de la enfermedad (Wahl et al. 2009).

En el primer caso propuesto, se evaluó el desempeño de la herramienta sobre un caso simple donde se podía conocer por adelantado la curva esperada, sabiendo que el SUV se relaciona a la masa corporal del paciente y la actividad inyectada para el estudio, que es fácilmente computable. Teóricamente al administrar una cantidad de actividad se espera que ésta se distribuya homogéneamente en todo el recipiente obteniendo mismos valores SUV en todo el contenido. Por otro lado, al estar los datos corregidos por decaimiento radiactivo, se esperaban gráficas constantes de principio a fin. Sin embargo, la presencia de ruido estadístico se manifiesta en saltos abruptos sobre los primeros grupos de imágenes debido a la corta duración de los mismos (y escaso número de eventos) en la etapa de adquisición de las imágenes de PET-CT.

Sobre el segundo caso, se eligió un caso clínico donde la próstata resulta una región accesible pudiendo ejecutar un trazado de la zona relativamente simple pensando quizás en lograr la implementación a futuro del código creado en los servicios de medicina nuclear del mismo modo en que se utilizan los programas de cuantificación de imágenes médicas en la actualidad.

Como tercer caso, se encuentra un estudio en un primate. En este caso la dificultad consistió, por un lado, en el manejo de un número de adquisiciones elevado, que al utilizar esta herramienta que se pretende implementar se aborda de manera sencilla creando, si se quiere, varios archivos CSV dependiendo de las zonas a cuantificar y, por el otro, en la definición de ROIs sobre diversas regiones anatómicas, algunas de ellas de pequeño tamaño.

Los resultados obtenidos en las pruebas de validación reflejan una buena concordancia con las curvas trazadas con PMOD mostrando así la confiabilidad de la herramienta desarrollada. Sin embargo, debe tenerse en cuenta el trazado de las mismas para evitar contribuir al efecto por volumen parcial e incidir directamente en la información de las magnitudes resultantes traducido

en la aparición de falsos positivos (Wahl et al. 2009). No obstante, se trazaron curvas acerca de magnitudes relacionadas al SUV_{max} principalmente para lograr una correcta información incluyendo datos acerca de los pixeles vecinos sobre cada ROI con la menor contribución de ruido al efectuar un suavizado de varios máximos hallados en cada uno de esos trazos elegidos.

Asimismo, las medidas relacionadas al SUV dependen de variadas cuestiones para considerar válidos los datos obtenidos, principalmente variantes fisiológicas corporales, preparación del paciente y del equipo de PET-CT y de la técnica empleada para el trazo de las ROIs (Lodge 2017). Sin embargo, la consideración de estos aspectos no formó parte de los objetivos de este trabajo, centrado sólo en el desarrollo de una herramienta de análisis de datos.

Así, se ha demostrado la potencialidad del código creado sobre tres grupos de estudios con características bien distintas, sin necesidad de abrir el estudio en varias oportunidades salvo que se requiera la modificación de alguna de las delimitaciones de las ROIs realizadas.

10 Conclusión

En este trabajo se logró implementar un código de visualización y extracción de datos de captación correspondientes a imágenes dinámicas de la modalidad de PET-CT. El código está basado en el programa Fiji y en un plugin del mismo llamado PET-CT Viewer. La elección de Fiji se basó en su calidad de código abierto y gratuito y en su capacidad de operación en múltiples sistemas operativos. El programa PET-CT Viewer se eligió por ser una herramienta validada de visualización y tratamiento de imágenes multimodales que permite, entre otras cosas, extraer diversas magnitudes de captación del radiofármaco en PET asociadas al SUV y sobre regiones de interés volumétricas (VOIs). Las magnitudes de captación son *SUVmedio*, *SUVmax*, *SUVpico* y *SUV q pico*; para los dos últimos el resultado se da como valor global para todo el VOI y como promedio de dichos valores pero obtenidos corte a corte dentro del VOI. Una vez procesado el estudio dinámico deseado en PET-CT Viewer, la presente herramienta permite visualizar los resultados mediante gráficos de evolución temporal (en términos del campo DICOM (0054,1300) *Frame reference time*) de las magnitudes de captación. Estos gráficos se realizan para los VOIs y magnitudes de captación que el usuario decida. Al utilizar nuestra herramienta los resultados totales generados inicialmente por PET-CT Viewer, el cambio de variables a visualizar o de VOIs a realizar no requiere de una nueva ejecución de PET-CT Viewer, siempre que se trabaje sobre resultados ya guardados en un archivo CSV. Todos los resultados visualizados con la presente herramienta pueden ser exportados a archivos de texto para su posterior reutilización en Fiji o en programas externos.

Las evaluaciones realizadas fueron satisfactorias al comparar el conjunto de resultados generados en común con el programa PMOD, ampliamente validado. Estas evaluaciones sirven de soporte a la confiabilidad del código que hemos generado.

Por su concepción, nuestra herramienta resulta de gran practicidad a la hora de trabajar con imágenes provenientes de cualquier equipo PET-CT, siempre que se encuentren en formato DICOM. De esta manera, se facilita a la comunidad de profesionales el acceso al análisis de estudios dinámicos –sobre cualquier computadora, con diversidad de sistemas operativos– no sólo a través de biomarcadores ampliamente utilizados, como el SUV medio y máximo, sino también mediante otros que tienden a minimizar las deficiencias de los anteriores, particularmente el *SUVpico* y *SUV q pico*.

11 Referencias bibliográficas

- Cherry S R, Sorenson J A, Phelps, M E; Physics in Nuclear Medicine 4th edición; Elsevier, Saunders (2012)
- Fedorov A, Beichel R, Kalpathy-Cramer J, Finet J, Fillion-Robin J-C, Pujol S, Bauer C, Jennings D, Fennessy F, Sonka M, Buatti J, Aylward S R, Miller J V, Pieper S, Kikinis R; “3D Slicer as an image computing platform for the quantitative imaging network”; *Magnetic Resonance Imaging*, **30**(9), 1323-41 (2012). PMID: 22770690. <https://www.slicer.org>
- Ferreyra J M; “Evaluación de los factores metodológicos que afectan la cuantificación de imágenes PET-CT”. Tesis de Maestría en Física Médica, Instituto Balseiro UNC.4, 45-61, (2011)
- Hasenclever D, Kurch L, Mauz-Körholz C, Elsner A, Georgi T, Wallace H, Landman-Parker J, Moryl-Bujakowska A, Cepelová M, Karlén J, Álvarez Fernández-Teijeiro A, Attarbaschi A, Fosså A, Pears J, Hraskova A, Bergsträsser E, Beishuizen A, Uyttebroeck A, Schomerus E, Sabri O, Körholz D, Kluge R, “qPET - a quantitative extension of the Deauville scale to assess response in interim FDG-PET scans in lymphoma”; *European Journal of Nuclear Medicine and Molecular Imaging*, **41**, 1301–1308 (2014)
- Lodge M A; “Repeatability of SUV in oncologic 18F-FDG PET”; *Journal of Nuclear Medicine*, **58**, 523-532 (2017)
- Nahmias C and Wahl L M; “Reproducibility of standardized uptake value measurements determined by 18F-FDG PET in malignant tumors”, *Journal of Nuclear Medicine*, **49**, 1804-1808 (2008)
- NEMA (National Electrical Manufacturers Association), Digital Imaging and Communications in Medicine (DICOM) ps3.3 (2018): Information Object Definitions. Disponible en: <http://dicom.nema.org/medical/dicom/current/output/pdf/part03.pdf>
- Rasband W S, ImageJ, U. S. National Institutes of Health, Bethesda, Maryland, USA, <https://imagej.nih.gov/ij/>, 1997-2018, página visitada el 29 de agosto de 2019
- Schindelin J, Arganda-Carreras I, Frise E et al. (2012). "Fiji: an open-source platform for biological-image analysis", *Nature Methods*, **9**(7), 676-682 (2012)
- Schneider C A, Rasband W S, Eliceiri K W; "NIH Image to ImageJ: 25 years of image analysis"; *Nature Methods*, **9**(7), 671-675 (2012)
- Spinelli C P; “Efecto de la aplicación de un filtro gaussiano sobre el índice SUV de la imagen PET, determinado en un fantoma”. Tesis de Licenciatura en Diagnóstico por Imágenes, Universidad Nacional de San Martín (2018)
- Tal I, Kanoun S, PET/CT viewer, <http://www.petctviewer.org/>, página visitada el 29 de agosto de 2019
- Wahl R L, Jacene H, Kasamon Y, Lodge M A; “From Recist to Persist: evolving considerations for PET response criteria in solid tumors”, *Journal of Nuclear Medicine*, **50**(5 suppl), 122S-150S (2009)

12 Anexo

12.1 Código de programación creado en Fiji para cuantificar Imágenes dinámicas en PET-CT

```
macro "Busqueda de la serie dinamica [1]" { //Macro que permite la busqueda de
la serie dinamica
requires("1.52a");
setBatchMode(true); //Evitamos la visualizacion de todas las imagenes

Table.create("Archivos DICOM"); //Se crea una tabla donde se veran los nombres
de los archivos en busqueda del stack dinamico
fila = 0;
waitForUser("B"+fromCharCode(250)+"squeda de la serie
din"+fromCharCode(225)+"mica", "Debes seleccionar una imagen DICOM a la vez de
cada carpeta del estudio hasta encontrar la que corresponde a la serie
din"+fromCharCode(225)+"mica");
do {
NombreArchivo = File.openDialog("Elije un archivo"); //Se abre un cuadro
de dialogo para elección de una imagen de alguna de las carpetas DICOM
open(NombreArchivo); //Muestra la imagen seleccionada
IDimagen = getImageID();
carpeta = File.directory; //Selecciona la ruta de acceso al ultimo
archivo elegido para añadirlo a la tabla
enc = getImageInfo(); //Comando para abrir encabezado DICOM
NombreDeLasSeries = getInfo("0008,103E"); //Se llena el matriz de datos
creada con los nombres de las series de acuerdo a las imagenes seleccionadas
del codigo tag del estandar DICOM
NombrePaciente = getInfo("0010,0010");
Table.set("Nombres de carpeta",fila,carpeta,"Archivos DICOM"); //Se
establecen los nombres y datos de las columnas con los cuales se va llenando la
tabla
Table.set("Discripcion de las series",fila,NombreDeLasSeries,"Archivos
DICOM");
Table.set("Nombre de paciente",fila,NombrePaciente,"Archivos DICOM");
Table.update("Archivos DICOM"); //Se actualiza la tabla con aquellos
datos de las tags mencionadas a medida que se van eligiendo archivos
if (fila == 0) //Se condiciona para que en la primer eleccion de un
archivo se pueda expandir la tabla y ver mejor los datos en ella
waitForUser("Acomoda la tabla de ser necesario para mejor
visualizaci"+fromCharCode(243)+"n de los datos.");
fila = fila+1; //fila se va incrementando hasta que el usuario deje de
elegir carpetas porque encontro la carpeta que contiene la serie dinamica
} while (getBoolean(fromCharCode(191)+"Desea seleccionar otra carpeta?.",
"S"+fromCharCode(237), "No"));
selectImage(IDimagen);
close();
setBatchMode(false);
} //fin del macro busqueda de la serie dinamica

macro "Grafica de variables sobre series dinamicas de PET-CT [2]" {
```

```

waitForUser("Atenci"+fromCharCode(243)+"n...", "Debes seleccionar una de las
imagenes de la serie din"+fromCharCode(225)+"mica");
setBatchMode(true); //A partir de aca se ejecuta el macro escondiendo las
imagenes que utiliza
NombreArchivo = File.openDialog("Elije un archivo"); //Cuadro de dialogo para
elegir una de las imagenes de la serie dinamica encontrado anteriormente
open(NombreArchivo); //abrir el archivo Dicom deseado
enc = getImageInfo(); //abrir el encabezado DICOM de las imagenes a
analizar

numero_img = getInfo("0020,0013"); //se seleccionan datos relevantes de las
imagenes sobre el encabezado DICOM
num_img = parseFloat(numero_img); //Transformacion de caracteres a numeros

ss = getInfo("0054,0081");
slices = parseFloat(ss);

fs = getInfo("0054,0101");
frames = parseFloat(fs);

tot_imagenes = slices*frames;

t_inicio = newArray(frames); //Creacion de matriz de datos para alojar
valores temporales
t_gruposIMGS = newArray(frames);
t_ref = newArray(frames);

//dir = getDirectory("Elegi un directorio"); //Abre un cuadro de
dialogo para elegir una carpeta determinada
dir = File.directory;
lista = getFileList(dir); //Se arma una lista con los nombres de los
archivos presentes en la carpeta seleccionada
fabricante = getInfo("0008,0070"); //Se crea una variable donde se ubicara la
cadena de caracteres de ese tag en particular de la ultima imagen abierta

listaNumOrd = newArray(lengthOf (lista)); //Generamos la variable listaNumOrd
como un matriz de elementos con la misma dimension que contiene la variable
lista
if (fabricante == " Philips Medical Systems ") { //Si el equipo donde se
realizo el estudio es Philips se crea el siguiente ciclo
    lista = Array.slice(lista,1); //Eliminamos el primer elemento del matriz
Lista (DIRFILE)
    for (i=0; i<lista.length; i++){ //Se genera un ciclo que vaya recorriendo
los elementos del matriz de datos Lista
        listaNumOrd[i] = parseInt(substring(lista[i],1)); //Para cada elemento de
la matriz eliminar el primer caracter (I en este caso)
    } //fin del ciclo de eliminacion del primer caracter del nombre del archivo
dinamico
    Array.sort(listaNumOrd); //Se ordenan los elementos del matriz de datos de
menor a mayor
    for (i=0; i<frames; i=i+1) { //Creacion de un ciclo organizando datos
temporales de las distintas grupos de imagenes del scan dinamico

```

```

        open(dir+"I"+listaNumOrd[i*slices+1]); //Abre los archivos compuestos
por la variable dir(ruta de acceso) mas la letra I(presente en los archivos de
las imgs) mas la variable listaNumOrd
        t_inicio[i] = getInfo("0008,0032"); //Se llena el matriz de datos
t_inicio con los valores temporales correspondientes a la hora de inicio de
cada una de los grupos de imagenes
        t_gruposIMGS[i] = getInfo("0018,1242"); //Se llena el matriz de datos
t_gruposIMGS con los valores correspondientes a los valores de duracion de cada
una de los grupos de imagenes
        t_ref[i] = parseFloat(getInfo("0054,1300"))/1000;
//print(i+"          "+t_inicio[i]+"          "+t_gruposIMGS[i]+"
          "+t_ref[i]); //Imprime en columnas la serie de datos que
obtuvimos de los grupos de imagenes
    } //fin del loop de arrays temporales de los grupos de imagenes en
equipo Philips
}

if (fabricante == " SIEMENS ") { //Si el equipo donde se realizo el estudio es
Siemens se crea el siguiente ciclo
    for (i=0; i<lista.length; i++){ //Se genera un ciclo que vaya
recorriendo los elementos de la matriz Lista
        listaNumOrd[i] = parseInt(lista[i]); //Para cada elemento de la matriz se
transforma los caracteres en numeros
    } //fin del ciclo
    Array.sort(listaNumOrd); //Se ordenan los elementos del matriz de menor a
mayor
        for (i=0; i<frames; i=i+1) { //Creacion de un ciclo organizando datos
temporales de los distintos grupos de imagenes de la serie dinamica
            open(dir+listaNumOrd[i*slices+1]); //Abre los archivos compuestos por
la variable dir(ruta de acceso) mas la variable listaNumOrd (con los numeros de
los archivos de las imagenes)
            t_inicio[i] = getInfo("0008,0032"); //Se llena la matriz t_inicio con
los valores temporales correspondientes a la hora de inicio de cada una de los
grupos de la imagenes
            t_gruposIMGS[i] = getInfo("0018,1242"); //Se llena la matriz
t_gruposIMGS con los valores correspondientes a los valores de duracion de cada
una de las grupos de imagenes
            t_ref[i] = parseFloat(getInfo("0054,1300"))/1000; //Se llena la
matriz t_ref con los valores correspondientes a los valores de referencia a
mitad de cada una de las grupos de imagenes en segundos
            //print(i+"          "+t_inicio[i]+"          "+t_gruposIMGS[i]+"          "+t_ref[i]);
//Imprime en columnas la serie de datos que obtuvimos de los grupos de imagenes
        } //fin del loop de arrays temporales de los grupos de imagenes
    }
setBatchMode(false); //Indica el cierre de la orden otorgada de ocultar las
imagenes dentro de macro ejecutado
Array.reverse(t_ref);

waitForUser("Archivo CSV", "Debes seleccionar el archivo csv obtenido del
trazado de las ROIs");
eleccion = File.openDialog("Elige una archivo CSV"); //Al grabar el macro y al
abrirlo, se muestra un cuadro de dialogo con el nombre elegido para realizar
una accion determinada

```

```

 analisis = File.openAsString(eleccion); //Creacion de una variable para la
 apertura de un archivo (que contiene datos de las ROI'S trazadas) como una
 cadena de caracteres
 analisis = replace(analisis,"Phase [0-9]+\n",""); //Actualizacion de la
 variable creada donde se reemplazada la palabra PHASE mas un numero por un
 caracter vacio
 analisis_renglones = split(analisis,"\n"); //Se cambia el nombre de la
 variable creada en el renglon anterior con la orden de romper la cadena de
 caracteres ante la presencia de un salto de renglon
 nrenglonestot = lengthOf(analisis_renglones); //Se crea otra variable con el
 valor en longitud de la variable actualizada
 analisis = replace(analisis,fromCharCode(177),","); //Se actualiza la variable
 analisis reemplazando el caracter ± por una coma, de esta manera, se agregan
 mas caracteres

 i=1; //SE CREA LOOP CON LA FINALIDAD DE SECTORIZAR DATOS RELEVANTES Y CON LOS
 QUE SE NECESITA TRABAJAR
 do { //Se crea un ciclo actualizando la variable ANALISIS_RENGLONES
     indice = indexOf(analisis_renglones[i],"Number of ROIs ="); //Se coloca
 un substring que se va a buscar en la cadena de caracteres mencionada en el
 renglon de arriba y lo que se realiza con la funcion indexOf, es mostrar todos
 los elementos del caracter previo al indice que contiene ese subcaracter
 determinado
     if (i==nrenglonestot-1){
         exit("El numero de renglones con datos de ROIs excede el
 tama"+fromCharCode(241)+"o del archivo");
     }
     i++; //Incrementador de i en uno, o sea va recorriendo elemento tras
 elemento de la matriz
 } while (indice == -1); //La condicion del ciclo implica que continue hasta
 que encuentre el subcaracter citado mas arriba "Number of ROIs ="
 nrenglones = i-4; //Se restan 2 unidades primero porque antes de salir del
 ciclo se incremento y otro porque el renglon que posee el subtring determinado
 queremos borrarlo, de esta manera, se muestra hasta el renglon previo

 nrengroi = (nrenglones+3); //Se actualiza la variable nrengroi que vale
 40(variables + datos numericos sin anonymized) se le agrega el tres
 unidades(renglones) hasta alcanzar el renglon que tenga el numero de ROI que se
 trazaron
 ROIrenglon = split(analisis_renglones[nrengroi], "="); //Se divide el caracter
 en 2 cuando encuentre un signo "="
 nROI = parseInt(ROIrenglon[1]); //Se selecciona solo el subcaracter que
 contenga el caracter que representa un numero y se lo transforma en numero

 analisis_renglones = split(analisis,"\\,|\n"); //Actualizar variable
 seccionando la cadena de caracteres
 tam = lengthOf(analisis_renglones); //Creacion de una variable con el valor en
 longitud de la matriz analisis_renglones

 Nombre_ROI = newArray(nROI+1); //Se crean variables con los nombres de los
 valores que surgen del analisis de las ROI'S trazadas
 roi = newArray(nROI+1); //Se acorta la matriz por presentar valores repetidos
 cada nROI mas 1

```

```

tipo = newArray(nROI+1);
volumen = newArray(nROI+1);
volumen_medio = newArray(nrenglones);
SUV_medio = newArray(nrenglones);
SD = newArray(nrenglones);
SUV_pico = newArray(nrenglones);
SUVmedio_Wahl = newArray(nrenglones);
SUV_3picos = newArray(nrenglones);
SUV3picos_medio = newArray(nrenglones);
SUV_maximo = newArray(nrenglones);
SUVmax_medio = newArray(nrenglones);
HUpromedio = newArray(nrenglones);
HUstd = newArray(nrenglones);

for(i=0; i<nROI+1;i++){
    Nombre_ROI[i] = analisis_renglones[14 + i*15]; //Aqui llenamos las
matrices creadas fuera del ciclo analizando la posicion del primer valor de la
primer variable citada (Nombre_ROI) y de ahi en mas para las demas variables la
posicion incrementa en una unidad
    roi[i] = analisis_renglones[15 + i*15];
    tipo[i] = parseInt(analisis_renglones[16 + i*15]);
    volumen[i] = parseFloat(analisis_renglones[17 + i*15]);
} //fin de recorte de las primeras variables con datos repetitivos
Nombre_ROI[nROI] = "total (" + Nombre_ROI[nROI] + ")";

for (i=0; i<nrenglones; i++) { //Se crea ciclo para agrupar los datos del
archivo csv segun cada una de las variables cuantitativas obtenidas del
 analisis de las ROI'S
    volumen_medio[i] = parseFloat(analisis_renglones[18 + i*15]);
    SUV_medio[i] = parseFloat(analisis_renglones[19 + i*15]);
    SD[i] = parseFloat(analisis_renglones[20 + i*15]);
    SUV_pico[i] = parseFloat(analisis_renglones[21 + i*15]);
    SUVmedio_Wahl[i] = parseFloat(analisis_renglones[22 + i*15]);
    SUV_3picos[i] = parseFloat(analisis_renglones[23 + i*15]);
    SUV3picos_medio[i] = parseFloat(analisis_renglones[24 + i*15]);
    SUV_maximo[i] = parseFloat(analisis_renglones[25 + i*15]);
    SUVmax_medio[i] = parseFloat(analisis_renglones[26 + i*15]);
    HUpromedio[i] = parseFloat(analisis_renglones[27 + i*15]);
    HUstd[i] = parseFloat(analisis_renglones[28 + i*15]);
} // fin for nrenglones

titulo1 = "Datos adicionales de las ROIs"; //se genera un titulo para el
cuadro de texto
titulo2 = "[" + titulo1 + "];
t = titulo2;
datosROIs = 0;
if (isOpen(titulo1)){ //LOOP PARA AGRUPAR O NO AQUELLOS DATOS DE ROIs A
MEDIDA QUE SE EJECUTA EL MACRO VARIAS VECES
    datosROIs = getBoolean(fromCharCode(191) + "Desea sobrescribir los nuevos
datos adicionales de ROIs?", "Si", "No");
    if (datosROIs == 1)
        print(t, "\\Update:");
    else {

```



```

        Dialog.addString("Elige un nuevo titulo"); //cambiar titulo si se
desea cambiar datos
        Dialog.show();
        titulo1 = Dialog.getString();
        titulo2 = "["+titulo1+"];
        t = titulo2;
    }
}

s1 = newArray(nROI+1); //se crean matrices para llenar con datos referentes a
nombres de ROIs, volumen y HU para todas las ROIs trazadas
s2 = newArray(nROI+1);
s3 = newArray(nROI+1);
for (k=0; k<nROI+1; k++){ //CREACION DE CUADRO CON DATOS ADICIONALES DE LAS
ROIs
    s1[k] = "Nombre de la ROI: "+ Nombre_ROI[k]; //se llenan las matrices con
los datos de los nombres de las ROIs
    s2[k] = "volumen (ml): "+ volumen[k]; //se llenan las matrices con los
datos de las volúmenes de las ROIs
    s3[k] = "HU: "+ HUpromedio[k]+fromCharCode(177)+HUstd[k]; //se llenan
las matrices con los datos de las unidades de HU de cada ROI

    if (isOpen(titulo1))
        print(t, s1[k]+"\\n"+s2[k]+"\\n"+s3[k]+"\\n"+"\\n");
    else {
        run("Text Window...", "name="+titulo2+" width=72 height=8 menu");
        print(t, s1[k]+"\\n"+s2[k]+"\\n"+s3[k]+"\\n"+"\\n"); //se especifican los
datos para que se llene el cuadro de texto
    }
}

nombresDeVariables = newArray("Volumen medio","SUV medio","Desvio estandar de
SUV","SUV pico","SUV pico medio (Wahl)","SUV Q pico","SUV Q pico medio","SUV
maximo","SUV maximo medio","HU promedio","Desvio estandar de HU"); //Se crea
array con los nombres de las variables que poseen los datos numericos de las
ROIs trazadas
NombreDatosDeROIs = newArray("Nombre de ROI", "Numero de ROI", "Tipo de ROI");
//Se crea otra matriz con datos referidos a las Rois trazadas
selec = newArray(nROI+1); //variable correspondiente a cantidad de ROIs
trazadas y el elemento suma

barr_error_SUV = 0; // inicializacion de variable booleana
relacionada con la graficar barras de error
barr_error_HU = 0;

do {
    Dialog.create("Seleccion de variables a graficar"); //Se crea una caja
de dialogo para que el usuario elija que variables graficar
    Dialog.addChoice("Variables",nombresDeVariables,nombresDeVariables[3]);
//Se agrega a la caja de dialogo una ventana que despliega las variables con
las que se puede trabajar
    Dialog.addCheckboxGroup(5,10,Nombre_ROI,selec); //Se agrega al dialogo
tambien items para seleccionar(los nombres de las ROIs trazadas)

```

```

Dialog.show(); //Visualizacion del cuadro de dialogo
variableY = Dialog.getChoice();
if (variableY == "SUV medio") {
    barr_error_SUV = getBoolean(fromCharCode(191)+"Deseas graficar el
desv"+fromCharCode(237)+"o est"+fromCharCode(225)+"ndar de 'SUV medio' como
barras de error?", "Si", "No");
}
if (variableY == "HU promedio"){
    barr_error_HU = getBoolean(fromCharCode(191)+"Deseas graficar el
desv"+fromCharCode(237)+"o est"+fromCharCode(225)+"ndar de 'HU promedio' como
barras de error?", "Si", "No");
}
ROIs_selecc = newArray(nROI+1); //Creacion de una matriz cuyos elementos
representan las Rois trazadas
for (i=0; i<nROI+1; i++) { //Se crea un ciclo que recorra la variable
generada previamente
    ROIs_selecc[i] = Dialog.getCheckbox(); //Para llenar la matriz con
valores ceros y unos, coloca uno cuando el item se selecciona y 0 en caso
contrario
} //fin de roi_selecc.

Plot.create(variableY, "Tiempo[seg]", "Intensidad"); //Se crea la ventana
del grafico con nombre del mismo y de los ejes
Plot.setLineWidth(2);
colores = newArray("blue", "cyan", "darkGray", "green", "magenta",
"orange", "pink", "red", "black", "gray", "lightGray", "yellow"); //Se crea una
matriz con 8 posibles colores para diferenciar las curvas trazadas
leyenda = ""; //Se crea una cadena de caracteres con el nombre leyenda que
se utilizara a posteriori para llenar cada leyenda en la grafica con aquellas
ROI's seleccionadas por el usuario
datosY = newArray(frames);
errorY = newArray(frames);
icolor = 0;
for(i=0; i<nROI+1; i++) { //Se crea un ciclo para graficar la variable
elegida por el usuario y el ciclo se recorre tantas veces como Rois trazadas
mas 1 haya dependiendo de la cantidad de rois seleccionadas por el usuario que
quiera graficar
    if (ROIs_selecc[i]==1) { //Se condiciona el ciclo teniendo como
referencia el item de las variables que el usuario ha seleccionado y el ciclo
va a continuar solamente cuando encuentre valores = 1 (los items seleccionados)
        for (j=0; j<frames; j++) { //Se usa otro indice (j) para
trabajar sobre las matrices creados de las variables obtenidas a partir de las
ROIs trazadas
            if (variableY == "Volumen medio") //Se utilizan
sentencias if comparando (==) la eleccion del usuario con cada una de las
variables a graficar
                datosY[j] = volumen_medio[i+(j*(nROI+1))]; //En tal caso
que haya coincidencia a la matriz datosY se le atribuyen los datos de alguna de
las variables mencionadas (dependiendo de la eleccion) pero solamente aquellos
valores de la roi seleccionada (cada variable contiene datos de todas las ROIs
para graficar solo una de las ROIs que se selecciona se debe dividir en tantos
grupos como ROIs se trazaron)
            if (variableY == "SUV medio") {

```

```

        datosY[j] = SUV_medio[i+(j*(nROI+1))];
        errorY[j] = SD[i+(j*(nROI+1))];
    }
    if (variableY == "Desvio estandar de SUV")
        datosY[j] = SD[i+(j*(nROI+1))];
    if (variableY == "SUV pico")
        datosY[j] = SUV_pico[i+(j*(nROI+1))];
    if (variableY == "SUV pico medio (Wahl)")
        datosY[j] = SUVmedio_Wahl[i+(j*(nROI+1))];
    if (variableY == "SUV Q pico")
        datosY[j] = SUV_3picos[i+(j*(nROI+1))];
    if (variableY == "SUV Q pico medio")
        datosY[j] = SUV3picos_medio[i+(j*(nROI+1))];
    if (variableY == "SUV maximo")
        datosY[j] = SUV_maximo[i+(j*(nROI+1))];
    if (variableY == "SUV maximo medio")
        datosY[j] = SUVmax_medio[i+(j*(nROI+1))];
    if (variableY == "HU promedio") {
        datosY[j] = HUpromedio[i+(j*(nROI+1))];
        errorY[j] = HUstd[i+(j*(nROI+1))];
    }
    if (variableY == "Desvio estandar HU")
        datosY[j] = HUstd[i+(j*(nROI+1))];
} //fin del ciclo de acuerdo a la seleccion del usuario
Plot.setColor(colores[icolor%lengthOf(colores)],
colores[icolor%lengthOf(colores)]); //Se utiliza un color de la matriz de
colores generado dependiendo de las ROI's seleccionadas a graficar
Plot.add("lines", t_ref, datosY); //Seteo de la forma del trazo de
la curva (en forma de triangulos) y los valores tanto del eje x como del eje y,
los del eje x son constantes (los valores temporales de cada una de los grupos
de imagenes) y los del eje y dependen de la variable seleccionada por el
usuario
    leyenda = leyenda + Nombre_ROI[i] + "\t"; //Aqui se actualiza la
variable leyenda agregando ademas el nombre de la ROI trazada (Nombre_ROI) y un
delimitador de espacio que implica la adiccion de uno o mas nombres de acuerdo a
la eleccion del usuario y cantidad de ROI's a graficar
    Plot.setLimitsToFit();
    Plot.addLegend(leyenda, "top-right"); //Visualizar leyenda
del lado derecho del grafico de acuerdo a la cantidad de ROI's seleccionadas
por el usuario
    icolor = icolor + 1;
} // fin if ROIs_selecc[i]==1.
} //fi del ciclo para graficar variables
if (barr_error_SUV == 1){ //se grafican las barras de error si el usuario
indica que si
    Plot.add("error bars", errorY);
    barr_error_SUV = 0;
}
    if (barr_error_HU == 1){
Plot.add("error bars", errorY);
barr_error_HU = 0;
}
    Plot.show(); //Impresion del grafico

```

```
    eleccionFinal = getBoolean(fromCharCode(191)+"Deseas graficar otra de las
variables?", "Si", "No");

} while (eleccionFinal==1);
} //Fin macro grafica de variables
```