



Universidad
Católica
de Valencia
San Vicente Mártir

Facultad de Psicología

Grado en Psicología

Trabajo de Fin de Grado

Técnicas automatizadas de evaluación de conductas sociales en ratones: metodología y resolución de problemas.

Presentado por: Gema Jiménez Pérez.

Tutor/a: Dra. Ana Pérez Villalba.

Valencia, a 28 de mayo de 2023.

Agradecimientos.

En primer lugar, deseo agradecer a mi tutora, la Dra. Ana Pérez Villalba, todo su apoyo a lo largo del TFG. Gracias a su colaboración, su interés por que el trabajo sea lo mejor posible, su conocimiento y su ánimo.

Gracias a ella he descubierto una pasión por el mundo de la investigación que anteriormente no me había planteado.

También deseo dar las gracias a mis familiares y amigos por todo el apoyo. Han sido un pilar fundamental para seguir adelante a pesar de las dificultades encontradas en el desarrollo del trabajo.

Gracias a todos vosotros estoy aquí.

Resumen.

El uso de ratones para la evaluación de la conducta social permite obtener modelos de experimentación posteriormente replicables en humanos. La necesidad de su uso ha quedado justificada en el ámbito científico, sin embargo, las técnicas para su análisis se encuentran todavía limitadas. Se suele realizar una evaluación manual acerca de las conductas realizadas por los ratones, mas esto presenta una amplia serie de desventajas donde la fiabilidad y la replicabilidad se encuentran comprometidas.

Para solventar dicho problema, se ha tratado de explorar tres técnicas automatizadas de evaluación de la conducta social en ratones: “SMART”, “Id Tracker” y “DeepLabCut”. Finalmente, dadas las limitaciones halladas en los resultados, se ha añadido un *software* complementario (“*simBA*”) a “DeepLabCut” para obtener la información pertinente.

En el trabajo se han adaptado las técnicas al análisis con ratones, se han creado protocolos acerca de su instalación y uso, así como se ha comparado su capacidad para detectar el número de contactos y la latencia de estos con la medición manual.

El análisis comparativo con cada uno de los *softwares* permitió ver las ventajas y desventajas de cada método. De este modo, se seleccionan los más funcionales (“DeepLabCut”) y los más prácticos (“SMART”) para el análisis de conducta.

Palabras clave.

“técnicas automatizadas”, “conducta social”, “ratones”, “deeplabcut”, “smart”.

Abstract.

The use of mice for the evaluation of social behaviour makes it possible to obtain experimental models that can later be replicated in humans. The need for their use has been justified in the scientific field; however, the techniques for their analysis are still limited. The evaluation of the behaviours performed by mice is usually carried out manually, but this presents a wide range of disadvantages where accuracy and replicability are compromised.

To overcome this problem, three automated techniques for assessing social behaviour in mice: "SMART", "Id Tracker" and "DeepLabCut" were explored. Finally, given the limitations found in the results, a complementary software ("simBA") was added to "DeepLabCut" to obtain the relevant information.

This study adapted the techniques for the analysis with mice, created protocols for their installation and use, and compared their ability to detect the number of contacts and the latency of these with manual measurement.

The comparative analysis with each of the softwares allowed to see the advantages and disadvantages of each method. Thus, the most functional ("DeepLabCut") and the most practical ("SMART") are selected for behavioural analysis.

Key words.

“automated techniques”, “social behaviour”, “mice”, “deeplabcut”, “smart”.

Índice.

Resumen.....	4
Abstract.....	5
1. Introducción.....	8
2. Objetivos.....	13
3. Metodología.....	14
4. Resultados.....	24
5. Discusión.....	31
7. Referencias.....	39
8. ANEXO.....	42
Anexo I. Vídeos.....	42
Anexo II. Protocolo de descarga e instalación de “Id Tracker”.....	43
Anexo III. Protocolo de “DeepLabCut” para ordenadores con CPU.....	45
Anexo IV. Protocolo para ordenadores con GPU.....	48
Anexo V. Cómo analizar un vídeo con “Id Tracker”.....	54
Anexo VI. Cómo analizar un vídeo con “DeepLabCut”.....	55
Anexo VII. Cómo instalar simBA.....	61
Anexo VIII. Cómo utilizar simBA.....	63
Anexo IX. Análisis de video con “SMART”.....	72

Índice de imágenes.

Imagen 1.	19
Imagen 2.	19
Imagen 3.	20
Imagen 4.	21
Imagen 5.	22
Imagen 6.	22

Índice de gráficas.

Gráfica 1.	25
Gráfica 2.	25
Gráfica 3.	26
Gráfica 4.	27
Gráfica 5.	27
Gráfica 6.	28
Gráfica 7.	29
Gráfica 8.	30
Gráfica 9.	30
Gráfica 10.	33
Gráfica 11.	34

1. Introducción.

Los animales se utilizan en investigación por diversas razones. Las cuatro principales razones son: el progreso del conocimiento científico, la creación de modelos para el estudio de la enfermedad, el desarrollo y examinación de las potenciales formas de tratamiento y la protección de la seguridad de las personas y del medioambiente (Animal research info: the global source for animal use in science, 2014). Entre las áreas estudiadas en ratones se encuentra la conducta social.

Estudio de la conducta social a través de ratones.

El interés en estudiar la conducta social en ratones se debe a que es esencial para el avance del conocimiento científico. De hecho, Macrì, et al. (2015) exponen la importancia de evaluar el comportamiento social en roedores dada su significancia para la investigación de múltiples campos de la ciencia entre ellos: neurociencia, psicología, biología y farmacología.

Lorbach et al. (2015) recalcan el interés del estudio de sus interacciones como indicadores de desórdenes neurológicos y psiquiátricos. Otros autores (Thanos et al., 2015) destacan su uso como modelo previo antes del trabajo con humanos, bien sea con trabajos de genética, farmacología o simplemente identificando diferentes factores de riesgo, así como el desarrollo de estrategias de tratamiento. Su utilidad no se encuentra limitada a un uso básico, previo al estudio con humanos, puede ser utilizado también para evaluar el desarrollo de los patrones de comportamiento que se han conservado a través de la evolución. Estos pueden proporcionar la capacidad de descubrir y cuantificar diferencias más exactas incluso entre fenotipos (Macrì, et al., 2015; Klibaite et al., 2018).

Asimismo, estos patrones de comportamiento frecuentemente implican la interacción con otros animales o seres humanos. Por lo tanto, conocer los sistemas que controlan dichos comportamientos requiere registrar, así como manipular, la actividad neural mientras que, al mismo tiempo, se mide el comportamiento de los animales (Segalin et al., 2021).

Es más, los ratones son animales sociales, por lo que viven en pequeños grupos. De este modo, el entorno social tiene un impacto importante tanto en su epigenética como en su salud general y la vulnerabilidad ante enfermedades. Se ha probado que el

entorno social desempeña un papel importante en la aparición de enfermedades como la depresión, la ansiedad o la adicción. Por tanto, se precisa una evaluación detallada de las interacciones sociales para poder estudiar el comportamiento social normal y anormal a través de las diferentes fases del desarrollo, su interacción al estrés, el entorno y en enfermedades como el autismo, entre otros (Thanos et al., 2015).

A través del estudio de la conducta social en ratones, se puede comprender la contribución de los genes individuales y el ambiente en el comportamiento. Gracias a esto, es posible descubrir los mecanismos por los que el comportamiento se ve alterado en los trastornos neuropsiquiátricos. Comportamientos tan complejos como la agresión, la preferencia social, la memoria social, el cuidado maternal, la ansiedad, el juego con el mismo sexo, así como el sexo contrario, son todos estudiados a través de ratones, de modo que se proporciona información esencial a la neurobiología del comportamiento humano. El estudiar con ratones como modelos los trastornos como la esquizofrenia, el autismo, los comportamientos sociales, entre otros, es vital para probar la validez de un modelo y proporciona una oportunidad esencial para evaluar la eficacia de diferentes modelos terapéuticos (Page et al., 2009)

Sin embargo, a pesar de la importancia del estudio de su comportamiento, las técnicas existentes son limitadas (Klibaite et al., 2018). El estudio del comportamiento en ratones suele monitorizarse a mano por los propios experimentadores (Segalin et al., 2021). Mas, este tipo de análisis toma una gran cantidad de tiempo perdiendo así su funcionalidad. La anotación manual por parte del experimentador acarrea una serie de desventajas, así como sesgos que pretenden ser compensados con el desarrollo de técnicas automatizadas.

Entre las desventajas y sesgos se encuentran: el tiempo de reacción, fatiga, sesgos propios de la subjetividad, una mayor cantidad de tiempo necesaria, un aumento del coste y una baja estandarización, entre otros (Abbas y Masip-Rodo, 2019; Macrì, et al., 2015; Marcuccio, Savonenko y Etienne-Cummings, 2018; Unger et al., 2017). De hecho, se ha visto que los evaluadores no coinciden al 100%, tornándola por lo tanto en una anotación no reproducible. Incluso, un mismo investigador puede encontrar disparidades en su propia anotación. Este problema también se encontraría solventado por el desarrollo de la evaluación automatizada (Benice y Raber, 2007).

Desarrollo de técnicas automatizadas.

De este modo, debido a la necesidad de disminuir tanto el coste, como el sesgo existente en los observadores humanos, terminan apareciendo las técnicas automatizadas.

Estas han tenido un desarrollo exponencial durante la última década. Los primeros sistemas tan sólo eran capaces de seguir un animal en entornos muy artificiales. En ocasiones incluso era necesaria una participación relativamente activa del experimentador, por ejemplo, siguiendo el trazo del camino realizado por el ratón en el ordenador (Spink, et al., 2001).

Los sistemas modernos, que se basan en el vídeo a color, pueden rastrear a los animales incluso en una variedad de fondos complejos, pudiendo ser entornos más naturales. La demanda de un mayor rendimiento está impulsando la automatización no sólo más rápida, sino que permita realizar experimentos más variados. De hecho, en ratones, la inspección del comportamiento en sus jaulas domésticas ha sido un discriminador importante de las diferencias no sólo de cepa sino incluso fenotipos de enfermedad. Por ejemplo, diferencias a la hora de caminar, acicalarse o dormir (Schaefer y Claridge-Chang, 2012).

Todavía en 2012 se consideraba el ámbito de la automatización para evaluar el comportamiento social como una rama todavía por desarrollar. Schaefer y Claridge-Chang (2012) consideran el seguimiento de múltiples identidades inaccesible para roedores. Aun así, se crean métodos de identificación por radiofrecuencia (RFID). En estos es necesario un marcaje con etiqueta para poder diferenciar y realizar un seguimiento de los ratones. Es más, en la actualidad existen varios de estos métodos como son los creados por Habedank et al. (2021) y Peleh, et al. (2019). Sin embargo, como se verá a lo largo del trabajo, los métodos anteriormente mencionados no se han considerado, al resultar invasivos y, por tanto, alterar los resultados de la evaluación.

Así pues, con el desarrollo de las técnicas automatizadas aparecen diferentes enfoques: los basados en la sustracción del fondo y los basados en el aprendizaje (Abbas y Masip-Rodo, 2019).

Los basados en la sustracción del fondo tratan de detectar y seguir el movimiento restando la imagen actual píxel por píxel de una imagen de referencia en la que se vea el fondo sin los animales. Los píxeles que muestran una diferencia superior al umbral

de referencia se consideran como la diferencia u objeto. Este es el caso de dos de los programas que se analizarán a lo largo del presente trabajo: “Id Tracker” y “SMART”.

Por otro lado, los enfoques basados en el aprendizaje pueden ser realizados de diferentes formas. En este caso, cabe destacar los centrados en el aprendizaje del aspecto del objeto. De este modo, rastrean objetos similares en cada fotograma o incluso aprenden el aspecto del objeto. Este es el caso de otra de las técnicas que serán estudiadas a lo largo del trabajo: “DeepLabCut”.

Un aspecto importante que destacar acerca de por qué es necesaria la creación de técnicas de evaluación de conducta en ratones, en lugar de utilizar las creadas para humanos, según Abbas y Masip-Rodo (2019), se debe a las diferencias en:

- Los patrones de movimiento. La marcha humana es diferente al movimiento realizado por los animales cuadrúpedos.
- Los entornos. Las técnicas se encuentran desarrolladas para entornos naturales humanos siendo ineficaces en entornos de laboratorio.
- La precisión de los movimientos. Para que el fenotipado del comportamiento sea fiable es necesario un seguimiento y una estimación muy precisos. Las técnicas desarrolladas para humanos no son aplicables para animales pequeños o roedores.

Por tanto, las técnicas desarrolladas para seres humanos, a pesar de encontrarse más avanzadas, no pueden aplicarse directamente sobre animales de laboratorio como los roedores. Sin embargo, sí se han utilizado como base para desarrollar las técnicas necesarias en el ámbito de laboratorio.

Sin embargo, entre los principales problemas se halla que, pese haber logrado avanzar con el desarrollo de técnicas automatizadas de evaluación del comportamiento en ratones de forma individual (Giovannucci et al., 2017), no ocurre lo mismo cuando se requiere evaluar la conducta de dos o más ratones. Algunos programas han aparecido tratando de solventar el problema, sin embargo, resulta complicado. Esto se debe a las múltiples dificultades asociadas al seguimiento de las diferentes identidades a lo largo del vídeo (Pereira et al., 2022).

Para solucionar dicho problema, comienzan a aparecer diversos *softwares* con el objetivo de realizar el seguimiento de varios animales al mismo tiempo (Barbera et al., 2020; Dandan y Lin, 2016; Hong et al., 2015; Klibaite et al., 2018; Page et al., 2009; Pereira et al., 2022; Robie et al., 2017; Thanos et al., 2015). Sin embargo, para la exploración de las técnicas automatizadas en la evaluación del comportamiento social en ratones, se descarta el uso de los anteriormente mencionados por las siguientes razones: no tienen acceso gratuito; las condiciones requeridas no son gratuitas (Barbera et al., 2020); sus condiciones de análisis no se ajustan a los objetivos (Dandan y Lin, 2016; Hong, et al., 2015); no se encuentran preparados para el análisis de la conducta social en ratones (Klibaite et al., 2018); incapacidad de acceder al *software* (Thanos et al., 2015) o tener las mismas condiciones de análisis que otro software elegido (Tabris, et al., 2022).

Técnicas elegidas para el análisis.

Las técnicas elegidas para evaluar el comportamiento social en ratones son “SMART”, “Id Tracker” y “DeepLabCut”. El primer *software* mencionado es el *software* que se utiliza actualmente en el *Laboratory of Animal Behavior Phenotype* (L.A.B.P). Se trabajará con la nueva versión que permite el seguimiento de varios ratones al mismo tiempo. Se trata de un *software* de pago, sin embargo, dado que es el utilizado actualmente, se precisa realizar la comparación de su funcionalidad con sus alternativas gratuitas.

“Id Tracker” (Pérez-Escudero et al., 2014) es un *software* gratuito dedicado al seguimiento de diferentes individuos dentro de un grupo. Este fue originalmente creado para el seguimiento de movimiento en bancos de peces, así como animales que viven en grandes grupos. Es elegido como alternativa gratuita a “SMART” debido a su similitud. Ambos realizan comparación de imagen versus fondo. Además, el análisis se describe como rápido y sencillo por parte de los creadores (Pérez-Escudero et al., 2014). En caso de lograr ser adaptado al uso con ratones y mostrar un nivel aceptable de precisión en el análisis podría sustituir tanto la medición manual como la medición con el *software* utilizado actualmente en el L.A.B.P.

Finalmente, “DeepLabCut” (Mathis et al., 2018) es un método de seguimiento que funciona a través del *deep learning*. Este permite un análisis más profundo dado que

no sólo sigue al animal, sino que pueden identificarse aquellas partes del cuerpo en las que se requiere una mayor atención.

Mediante esta técnica se enseña al programa a reconocer los patrones, animales, zonas críticas, etc., de modo que el análisis es mucho más preciso, así como especializado en aquellas áreas preferidas (Hao et al., 2016). Debido a estas características, fue seleccionada como tercera herramienta comparativa. Si bien no se encuentra tan asentada como las anteriores, se espera una mayor precisión en los resultados del análisis.

Cabe recordar que actualmente el análisis que es utilizado en el seguimiento de los animales o ratones es la comparación imagen versus fondo (Dandan y Lin, 2016), como ocurre en los otros dos *softwares* seleccionados para la comparativa. Si bien esta metodología proporciona resultados, el nivel de profundidad de estos resulta precario en comparación con los resultados que pueden obtenerse con el uso del *deep learning*.

De este modo, se pretende explorar tres técnicas automatizadas de evaluación del comportamiento social en ratones para lograr comparar la utilidad, precisión y eficiencia de cada una de ellas.

2. Objetivos.

El objetivo principal del presente Trabajo de Fin de Grado es determinar la técnica automatizada de análisis más adecuada para evaluar el comportamiento social en ratones.

Por lo tanto, para lograr el objetivo principal, se proponen los siguientes objetivos específicos:

1. Explorar tres técnicas automatizadas (“SMART”, “Id Tracker” y “Deep Lab Cut”) de evaluación del comportamiento social.
2. Adaptar las técnicas para el análisis del comportamiento en ratones.
3. Comparar la utilidad de cada una de las técnicas elegidas.
4. Comparar la precisión del análisis en cada una de las técnicas elegidas.
5. Comparar la eficiencia en cada una de las técnicas elegidas (en función del tiempo requerido para el análisis).

6. Proporcionar pautas acerca del uso de los diferentes *softwares* utilizados.

3. Metodología.

Diseño.

El diseño del estudio es del tipo metodológico experimental. Por lo tanto, se proporcionará información acerca de la metodología en base a la experimentación realizada. Los protocolos desarrollados de descarga, instalación y uso se encuentran en los anexos del II al IX.

Los elementos con los que ha trabajado son los siguientes *softwares*: “SMART”, “Id Tracker” y “DeepLabCut”. Cuyas características se encuentran detalladas en la siguiente tabla (Tabla 1):

Tabla 1.*Softwares utilizados y sus características.*

	SMART	ID TRACKER	DLC
Instalación	<ul style="list-style-type: none"> - Compra del producto. - Instalación sencilla. 	<ul style="list-style-type: none"> - Descargar e instalar Matlab Compiler RunTime de la página oficial. - Instalación sencilla. 	<ul style="list-style-type: none"> - Descargar e instalar: anaconda y python3. - Descargar “conda file” de la página oficial. - Instalación de complicación alta.
Tiempo de procesamiento	<ul style="list-style-type: none"> - Minutos. 	<ul style="list-style-type: none"> - Depende de la duración del vídeo. 	<ul style="list-style-type: none"> - Depende del procesador. - Suele tomar más de media hora.
Nivel de análisis	<ul style="list-style-type: none"> - Medio. 	<ul style="list-style-type: none"> - Bajo. 	<ul style="list-style-type: none"> - Alto.
Características del vídeo (FPS, mínimo de minutos)	<ul style="list-style-type: none"> - Flexibilidad con tipo de cámara y vídeo. 	<ul style="list-style-type: none"> - Se recomiendan 30 minutos de vídeo. - 25-49 fps. - Cámara resolución 1080x1080 pixeles. 	<ul style="list-style-type: none"> - Flexibilidad con el tipo de cámara. - Mínimo 100-200 fotogramas.
Ventajas	<ul style="list-style-type: none"> - Interfaz fácil. - Servicio de atención al usuario. 	<ul style="list-style-type: none"> - Simple. - Gratuito. - Menores requisitos informáticos. 	<ul style="list-style-type: none"> - Gratuito. - Mayor nivel de análisis.
Inconvenientes	<ul style="list-style-type: none"> - Servicio de pago. 	<ul style="list-style-type: none"> - Menor nivel de análisis. 	<ul style="list-style-type: none"> - Lenguaje informático muy técnico. - Supera las posibilidades de un ordenador promedio.
Requisitos informáticos.	<ul style="list-style-type: none"> - 2GHz procesador multicore (Intel Core 2 Duo, AMD Phenom X2 o más. - 4GB RAM. 	<ul style="list-style-type: none"> - Windows 7 o XP. - Necesario cerca de 500 MB libres en el disco duro. - 8 GB de RAM. 	<ul style="list-style-type: none"> - NVIDIA GPU, 8 GB de memoria RAM. - AMD/Intel/Nvidia GPU tiene otra instalación posible. - Linux (Ubuntu), MacOS (Mojave), Windows 10.

Muestra.

Para el estudio se recibieron varios vídeos de dos ratones C57BL/6 y un vídeo de ratones CD1 en una arena de 45x45cm sin estímulos de modo que pudiera cuantificarse la interacción social entre ambos. Los vídeos fueron proporcionados por la Dra. Ana Pérez Villalba.

Instrumentos.

Para la visualización y análisis de los vídeos proporcionados se hizo uso de dos ordenadores portátiles. Las características de estos son:

Primer ordenador (ordenador 1):

- 8 GB de RAM.
- 256 GB memoria externa.
- IntelCore i5, 2.50 GHz.
- Windows 11.
- CPU.

Segundo ordenador (ordenador 2):

- 16 GB de RAM.
- 500 GB memoria externa.
- IntelCore i7, 2,70 GHz.
- Tarjeta gráfica NVIDIA GeForce RTX 3060, 6 GB memoria dedicada.
- Windows 10.
- GPU.

Con la capacidad indicada en el primer ordenador fue posible analizar los vídeos proporcionados en “Id Tracker”. Sin embargo, para analizar el material en “DeepLabCut” fue necesario hacer uso del segundo ordenador. Además, se solicitó el superordenador Tirant (Universitat de València, 2023). En este caso, se puede apreciar una mayor potencia de procesamiento en el superordenador:

- Dos procesadores con 32 GB de RAM.
- 10TB de memoria principal.
- Intel Xeon SandyBridge E5-2670 a 2,6 Ghz.

- OpenSuSE Leap 42.3.

Para analizar los vídeos en “SMART” se hizo uso del segundo ordenador. Aun así, debido a sus características podría ser utilizado en el ordenador 1 sin problema.

Para el análisis de los datos proporcionados en “Id Tracker” se hizo uso del sistema MatLab, un lenguaje de alto rendimiento para la computación técnica. Incluye computación, visualización y programación. (Gilat, 2006).

Respecto a las grabaciones, se adjuntan en el anexo (véase anexo I) una lista de enlaces para la visualización de los vídeos proporcionados para su posterior análisis.

El vídeo 1 se proporcionó como ejemplo para poder realizar un análisis de prueba con los diferentes *softwares*.

El vídeo 2 fue proporcionado de forma adicional. En este los ratones tienen la cola pintada de color rojo y blanco para una diferenciación más simple. Las razones para esto se explican más adelante.

El vídeo 3 fue utilizado con el objetivo de solventar nuevos problemas hallados en el análisis. Para una mejor diferenciación por parte de los programas de análisis se trató de hacer uso de ratones blancos con el pelo tintado.

El espacio en el que se localizaban los ratones era una arena cuadrada de 45x45cm, esta se puede observar en los vídeos del anexo I. Dos ratones eran posicionados en esta durante aproximadamente cinco minutos. Todo este tiempo siendo grabados con una cámara GoPro situada en la zona superior de la arena.

Procedimiento.

SMART.

Este es el único *software* de pago con el que se ha trabajado. Por tanto, era necesaria la aprobación de la compra por parte de OTRI antes de poder adquirirlo. Los tiempos de espera se alargaron demasiado, por ello, se terminó adquiriendo la versión de prueba disponible en la página oficial de los creadores.

Tras su descarga se recibió un breve tutorial por parte de la tutora dado que es la herramienta de uso en el L.A.B.P.

La instalación es sencilla, simplemente es necesaria la descarga del programa y su ejecución. Por otro lado, el análisis, en la versión de prueba permitía explorar todas las posibilidades ofrecidas por el programa.

Los pasos más detallados acerca de su uso pueden verse en el anexo IX. Se analizó desde un principio el vídeo 2. Esto se debe a que fue el último programa con el que se trató de realizar el análisis, por ello, ya se conocían las limitaciones presentes los otros programas. Para el análisis se seleccionaron aquellos parámetros que permitían medir la conducta social entre dos ratones. Cabe destacar que se realizaron dos análisis previos de modo que fuera posible familiarizarse con la herramienta y, así, asegurar también la calidad de la evaluación.

Una vez se obtuvieron los resultados, se observó un problema en la medición relacionado con el umbral de detección. Por lo tanto, se volvió a analizar el vídeo, ajustando los parámetros. Además, dado que se contaba con la versión completa, al ser una prueba, se hizo uso de la función *TriWise*. Esta permite establecer tres puntos de detección en el ratón, asegurando así una mayor precisión en los resultados del análisis.

Los resultados obtenidos en el tercer análisis permitieron la comparación con la medición manual del número de contactos por minuto, la latencia de los contactos, así como el porcentaje total de interacción a lo largo de la evaluación.

Id Tracker.

El primer paso fue la creación de un protocolo acerca de cómo descargar, instalar y hacer uso del programa, al mismo tiempo que se realizaban estos pasos.

La descarga del programa es simple a través de su página web (Pérez-Escudero, et al., 2014). Las indicaciones más detalladas acerca de la instalación se pueden leer en el anexo II.

El primer vídeo (vídeo 1) se analizó en el ordenador 1. Los pasos más detallados acerca de cómo se realizó la primera exploración del *software* se pueden leer en el anexo V. Tras lograr realizar la evaluación del vídeo, se obtuvieron una serie de resultados proporcionados en dos tipos de archivo, uno para Excel y otro para su análisis con MatLab.

El siguiente objetivo fue lograr adaptar los datos resultantes a las necesidades de la investigación. En el archivo de MatLab se proporcionaban dos columnas, cada una para un ratón. Por otro lado, las filas hacen referencia a los “frames” del vídeo. Con estos datos, de acuerdo con “Id Tracker” era posible crear una imagen que resumiera los resultados. Sin embargo, no fue posible crear ninguna imagen adecuada para las necesidades.

Por lo tanto, se hizo uso del archivo en formato dedicado para Excel en MatLab. Con este sí fue posible analizar mejor los datos. Este archivo proporcionaba cuatro columnas y 9000 filas (los “frames”). Las dos primeras columnas proporcionaban las coordenadas del primer ratón, la tercera y cuarta las coordenadas del segundo ratón.

Con esto, fue posible crear imágenes del recorrido individual de cada ratón (imágenes 1 y 2), los recorridos superpuestos (imagen 3), así como un vídeo (gráfico en movimiento) donde se podía observar a tiempo real el recorrido del ratón (véase anexo 1, enlaces 4 y 5). Sin embargo, de nuevo, esto no proporcionaba los datos necesarios para cumplir los objetivos.

Imagen 1.

Recorrido primer ratón.

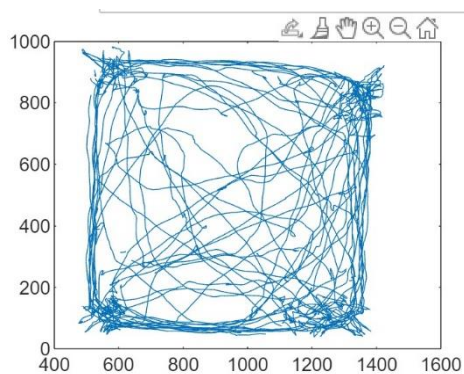
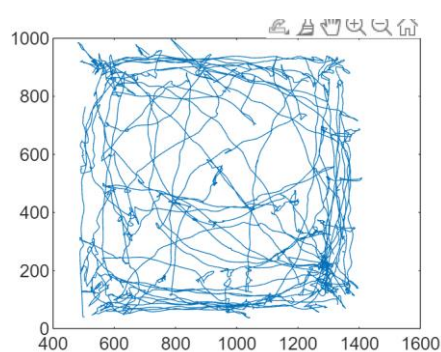


Imagen 2.

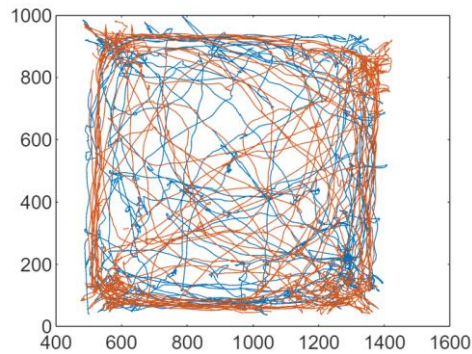
Recorrido del segundo ratón.



Nota. En las imágenes se puede apreciar el recorrido total realizado por el ratón 1 (imagen 1) y el ratón 2 (imagen 2). Las imágenes fueron creadas en MatLab mediante las coordenadas de los ratones en vídeo 1.

Imagen 3.

Recorrido de ambos ratones



Nota. En la imagen se muestra el recorrido del ratón 1 y el recorrido del ratón 2, uno encima del otro. Se creó la imagen con el recorrido de ambos ratones superpuesto con el objetivo de tratar de visualizar los contactos. Sin embargo, como se puede observar en la imagen no resulta funcional.

Esto llevó a la búsqueda de soluciones alternativas, una opción consistía en tomar las coordenadas de los ratones 1 y 2 restando la diferencia entre sus respectivas coordenadas. Es decir, a la coordenada X del ratón 1 se le restaba la coordenada X del ratón 2. Lo mismo con sus coordenadas Y. Se indicó que, si la diferencia entre ambas era inferior al número seleccionado, debía crearse una nueva tabla. En esta tabla, se ordenó lo siguiente:

- Si la diferencia entre X_1 y X_2 (ratón 1 y ratón 2) era inferior a 45, esto era igual a 1, si no, 0. (En lenguaje de programación 1 = “True”, 0 = “False”). Estos resultados aparecieron en una columna.
- Si la diferencia entre Y_1 e Y_2 era inferior a 45, esto era igual a 1, si no 0. Estos resultados aparecieron en la segunda columna.
- Para poder comparar esto con las coordenadas reales de los ratones, se programó que aparecieran las coordenadas del ratón 1, en las columnas 3 y 4, y las coordenadas del ratón 2, en las columnas 5 y 6.

Se puede apreciar parte de la tabla creada en la imagen 4.

Imagen 4.

Parte de la tabla creada con las coincidencias.

	A	B	C	D	E	F
1	0	0	738,95	888,77	576,95	187,42
2	0	0	749,82	889,86	574,96	186,09
3	0	0	762,11	890,58	578,49	187,77
4	0	0	766,02	891,17	578,13	188,28
5	0	0	777,91	893,2	576,09	187,39
6	0	0	790,81	894,74	577,28	186,99
7	0	0	796,57	895,47	577,05	186,74
8	0	0	807,72	897,55	576,76	186,88

Nota. En la imagen se puede observar un trozo de la tabla creada con las coincidencias, es decir, contactos entre ratones. Las dos primeras columnas hacen referencia a las coordenadas X (columna A) e Y (columna B). En este caso, al no existir coincidencias observamos que ambas muestran un 0. Las columnas C y D representan las coordenadas del ratón 1. Las columnas E y F las coordenadas del ratón 2. C y E hacen referencia a las coordenadas X de cada ratón respectivamente. D y F hacen referencia a la coordenada Y de cada uno de los ratones.

Se esperaba poder obtener la información deseada con estos resultados, sin embargo, en la comparativa con el vídeo se observa que los “contactos” realmente eran conductas de seguimiento. Precisamente, cuando ambos ratones se tocan el *software* tan solo nos proporciona “NaN”, es decir, una ausencia de datos. Por esto, para solucionar el problema se proporcionó el segundo vídeo (vídeo 2) en el que los ratones tenían las colas pintadas con colores. El objetivo era asegurar una mayor diferenciación entre ambos de modo que “Id Tracker” pudiera proporcionar datos en el contacto.

Se trató de analizar el vídeo del mismo modo que el anterior pero el problema de “NaN” seguía ocurriendo. Así pues, se trató de ajustar los valores para asegurar lo máximo posible un análisis detallado. Con los nuevos datos, se trató de realizar el análisis con MatLab de nuevo. En este caso, se ajustó lo máximo posible la diferencia. Se observó que lo más indicado era un número inferior.

Así pues, se compararon los nuevos datos obtenidos con el vídeo. En este caso, sí coinciden. Cuando se obtiene 1, en las columnas 1 y 2, los ratones se encuentran interaccionando en el vídeo. Ahora bien, se vuelve a encontrar el problema de los datos en “NaN”. En algunas interacciones sigue apareciendo. Se pudo apreciar que algunos de estos “NaN” aislados se debían a la posición del ratón. Por ejemplo, cuando se

encontraban explorando de forma vertical u horizontal, como se puede observar en las imágenes 5 y 6.

Imagen 6.

Ratón explorando de forma vertical.



Imagen 5.

Ratón en exploración horizontal.



Paralelamente, dados los problemas de análisis, se proporciona el vídeo 3. En este, se cuenta con dos ratones blancos, con el pelo tintado de rojo y de azul. Se espera que esta adaptación facilite el análisis con el *software*, disminuyendo o erradicando la cantidad de “NaN” hallados.

Sin embargo, lo que se encuentra es que, los ratones blancos, sobre un fondo de color similar, imposibilitan por completo su análisis al este depender del contraste con el fondo.

DeepLabCut.

Al igual que con “Id Tracker”, el primer paso fue la creación de un protocolo acerca de su instalación al mismo tiempo que se realizaba esta. En el caso de “DeepLabCut” la instalación resulta mucho más complicada, con muchos más pasos a seguir y requiere de conocimiento de programación, lenguaje “python”, entre otros. La instalación para cada uno de los ordenadores se puede encontrar en los anexos III y IV.

Una vez instalado, en el ordenador 1, el que se tenía inicialmente, se realizó una prueba de análisis de vídeo 1 proporcionado por la Dra. Ana Pérez. Los pasos detallados para analizar el vídeo se pueden encontrar en el anexo VI. Con el primer ordenador todos los intentos de análisis llevaron a error tras horas de análisis (se llegó a dejar el ordenador hasta 35 horas analizando sin ningún uso secundario). También se trató de analizar el

vídeo 2 con el ordenador 1 hallando los mismos problemas. Por ello, se descartó la posibilidad de uso de “DeepLabCut” con un ordenador promedio. La solución hallada fue solicitar el uso del superordenador Tirant. Sin embargo, dado el orden de prioridad se quedó en la lista de espera.

Paralelamente a esta espera se terminó adquiriendo el ordenador 2. Si bien este no tenía las características ideales de uso de “DeepLabCut” se esperaba poder, al menos, analizar los vídeos ya que contaba con una GPU. En el ordenador 2, se intentó analizar el vídeo 2 directamente. En este, tras lograr la adecuada instalación, se logró un análisis adecuado del vídeo. El siguiente paso era adaptar los resultados a los objetivos de la investigación.

Se terminó hallando que “DeepLabCut” no era capaz de proporcionar el análisis deseado. Para analizar el comportamiento social es necesario bien, adaptar los resultados por cuenta propia si se tienen conocimientos de programación, o buscar un *software* compatible que pueda analizar la información proporcionada por “DeepLabCut”. La propia página de “DeepLabCut” (DeepLabCut, 2021) recomienda algunos *softwares* complementarios. Se terminó determinando Simple Behavior Analysis (“simBA”) (Nilsson, et al., 2020) como el más adecuado.

“simBA” utiliza los datos proporcionados por “DeepLabCut” para crear una serie de clasificadores del comportamiento social en roedores. Es decir, es posible programarlo para que aprenda diferentes conductas y las analice de acuerdo con las necesidades del investigador. Para ello, se utiliza el aprendizaje automático supervisado. Gracias a esto, se pueden obtener resultados tan precisos que superar los realizados por observadores humanos (Nilsson, et al., 2020). Sin embargo, existe una limitación importante. Su capacidad de diferenciación entre ratones se ve considerablemente afectada en aquellos del mismo color, se recomienda utiliza ratones con diferentes tonalidades.

Los pasos para realizar la instalación y el análisis pueden leerse en los anexos VII y VIII. Tras la instalación, se realizó el análisis con el vídeo 2 y el vídeo 3 proporcionados por la tutora. El *software* realizó un análisis considerablemente preciso proporcionando la información que se necesitaba (número de contactos, latencia y contactos por minuto). Aun así, la precisión en la diferenciación de los ratones se veía afectada en el vídeo 2, al

ser dos ratones de pelo oscuro. De todos modos, el análisis del comportamiento no parecía verse afectado.

De este modo, se puede proceder a la evaluación de los resultados obtenidos con cada uno de los *softwares* explorados.

4. Resultados.

La comparación de los resultados se realizó con el estándar actual, una contabilización de la interacción de forma manual. De este modo, se pudo comparar de forma objetiva los resultados obtenidos en cada uno de los *softwares* con los resultados que se obtendrían de forma manual.

En los criterios de evaluación manual se consideró “interacción” cada vez que existía contacto real entre ambos ratones, es decir, sus cuerpos se encontraban tocando de forma intencionada el cuerpo del otro ratón. No se contabilizó como “interacción” el contacto entre las colas. Si las interacciones ocurrían de forma seguida, con una diferencia inferior a un segundo, entre interacción e interacción, se contó como una sola interacción. Si existía una diferencia de tiempo mayor como dos interacciones.

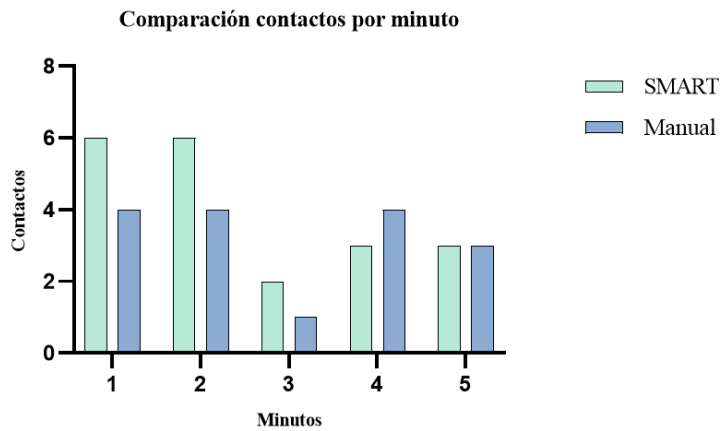
SMART.

Los resultados obtenidos permitieron el cálculo del número de interacciones de forma relativamente rápida. Si bien la información no se presentó de forma directa (con el cálculo de interacciones por minuto y el tiempo total) sí se proporcionó toda la información necesaria.

Entre los resultados, se presentó una descripción de las acciones de los ratones, así como la duración de estas. Mediante esta información se pudo obtener los siguientes datos:

Gráfica 1.

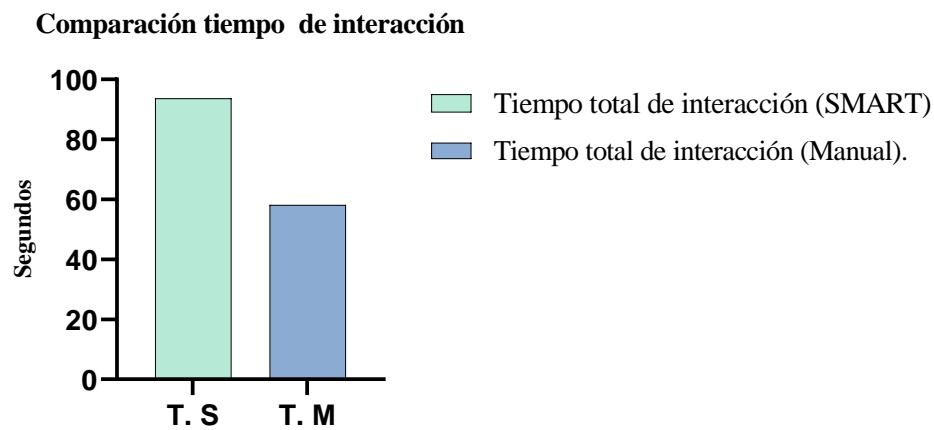
Comparación resultados de SMART vs. cálculo manual.



Nota. En la gráfica se observa la comparación de los resultados obtenidos en la medición de los contactos por minuto con el software SMART y la medición manual.

Gráfica 2.

Comparación del tiempo total de interacción en "SMART".

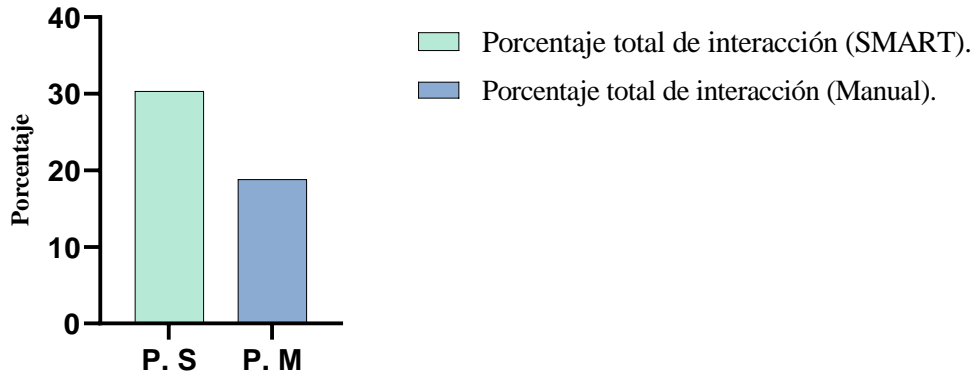


Nota. En la gráfica se puede observar la comparación entre el tiempo total de interacción obtenido con el software "SMART" (T.S) y el tiempo obtenido de forma manual (T.M).

Gráfica 3.

Comparación del porcentaje de interacción en “SMART”.

Comparación pocentaje de interacción



Nota. Se muestra el porcentaje total de interacción de “SMART” (P. S) y el porcentaje total de interacción obtenido de forma manual (P.M).

El tiempo empleado para el análisis fue relativamente bajo. La configuración junto con el procesamiento tomó cerca de 15 minutos. Por otro lado, la recogida de información acerca las interacciones duró 5 minutos más. Esto proporciona un total de 20 minutos para el análisis y la obtención de resultados con “SMART”.

Respecto a la precisión del análisis, se encuentra en un nivel medio. No es demasiado precisa, pero el error mostrado es bajo en comparación con otros *softwares* de análisis.

De este modo, el nivel de eficiencia puede ser considerado medio, teniendo en cuenta la rapidez para la obtención de resultados junto con el nivel de precisión.

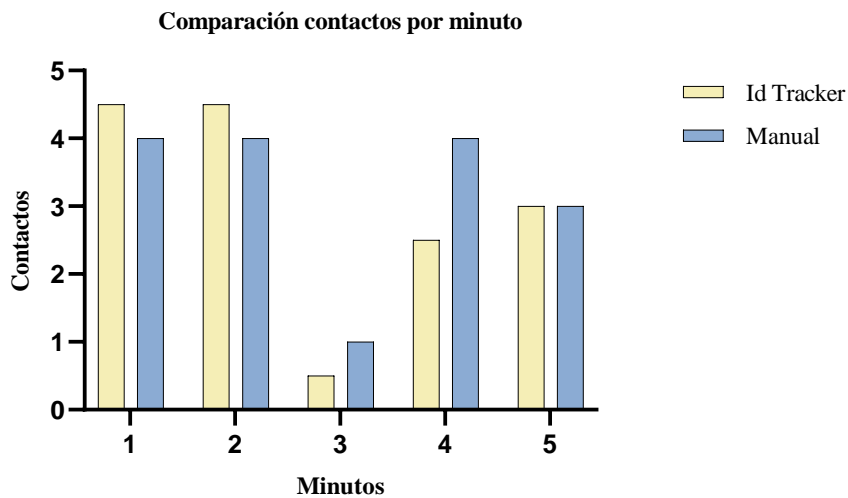
Id Tracker.

Los resultados de la fórmula proporcionados en Excel permitieron el cálculo del número total de contactos, número de contactos por minuto, duración total de los contactos y el cálculo del porcentaje de tiempo de contacto respecto a la duración del vídeo.

De este modo, en la comparación con el cálculo manual se obtuvieron los siguientes resultados:

Gráfica 4.

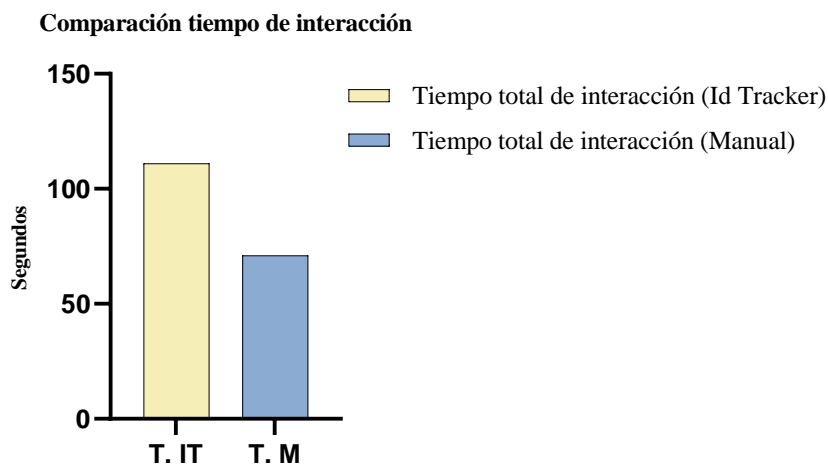
Comparación de resultados de Id Tracker vs. cálculo manual.



Nota. *En la gráfica se observa la comparación del número de contactos obtenidos a través de la medición con el software "Id Tracker" con la medición manual de los contactos. En el eje "X" se encuentran los minutos, mientras en el "Y" se observan los contactos realizados en ese minuto.*

Gráfica 5.

Comparación del tiempo total de interacción en "Id Tracker".

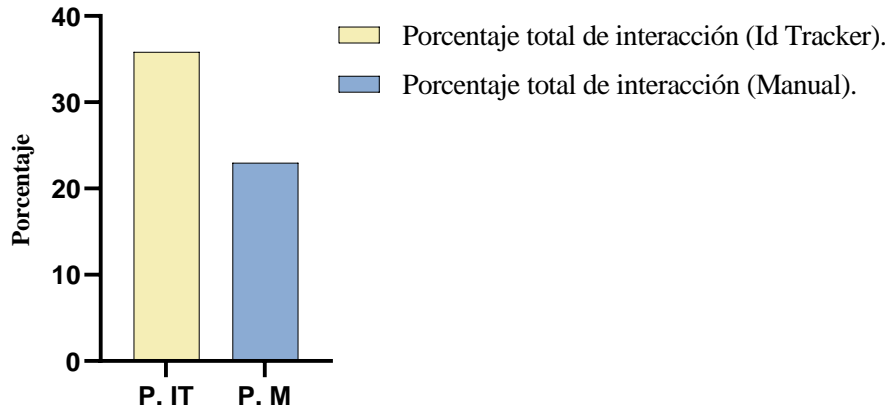


Nota. *En la gráfica se muestra la comparación entre el tiempo total de interacción obtenido con "Id Tracker" (T.IT) y el tiempo total de interacción obtenido en la medición manual (T. M).*

Gráfica 6.

Comparación del porcentaje de interacción en “Id Tracker”.

Comparación porcentaje de interacción



Nota. *Se muestra el porcentaje total de interacción en cada uno de ellos. “P.IT” muestra el porcentaje total de interacción de “Id Tracker”, “P.M” el porcentaje total de interacción de la medición manual.*

Respecto al tiempo empleado para el análisis, tomó un total de 15 minutos analizar el vídeo con “Id Tracker” y entre 15 y 20 minutos más analizar los resultados obtenidos. Es decir, pasar los resultados a MatLab para obtener el documento Excel donde obtener las interacciones, así como el tiempo total.

De este modo, el análisis de vídeo junto con interpretación de resultados con “Id Tracker” tomó un total entre 30 y 35 minutos.

Por otro lado, la precisión del análisis de “Id Tracker” es considerablemente limitada. Los resultados, contienen numerosos fallos por falta de información. Además, la interpretación ha sido propia, con la creación de una fórmula. Por lo tanto, la validez de estos también se encuentra cuestionada.

De este modo, se puede considerar el nivel de eficiencia (nivel de análisis versus tiempo requerido) como deficiente.

DeepLabCut / simBA.

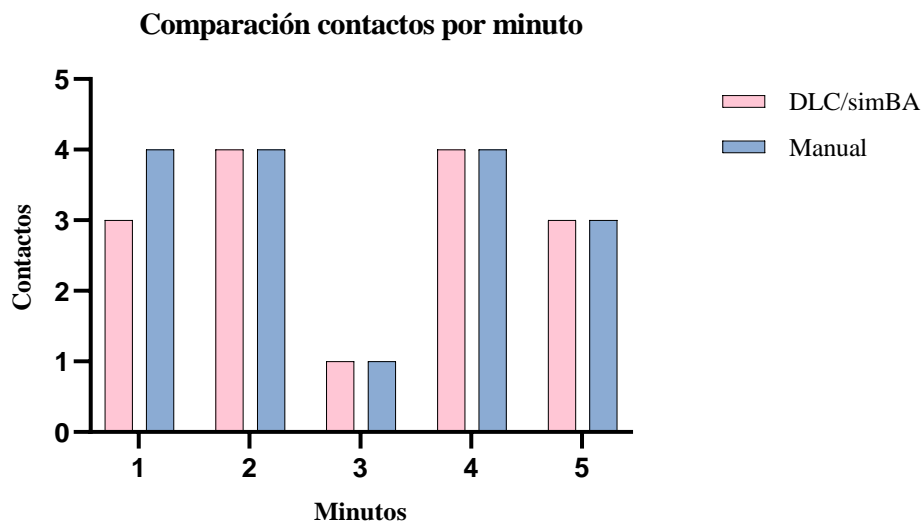
Los resultados obtenidos con “simBA” permitieron la comparación del número de interacciones por minuto y el cálculo del tiempo total de interacción a través de dos fuentes distintas. Para el número de interacciones por minuto se utilizó el gráfico *gannt* (se puede observar en el anexo I, vídeo 8). Este muestra cada interacción y su duración en tiempo real durante el vídeo. Por otro lado, para el total de interacción se utilizó el vídeo con información acerca de las interacciones creado con “simBA” (vídeo 7).

Se contó como interacción tan sólo el comportamiento 2, aquel en el que ambos ratones se encontraban en contacto real. Además, aquellos comportamientos de poca duración, cercanos a un contacto más largo, se contaron como un solo contacto debido a la gran precisión de la técnica de evaluación. Esta es capaz de diferenciar como dos contactos diferentes contactos con milésimas de segundo de diferencia.

De este modo, se pudieron comparar los resultados, como se puede observar en las gráficas 7, 8 y 9:

Gráfica 7.

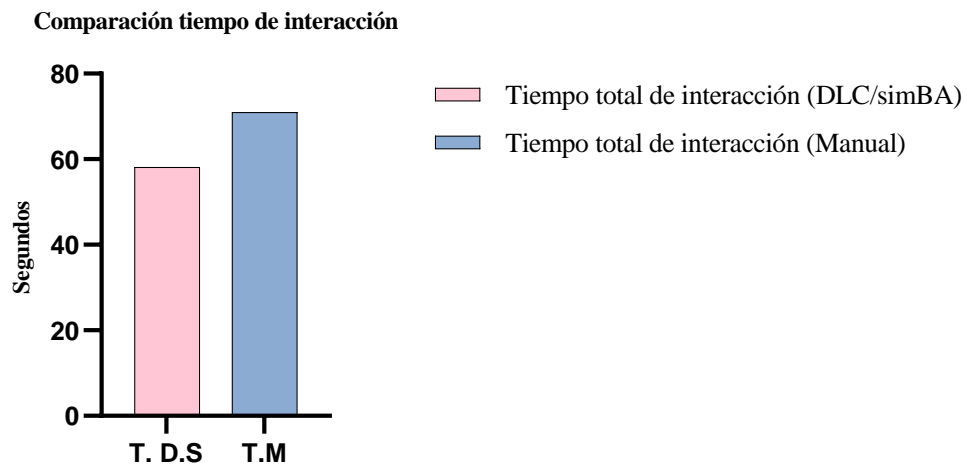
Comparación de resultados “DLC/simBA” vs. cálculo manual.



Nota. *En la gráfica se muestra la comparación de los resultados obtenidos mediante los softwares “DeepLabCut” y “simBA” en contraposición a los resultados obtenidos mediante el conteo manual de las interacciones.*

Gráfica 8.

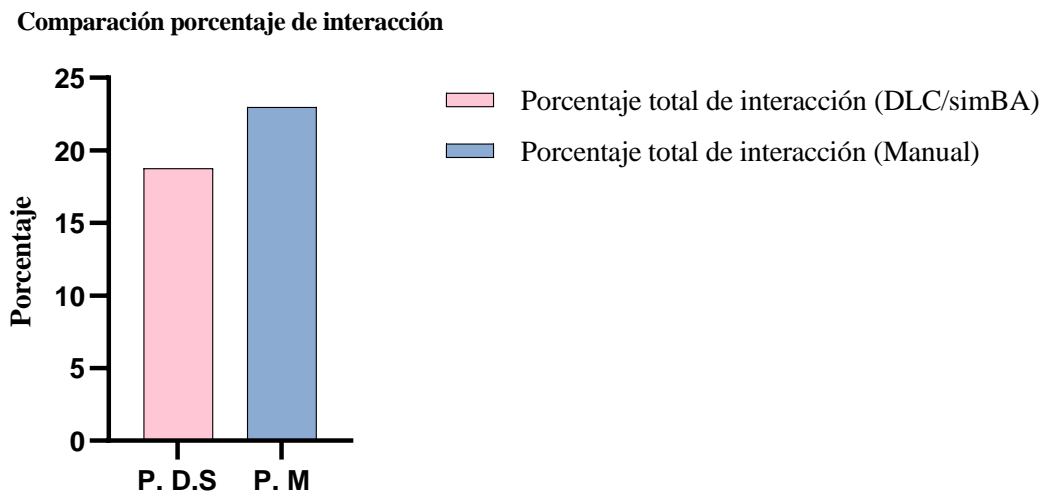
Comparación del tiempo y porcentaje total de interacción en “DLC/simBA”.



Nota. En la gráfica se puede observar la comparación del tiempo total de interacción entre la medición con “DeepLabCut” y “simBA” (T.D.S) versus la medición manual de este (T.M).

Gráfica 9.

Comparación porcentaje de interacción en “DLC/simBA”.



Nota. Se muestra el porcentaje total de interacción con la medición de “DeepLabCut” y “simBA” (P.D.S) y el porcentaje obtenido de forma manual (P.M).

Respecto al tiempo empleado en el análisis y obtención de resultados, tomó un total de 50 minutos analizar el vídeo con “DeepLabCut”. Por otro lado, el transformar esta

información a través de “simBA” para obtener los datos necesarios requirió cerca de 20 minutos más. Esto conlleva un total de 70 minutos para el análisis del vídeo y obtención de resultados.

Por otro lado, la precisión de los resultados con “DeepLabCut” y “simBA” es considerablemente buena. Es capaz de distinguir los comportamientos hasta el milisegundo, superando por mucho la medición manual.

De este modo, se puede considerar que la eficiencia (nivel de análisis versus tiempo requerido) es aceptable.

5. Discusión.

Tras obtener los resultados y compararlos con la medición manual es preciso revisar los objetivos planteados en un principio.

Se han explorado las tres técnicas de análisis automatizadas, tanto “SMART” como “Id Tracker” y “DeepLabCut”. Por lo tanto, se puede considerar como cumplido el primer objetivo. Además, a estas tres técnicas ha terminado por agregarse una cuarta (“simBA”) necesaria para poder analizar realmente con la tercera técnica (“DeepLabCut”).

Respecto al segundo objetivo, donde se pretendía adaptar las técnicas de análisis al comportamiento en ratones, se considera cumplido de forma parcial. “SMART” ya había sido creado para el uso con ratones por lo que no ha representado ningún problema. “Id Tracker” pudo ser utilizado para el análisis con ratones de color oscuro mas no para aquellos de tonalidad clara. Por esto mismo, los análisis se vieron limitados. Respecto a “DeepLabCut” no presentó problema alguno para el análisis ya que fue posible adaptar las características para ratones. Finalmente, en “simBA”, si bien el análisis con ratones de pelo claro fue excelente, no ocurrió lo mismo con ratones con tonalidades oscuras. Esto no representó una limitación para el análisis de las conductas sociales en los objetivos, pero sí podría ser una limitación en futuros estudios.

El tercer objetivo se centraba en comparar la utilidad de cada una de las técnicas elegidas. Tanto este como los siguientes objetivos (cuatro y cinco) se relacionan entre

sí por lo que serán abordados al mismo tiempo. En el objetivo número cuatro se pretende comparar la precisión del análisis en cada una de las técnicas elegidas y, en el cinco, se pretende comparar la eficiencia.

Las cuatro técnicas son útiles ya que han permitido el análisis de la conducta de los ratones de forma no manual. Además, el tiempo de análisis no ha sido excesivamente elevado en ninguna de ellas por lo que pueden ser utilizadas en lugar de la medición manual. Mas, en la utilidad destacan “SMART” y la combinación de “DeepLabCut” con “simBA”.

“SMART” ofrece resultados rápidos con una interfaz sencilla. Además, proporciona todos los datos deseados de forma clara, en un formato fácil de interpretar y manejar.

Por otro lado, “DeepLabCut” junto con “simBA” requieren más tiempo y la interfaz no es tan sencilla. Aun así, se considera que tiene la capacidad de ser tan útil como “SMART” o incluso llegar a sustituirlo. Esto se debe a que tiene una precisión mayor, como se comentará en el siguiente punto. Además, son dos *softwares* gratuitos, por lo que son accesibles para todo aquel que lo necesite. Un punto esencial que destacar también es el hecho de que se pueda programar para que analice tantas conductas como se desee, de este modo abriendo el abanico de posibles funciones y usos futuros. Su utilidad, por tanto, queda justificada.

En este caso, “Id Tracker” queda atrás en utilidad ya que los resultados son limitados y poco fiables. Además de proporcionar un análisis mucho más limitado.

En la precisión del análisis también es necesario realizar distinciones.

“SMART” trabaja mediante la comparación figura vs. fondo, esto quiere decir que no aprende a diferenciar en sí a los animales. Además, también se recomienda el uso de ratones con marcas de distinción claras. Por tanto, la precisión del análisis queda considerablemente comprometida. Esto se ha podido observar tanto en el trabajo con el *software* como con los resultados obtenidos. Estos mostraban una diferencia importante con la medición manual tanto a nivel de contactos como en el tiempo total de interacción.

Una situación similar ocurre con “Id Tracker” donde se hallan serias limitaciones. Las identidades se perdían en los casos en los que el contacto era especialmente estrecho. Existe una cantidad significativa de resultados en los que faltan datos. No se ha proporcionado resultados de forma directa a través del *software*, sino que fue necesario crear una ecuación para el cálculo propia lo que pone en compromiso la

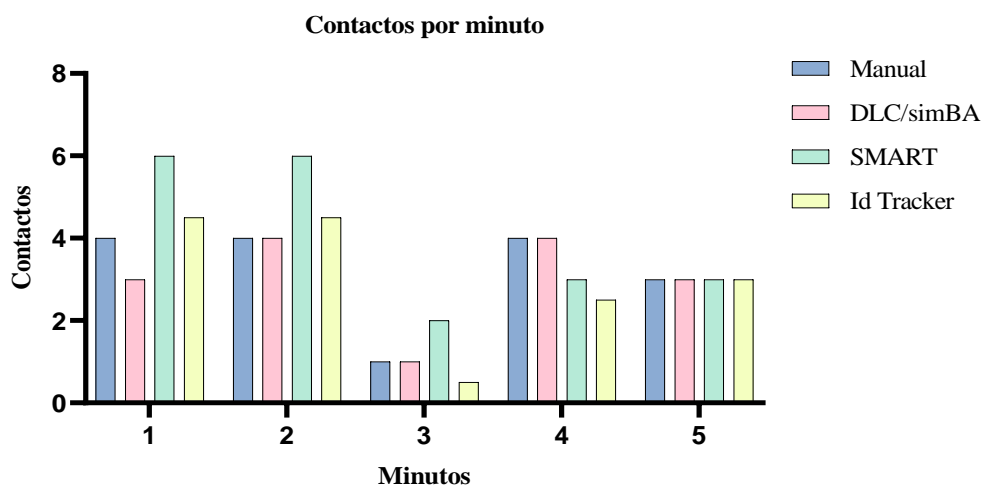
validez de los resultados. Además, los datos proporcionados eran simples, las coordenadas de cada ratón en el tiempo, no era capaz de proporcionar ningún otro dato ni eran adaptables de forma sencilla. Finalmente, la interfaz tan sólo permite el análisis del vídeo de un modo.

Por otro lado, “DeepLabCut” presenta una precisión mucho mayor. Se pueden señalar tantas partes del cuerpo del animal como se deseen. Además, la precisión del análisis puede ser ajustada a deseo del experimentador. El error puede llegar a ser mínimo, superando por mucho las capacidades de una persona humana. En este caso, “DeepLabCut” ha demostrado una mayor precisión. Cabe destacar que, si bien en el análisis de las partes es muy preciso, no es completo. Por ello, “simBA” como complemento fue esencial. En este caso, de nuevo encontramos una capacidad de precisión elevada. Detecta tantas conductas como se desee programar. Además, estas son analizadas en niveles muy altos, de modo que puede proporcionar resultados exactos al milisegundo. En comparación con los dos anteriores softwares, por tanto, se puede considerar que la combinación de “DeepLabCut” con “simBA” es la más precisa en el análisis.

La comparación de los contactos por minuto obtenidos en cada uno de los *softwares* versus la medición manual puede verse en la siguiente gráfica:

Gráfica 10.

Comparación final de contactos por minuto.



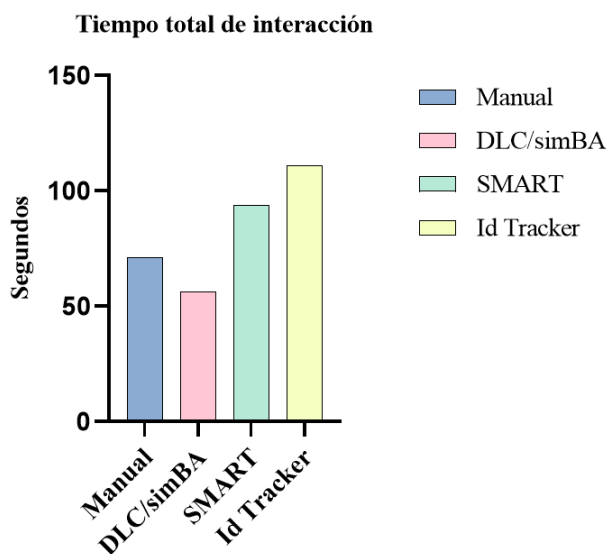
Nota. En la gráfica se realiza una comparación de los contactos por minuto detectados por tanto la medición manual como cada uno de los softwares explorados a lo largo del trabajo.

En la gráfica se puede apreciar cómo la mayor precisión parece encontrarse en la medición de “DeepLabCut” junto con “simBA”. Por otro lado, “SMART” y “Id Tracker” muestran un análisis de los contactos por minuto más errático.

Respecto al tiempo total de interacción encontramos resultados diversos:

Gráfica 11.

Comparación final del tiempo de interacción.



Nota. *En la gráfica se muestra la comparación final del análisis del tiempo de interacción obtenido en cada uno de los softwares y la medición manual.*

Cabe destacar cómo el tiempo total obtenido con “Id Tracker” es casi el doble del tiempo obtenido con “DeepLabCut” y “simBA”. Además, de nuevo se muestra cómo la medición manual se asemeja más a los resultados de “DLC” y “simBA”, mientras “SMART” y “Id Tracker” muestran semejanzas en sus tiempos totales. Esto podría deberse al estilo de medición. Tanto “SMART” como “Id Tracker” analizan por el contraste figura-fondo pudiendo haberse encontrado los mismos errores en el análisis.

Tras la comparación final de los resultados obtenidos tanto en la medición manual como en cada uno de los *softwares* explorados, se considera el objetivo número cuatro cumplido.

En el punto relacionado con la eficiencia (objetivo cinco) destaca “SMART”. Si bien el análisis no es tan preciso como el obtenido con “DLC/simBA”, es más fácil de realizar, los resultados son proporcionados de forma clara y toma una cantidad de tiempo óptima (15-20 minutos).

En comparación, “Id Tracker” no resulta eficiente. Si bien, al ser gratuito podría parecer una buena opción para sustituir “SMART”, ya que tiempo de análisis no es muy superior (35 minutos), la información proporcionada, la veracidad de esta, así como la variedad de conductas que puede analizar resultan insatisfactorias.

“DeepLabCut” y “simBA” toman más tiempo para analizar. Además, son más complicados de utilizar que “SMART” y “Id Tracker”. Sin embargo, se consideran una opción con gran potencial para desarrollar debido al nivel de análisis, la variabilidad de posibilidades y su acceso gratuito.

De este modo, dado que ha sido posible comparar todas las técnicas elegidas tanto en utilidad, precisión como eficiencia, se considera que los objetivos (3), (4) y (5) se han cumplido.

Finalmente, el sexto objetivo era proporcionar pautas acerca del uso de los diferentes softwares utilizados. Se han adjuntado todos los protocolos de descarga y uso de cada uno de los *softwares* en los anexos del II al IX. De este modo, se espera poder proporcionar la información necesaria para posibles futuros usuarios o aquellos que deseen replicar la comparación realizada en el presente trabajo. Asimismo, se logra cumplir también el sexto y último objetivo.

Todo esto, nos permite considerar el objetivo general del trabajo: determinar qué técnica es actualmente la más funcional dadas sus características. La decisión dependerá de las necesidades de análisis, así como medios del investigador. Si no se poseen conocimientos de programación, utilizar “DeepLabCut” y “simBA” a pesar de su superioridad en precisión y variabilidad de análisis no resulta funcional. Sin embargo, si se poseen conocimientos de programación o no se tiene la capacidad de adquirir un *software* de pago, como es el caso de “SMART”, estos podrían ser la opción más adecuada.

Por tanto, si se posee la capacidad económica y/o no se tienen conocimientos de programación, la opción más funcional es “SMART”. El análisis proporcionado es rápido

y la interfaz es fácil de manejar. Si bien la precisión no es tan elevada, la automatización de la evaluación permite obtener siempre los mismos resultados por lo que seguiría favoreciendo la replicabilidad de los resultados. Esto no ocurre en la medición manual, como bien se ha mencionado en la introducción. Asimismo, al ser una técnica automatizada reduce de forma significativa el tiempo total de evaluación.

En cambio, si se poseen conocimientos de programación o no se puede adquirir el *software* de pago, “DeepLabCut” junto con “simBA” son una buena opción. La precisión del análisis es muy elevada, no tiene ningún coste y presenta las mismas ventajas de automatización que presenta “SMART”. Una desventaja que destacar, más allá de los conocimientos informáticos necesarios, es el tiempo requerido para el análisis pudiendo llegar a perder su funcionalidad al ser tan elevado.

Se deben remarcar algunos aspectos acerca de los resultados en la comparación con la medición manual. Primero, se debe abordar la medición de “Id Tracker”. Se puede observar en los gráficos 5 y 6 que la medición de “Id Tracker” presenta una mayor diferencia en comparación con la manual. Entre las posibles razones y limitaciones que se asocian a esta anomalía en los datos, se consideran las siguientes:

Los datos proporcionados por el *software*, de por sí, contienen errores. Existe una gran cantidad “frames” con ausencia de datos, las identidades se pierden, etc., todo esto conlleva una medición inexacta. A esto, se debe agregar el hecho de que el cálculo de los contactos ha sido realizado a través de una ecuación de autoría propia lo que, de nuevo, afecta a la validez de las mediciones. Si bien se realizó una comparación para avalar todo lo posible que, aquellos momentos en los que se detectaba contacto en “Id Tracker” realmente coincidían con contacto en el vídeo real, es posible que la precisión de las mediciones se haya visto afectada.

Por otro lado, se considera importante abordar las diferencias, aunque menores, halladas entre el análisis de “simBA” y el manual. Los datos coinciden casi a la perfección excepto por el primer minuto, donde manualmente se considera la existencia de un contacto más (véanse las gráficas 7 y 8) y el total de tiempo donde existe más tiempo en la medición manual. La explicación que se plantea para estas diferencias radica en la precisión del análisis en el *software*. Por ejemplo, se considera que existe una mayor cantidad de tiempo en la medición manual debido al tiempo de reacción requerido por parte de un humano para pulsar el inicio o pausa del cronómetro. Se ha observado una

adición de casi un segundo por cada “contacto” entre ratones debido al tiempo de reacción requerido. Dado que el *software* no presenta esta limitación, la medición del tiempo resulta mucho más precisa.

Por lo tanto, cabe destacar entre las limitaciones los problemas hallados en “Id Tracker” tanto a nivel de análisis del vídeo como de interpretación de los resultados. Se debe reconocer la falta de validez de estos tanto por la falta de datos como por la ecuación creada. Los creadores del *software* no llegaron a contestar a ninguno de los e-mails enviados por lo que no se pudo resolver tampoco de este modo ninguno de los problemas.

Otra de las limitaciones sería, a pesar de que no ha afectado a los resultados, el hecho de que, por un lado “Id Tracker” no pueda reconocer ratones blancos, y “simBA” tenga problemas para diferenciar entre los ratones negros.

6. Conclusiones.

Tras haber revisado todos los *softwares* y comparar la utilidad de cada uno de ellos parece reafirmarse el problema hallado al inicio: las técnicas automatizadas no están lo suficiente desarrolladas para sustituir de forma óptima la medición manual de la conducta social en ratones.

Las técnicas automatizadas permiten asegurar una mayor fiabilidad en el análisis, una mayor replicabilidad de este, así como una disminución del tiempo de análisis. El objetivo de estas es el sustituir la medición manual y sus múltiples desventajas (Abbas y Masip-Rodo, 2019; Benice y Raber, 2007; Macri, et al., 2015; Marcuccio, Savonenko y Etienne-Cummings, 2018; Unger et al., 2017).

“DeepLabCut” y “simBA” muestran un gran potencial. La precisión del análisis supera la capacidad humana, además, permiten su adaptación para el aprendizaje de diferentes partes del cuerpo, así como diferentes conductas. Sin embargo, el tiempo requerido para el análisis es por el momento un factor importante que rebaja las ventajas de este análisis. Además, “simBA” por el momento no distingue entre ratones del mismo color sin marcas identificatorias claras. Si bien esto no es un problema para los objetivos del trabajo, sí es un problema para otros análisis propios de la conducta social en ratones. Asimismo, la necesidad de tener conocimientos de programación hace su uso poco accesible para una

cantidad considerable de personas en la comunidad científica, limitando de nuevo sus ventajas.

Aún así, cabe destacar que es una rama en desarrollo que, en un futuro, si se solventan las desventajas podría ser la más adecuada para el análisis de la conducta social en ratones.

Por otro lado, “Id Tracker” ha quedado totalmente descartado por sus múltiples limitaciones, así como la falta de fiabilidad de los resultados.

Finalmente, “SMART” aunque es la técnica que parece mostrarse más accesible actualmente, no permite obtener unos resultados tan precisos como sería deseable. Por ello, a pesar de su fácil uso, rapidez de análisis y variabilidad de opciones para el análisis, sigue probando la limitación existente en una rama tan importante como es el estudio de la conducta social. Si bien, es una buena opción que puede utilizarse en lugar de la medición manual, para una mayor fiabilidad de los resultados sería esencial mejorar su precisión.

En conclusión, el estudio de la conducta social en ratones es fundamental para el desarrollo de la ciencia en diferentes ramas como son la neurociencia, psicología, farmacología o biología (Macri, et al., 2015). Sin embargo, a pesar de su importancia, las técnicas automatizadas desarrolladas todavía presentan múltiples limitaciones que, si bien permiten su uso, no alcanzan el nivel óptimo que permitiría un análisis rápido y preciso.

Por ello, como propuesta de futuro, se espera que técnicas como “DeepLabCut” y “simBA” mejoren el tiempo requerido para el análisis, permitan el uso de ratones del mismo color y desarrollen una interfaz estable en la que no sean necesarios conocimientos de programación. Una vez solventados estos problemas, el análisis de la conducta social podría realizarse de forma rápida y eficaz. Además, sería posible asegurar la fiabilidad de los resultados, así como su replicabilidad. De este modo, la investigación en este ámbito podría ver un desarrollo global exponencial. Por ejemplo, mediante la aplicación de los *softwares* a modelos animales de conductas sociales alteradas como el autismo o la esquizofrenia, entre otros.

7. Referencias.

- Abbas, W., & Masip-Rodo, D. (2019). Computer Methods for Automatic Locomotion and Gesture Tracking in Mice and Small Animals for Neuroscience Applications: a survey. *Sensors (Basel)*, 19(15). doi:10.3390/s19153274
- Animal research info: the global resource for animal use in science*. (2014, septiembre 16). Retrieved from Avances médicos: los ratones y la genética: <https://www.animalresearch.info/es/avances-medicos/linea-de-tiempo/los-ratones-y-la-genetica/>
- Animal research info: the global source for animal use in science*. (2014, septiembre 15). Retrieved from El diseño de la investigación: ¿por qué se usan animales?: <https://www.animalresearch.info/es/el-diseno-de-la-investigacion/por-que-se-usan-animales/>
- Barbera, G., Liang, B., Moffitt, C., Li, Y., & Lin, D.-T. (2020). An open-source capacitive touch sensing device for three chamber social behavior test. *Elsevier*.
- Benice, T., & Raber, J. (2007). Object recognition analysis in mice using nose-point digital video tracking. *Journal of Neuroscience Methods*, 168, 422–430. doi:10.1016/j.jneumeth.2007.11.002
- Dandan, S., & Lin, X. (2016). A hybrid video and RFID Tracking System for Multiple Mice in Lab Environment. *3rd International Conference on Information Science and Control Engineering (ICISCE)*, (pp. pp. 1198-1202). Beijing. doi:10.1109/ICISCE.2016.257
- Deep Learning: tres cosas que es necesario saber*. (n.d.). Retrieved from MathWorks: <https://es.mathworks.com/discovery/deep-learning.html>
- DeepLabCut. (2021, Jan 31). *DeepLabCut-Utils*. Retrieved from GitHub: <https://github.com/DeepLabCut/DLCutils>
- Gilat, A. (2006). *Matlab: una introducción con ejemplos prácticos*. Barcelona: Reverté.
- Giovannucci, A., Pnevmatikakis, E., Deverett, B., Pereira, T., Fondriest, J., Brady, M., . . . Parés-D.Masip, P. (2017). Automated gesture tracking in head-fixed mice. *Journal of Neuroscience Methods*. doi:10.1016/j.jneumeth.2017.07.014
- Habedank, A., Urmersbach, B., Kahnau, P., & Lewejohann, L. (2021). O mouse, where art thou? The Mouse Position Surveillance System. *Behavior Research Methods*, 54, 676–689. doi:10.3758/s13428-021-01593-7

- Hao, X., Zhang, G., & Ma, S. (2016). Technical survey: deep learning. *International Journal of Semantic Computing*, 417-439. doi:10.1142/S1793351X16500045
- Hong, W., Kennedy, A., Burgos-Artizzu, X. P., Zelikowsky, M., Navonne, S. G., Perona, P., & Anderson, D. J. (2015). Automated measurement of mouse social behaviors using depth sensing, video tracking, and machine learning. *PNAS*. doi:10.1073/pnas.1515982112
- Klibaite, U., Berman, G. J., Cande, J., Stern, D. L., & Shaevitz, J. W. (2018). An unsupervised method for quantifying the behavior of paired animals. *Physical Biology*, 14(1). doi:10.1088/1478-3975/aa5c50
- Lorbach, M., Poppe, R., Van Dam, E., Noldus, L., & Veltkamp, R. C. (2015). Automated recognition of social behavior in rats: the role of feature quality. *Lecture notes in computer science*. 9280, pp. 565-574. Springer, Cham. doi:https://doi.org/10.1007/978-3-319-23234-8_52
- Macri, S., Mainetti, L., Patrono, L., Pieretti, S., Secco, A., & Sergi, I. (2015). A Tracking System for Laboratory Mice to Support Medical Researchers in Behavioral Analysis. *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, (págs. pp. 4946-4949). Milan.
- Marcuccio, F., Savonenko, A. V., & Etienne-Cummings, R. (2018). Automated Tracking System for Identification of Tagged Mice for Automatic Social Behavior Analysis. *IEEE Biomedical Circuits and Systems Conference, BioCAS*. Cleveland: Institute of Electrical and Electronics Engineers Inc. doi:10.1109/BIOCAS.2018.8584712
- Mathis, A., Mamidanna, P., Cury, K., Abe, T., Murthy, V., Weygandt-Mathis, M., & Bethge, M. (2018). DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature neuroscience*, 21, 1281-1289. doi:10.1038/s41593-018-0209-y
- Nilsson, S. R., Goodwin, N. L., Choong, J.-J., Hwang, S., Wright, H. R., Norville, Z. C., . . . Golden, S. A. (2020). Simple Behavioral Analysis (SimBA) – an open source toolkit for computer classification of complex social behaviors in experimental animals. *bioRxiv*. doi:10.1101/2020.04.19.049452
- Page, D. T., Kuti, O. J., & Sur, M. (2009). Computerized assessment of social approach behavior in mouse. *Frontiers in behavioral neuroscience*, 3(48). doi:10.3389/neuro.08.048.2009

- Peleh, T., Bai, X., Kas, M. J., & Hengerer, B. (2019). RFID-supported video tracking for automated analysis of social behaviour in groups of mice. *Journal of Neuroscience Methods*, 325. doi:10.1016/j.jneumeth.2019.108323
- Pereira, T. D., Tabris, N., Matsliah, A., Turner, D. M., Li, J., Ravindranath, S., . . . Murthy, M. (2022). SLEAP: A deep learning system for multi-animal pose tracking. *Nature*, 19, 486-495. doi:10.1038/s41592-022-01426-1
- Pérez-Escudero, A., Vicente-Page, J., Hinz, R., Arganda, S., & de Polavieja, G. (2014). idTracker: tracking individuals in a group by automatic identification of unmarked animals. *Nature Methods*, 11(7), 743-786. doi:10.1038/nmeth.2994
- Robie, A., Seagraves, K., Roian Egnor, S., & Branson, K. (2017). Machine Vision methods for analyzing social interactions. *Journal of experimental biology*, 220, 25-34. doi:10.1242/jeb.142281
- Schaefer, A. T., & Claridge-Chang, A. (2012). The surveillance state of behavioral automation. *Current Opinion in Neurobiology*, 22, 170–176. doi:10.1016/j.conb.2011.11.004
- Segalin, C., Williams, J., Karigo, T., Hui, M., Zelikowsky, M., Sun, J. J., . . . Kennedy, A. (2021). The mouse action recognition system (MARS) software pipeline for automated analysis of social behaviors in mice. *eLife*, 1-35. doi:10.7554/eLife.63720
- Spink, A., Tegelenbosch, R., Buma, M., & Noldus, L. (2001). The EthoVision video tracking system—A tool for behavioral phenotyping of transgenic mice. *Physiology & Behavior*, 73, 731– 744.
- Tabris, N., Pereira, T. D., Matsliah, A., Turner, D. M., Li, J., Ravindranath, S., . . . Murthy, M. (2022). SLEAP: A deep learning system for multi-animal pose tracking. *Nature Methods*, 19, 486-495. doi:10.1038/s41592-022-01426-1
- Thanos, P. K., Restif, C., O'Rourke, J. R., Lam, C. Y., & Metaxas, D. (2015). Mouse Social Interaction Test (MoST): a quantitative computer automated analysis of behavior. *Translational neurosciences*. doi:10.1007/s00702-015-1487-0
- Unger, J., Mansour, M., Kopaczka, M., Gronloh, N., Spehr, M., & Merhof, D. (2017). An unsupervised learning approach for tracking mice in an enclosed area. *BMC Bioinformatics*, 18(272). doi:10.1186/s12859-017-1681-1
- Universitat de València. (2023). *Red Española de Supercomputación*. Retrieved from Tirant: <https://www.res.es/es/nodos-de-la-res/tirant>

8. ANEXO.

Anexo I. Vídeos.

- **Vídeo 1** (primer vídeo de ratones proporcionado): [Vídeo 1 - YouTube](#)
- **Vídeo 2** (segundo vídeo de ratones proporcionado): [Vídeo 2 - YouTube](#)
- **Vídeo 3** (tercer vídeo proporcionado con puntos clave creados con “DeepLabCut”): [Vídeo 3: puntos aprendidos por DLC - YouTube](#)

Se eligieron como puntos la nariz, orejas y base de la cola debido a que era la configuración disponible para el análisis con “simBA” más adelante.

- **Vídeo 4** (gráfica en movimiento del recorrido de los ratones): [Gráfica en movimiento. Recorrido de los ratones. - YouTube](#)

La gráfica fue generada a través de “MatLab” con las coordenadas proporcionadas por “Id Tracker”.

- **Vídeo 5** (comparación aproximada del movimiento de los ratones con el recorrido de la gráfica): [Vídeo comparación gráfica-ratones APROXIMADO - YouTube](#)

Cabe destacar que la gráfica se encuentra volteada en el vídeo debido a que no era posible alterar su orientación en MatLab. En su creación se formaba por defecto al revés, sin embargo, se puede apreciar que, una vez dada la vuelta, coincide de forma aproximada con el recorrido de los ratones.

- **Vídeo 6** (análisis de conducta “simBA”): [Vídeo 6: Análisis conducta simBA - YouTube](#)
- **Vídeo 7** (análisis de conducta “simBA”, ratones negros): [Vídeo 7: análisis conductas simBA en ratones negros - YouTube](#)
- **Vídeo 8** (gráfica creada con los comportamientos): [Vídeo 8: Gráfica creada con los comportamientos. - YouTube](#)

Entre los resultados proporcionados por “simBA” se encuentra la gráfica mostrada en el vídeo. En ella, se observa a tiempo real cuándo y durante cuánto tiempo ocurre cada conducta. Por tanto, se puede apreciar en el eje horizontal el tiempo. En el vertical se encuentran las conductas 1 (seguimiento) y 2 (contacto). El seguimiento, se muestra en rojo y el contacto en negro.

De forma más práctica, se puede tomar la última imagen creada por la gráfica donde se expone el resultado final con el número total de interacciones expresado a lo largo del vídeo.

Anexo II. Protocolo de descarga e instalación de “Id Tracker”.

1. Acceda a la página web de “Id Tracker”: [idTracker - Automatic tracking of multiple animals](#). Al acceder a esta, se encuentra en la página principal donde se presenta el *software*. En la zona inferior, existen una serie de enlaces donde se puede pinchar para obtener más información.
2. En este caso, se desea acceder a la zona de descarga. Para ello, se debe clicar en el apartado superior derecho *download*.
3. Tras acceder a este, la página presenta los pasos a seguir para su instalación. Primero se debe descargar e instalar *MatLab Complier Runtime 8.3*. Para ello, simplemente se debe pinchar sobre este.
4. Además, es necesario descargar los archivos ejecutables de “Id Tracker” clicando en *Download the executables of idTracker*.
5. Se puede descargar una guía de uso gratuita, aunque no actualizada, si se aprieta en *User Guide*.
6. Vaya a la carpeta de descargas del ordenador. En ella podrá ver un archivo llamado *MatLab Complier Runtime* o similar que terminará en “.exe”. Este será el que se utilizará para la instalación. También se puede ver un archivo comprimido con los ejecutables.
7. Primero, descomprima ese archivo.
8. Ahora, acuda al *MatLab Complier Runtime* y clique en él para iniciar la ejecución.
9. Aparecerán múltiples pantallas solicitando permisos para ejecutar el programa, se deben aceptar todas ellas.
10. Después se abrirá una pestaña en la que se solicitará especificar dónde se desea instalar. Puede elegir dónde dando a *Browse* y seleccionando el lugar deseado. Seguidamente clique *Next*.
11. La siguiente pantalla muestra información acerca de la carpeta de instalación, así como el tamaño de la instalación (1.334 MB). Las opciones son *Back* (volver atrás), *Install* (para instalar el programa) o *Cancel* (cancelar). En este caso, se desea instalar el programa así que se debe dar a *Install*.
12. Se instalará el programa y aparecerá una última pantalla donde se debe pinchar en *Finish* para finalizar.

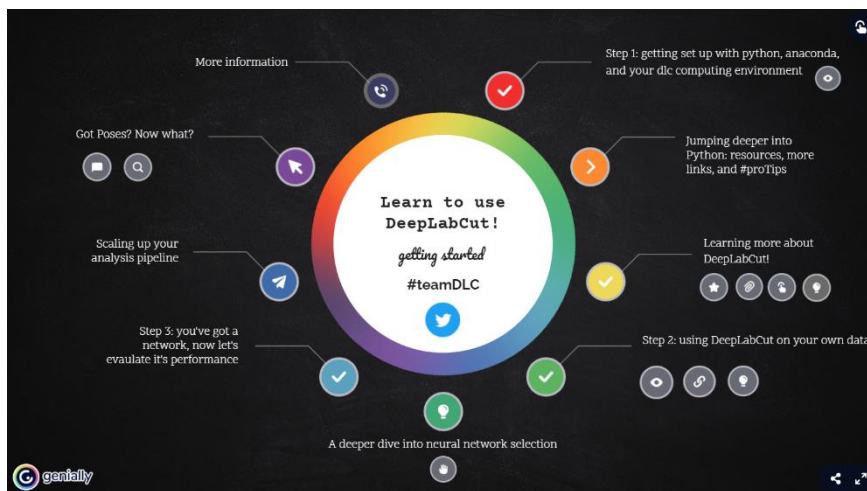
13. ¿Cómo abrir el programa instalado? Vuelva a la carpeta donde se encuentran los ejecutables que se han descomprimido, deberían hallarse en descargas, pero pueden moverse donde uno desee.
14. Si clica en “Id Tracker” podrá analizar el vídeo que desee.
15. Si clica en “IdPlayer”, podrá visualizar un vídeo ya analizado.

Anexo III. Protocolo de “DeepLabCut” para ordenadores con CPU.

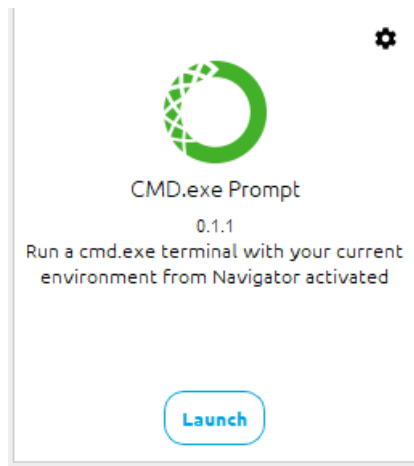
Advertencia: en un ordenador con CPU tan sólo se puede crear el conjunto de datos de entrenamiento, no puede llegar a analizar ni entrenar.

Advertencia: en los comandos introducidos, al copiarlos en el terminal, **no** se deben incluir las comillas.

1. Entrar en la página web de “DeepLabCut” (DLC): [DeepLabCut — The Mathis Lab of Adaptive Motor Control \(mackenziemathislab.org\)](https://mackenziemathislab.org).
2. Deslice hacia abajo hasta encontrar la siguiente imagen:



3. Clique en el paso 1 (*Step 1: Getting set up with python, anaconda and your dlc computing enviroment*).
4. Se redirigirá a la siguiente página: [How To Install DeepLabCut — DeepLabCut](https://mackenziemathislab.org/how-to-install-deeplabcut).
5. Deslice hacia abajo hasta encontrar *Step 1: You need to have python installed*.
6. Es necesario instalar anaconda, como bien se indica en las instrucciones de la web. Acceda al enlace *anaconda* para descargar anaconda en su dispositivo.
7. Se encontrará aquí: [Anaconda | Anaconda Distribution](https://anaconda.org/).
8. Descargue Anaconda clicando en *Download*.
9. Una vez descargado, acuda a su carpeta de descargas y ejecute el programa. La propia ventana que aparecerá le guiará adecuadamente a través de los pasos de instalación.
10. Ahora vuelva a la página principal de DLC. Si desliza hasta el final verá que aparece la opción *Download Conda File*. Es para descargar el archivo de conda con la información necesaria para instalar DLC.
11. Clique para descargar dicho archivo.



12. Ahora abra *Anaconda Navigator* en su ordenador, si no lo encuentra, haga una búsqueda y escriba “Anaconda Navigator”. Debería aparecer.

***IMPORTANTE:** Para abrirlo, haga clic derecho, seleccione abrir como administrador.

13. Una vez abierto, seleccione *launch* para “CMD.exe Prompt”.

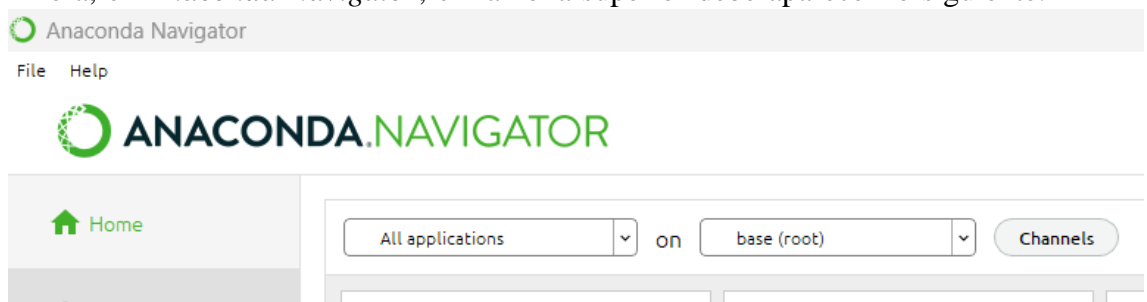
14. Ahora debe introducir el comando “cd” + la dirección de la ruta donde se encuentra su archivo descargado (conda file). Por ejemplo, si no la ha movido, debería encontrarse en su carpeta de descargas. De este modo el comando sería: “cd C:\Users\Usuario\Downloads”.

15. El siguiente paso es descargar los paquetes necesarios. Puede realizar esto fácilmente introduciendo: “conda env create -f DEEPLABCUT.yaml”.

16. Cuando termine la instalación, active el ambiente escribiendo: “conda activate DEEPLABCUT”.

17. Cierre la pestaña.

18. Ahora, en *Anaconda Navigator*, en la zona superior debe aparecer lo siguiente:



19. Haga clic en “base (root)”, verá que aparece un desplegable. En él verá “DEEPLABCUT” como opción. Pinche en ella.

20. Cuando cargue deberá instalar “CMD.exe Prompt”.

21. Después clique *launch*.

22. **IMPORTANTE:** para el funcionamiento adecuado de DLC debe introducir el siguiente comando: “pip install tensorflow”.
23. En algún momento de la instalación es posible que se le pida permiso para continuar clicando las teclas “y” (sí) o “n” (no). Introduzca “y” y presione *enter*.
24. Tras finalizar la descarga para iniciar una sesión con una guía visual debe escribir: “ipython”.
25. Después inserte: “import deeplabcut”.
26. Finalmente: “deeplabcut.launch_dlc()”.
27. Si ha logrado entrar adecuadamente aparecerá la siguiente pantalla:

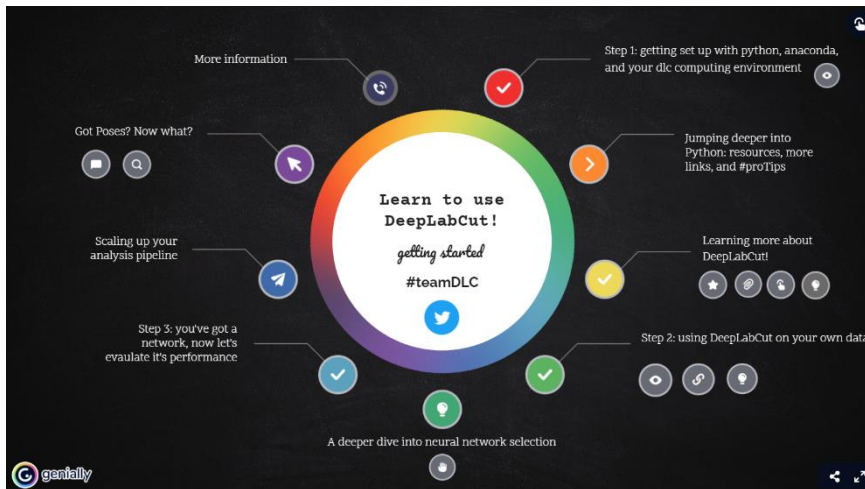


Anexo IV. Protocolo para ordenadores con GPU.

IMPORTANTE: debe tener Windows 10.

Pasos básicos.

1. Entre en la página web de “DeepLabCut” (DLC): [DeepLabCut — The Mathis Lab of Adaptive Motor Control \(mackenziemathislab.org\)](https://www.mathislab.org/deeplabcut/).
2. Deslice hacia abajo hasta encontrar la siguiente imagen:



3. Clique en el paso 1 (*Step 1: Getting set up with python, anaconda and your dlc computing enviroment*).
4. Se redirigirá a la siguiente página: [How To Install DeepLabCut — DeepLabCut](https://www.mathislab.org/deeplabcut/how-to-install-deeplabcut/).
5. Deslice hacia abajo hasta encontrar *Step 1: You need to have python installed*.
6. Es necesario instalar anaconda, como bien se indica en las instrucciones de la web. Acceda al enlace *anaconda* para descargar anaconda en su dispositivo.
7. Se encontrará aquí: [Anaconda | Anaconda Distribution](https://www.anaconda.com/distribution/).
8. Descargue Anaconda clicando en *Download*.
9. Una vez descargado, acuda a su carpeta de descargas y ejecute el programa. La propia ventana que aparecerá le guiará adecuadamente a través de los pasos de instalación.

ADVERTENCIA: a partir de este punto la instalación de CPU y GPU son diferentes.

Pasos especiales para GPU.

Si su ordenador tiene una GPU Nvidia siga los siguientes pasos:

1. Entre en la siguiente página de *tensorflow*: [Compatibilidad con GPU | TensorFlow](https://www.tensorflow.org/install/gpu) . En ella encontrará la información y los pasos necesarios.

2. Compruebe que su GPU Nvidia es compatible con CUDA: [CUDA GPUs - Compute Capability | NVIDIA Developer](#).
3. Si es compatible aparecerá entre las GPU de la página.

Instalación del driver.

1. Ahora debe instalar un controlador (*driver*) de GPU Nvidia. Para ello acceda a la siguiente página: [Official Drivers | NVIDIA](#)
2. Deberá seleccionar la información compatible con su GPU como sería el tipo de producto, serie, producto y sistema operativo. Esto es lo más importante para que sea adecuado y compatible. Una vez seleccionado, descargue el *Driver*.
3. Una vez descargado, acceda a su carpeta de descargas y ejecute el archivo descargado. Siga los pasos indicados por la ventana que aparecerá.

Instalación de “Visual Studio”.

1. Para el correcto funcionamiento, es necesario instalar la versión 2019 de “Visual Studio”. Esta es la versión compatible con “CUDA”.
2. Para ello acceda a la siguiente página: [Descargas anteriores de Visual Studio: 2019, 2017, 2015 y versiones anteriores \(microsoft.com\)](#)
3. Seleccione la versión de 2019 y clique “descargar”.
4. Se le pedirá que inicie sesión en Microsoft (es gratuito).
5. Tras iniciar sesión seleccione la versión de “Visual Studio” que desee. En este caso, *Visual Studio Community 2019*. Presione *download*.
6. Una vez descargado, ejecute el programa. La pantalla que aparecerá le guiará por los pasos necesarios para su correcta instalación.
7. Finalmente, aparecerá una pantalla que le permitirá elegir las extensiones que desea instalar. Seleccione todas aquellas que incluyan “Python”, “C++” y “MSVC”. Pueden encontrarse entre la primera y la segunda pestaña.
8. Una vez seleccionadas todas clique instalar.

Instalación “CUDA”.

1. Ahora debe descargar un kit de herramientas “CUDA”. Para acceder al enlace puede volver a la página de *tensorflow* o clicar aquí: [CUDA Toolkit Archive | NVIDIA Developer](#).
2. **IMPORTANTE:** la versión de *tensorflow* que instale, “Python”, “cuDNN” y “CUDA” deben ser compatibles. Para asegurar que elige las versiones adecuadas acceda a la siguiente página: [Construir desde la fuente | TensorFlow](#) y deslice hasta encontrar la información acerca de GPUs. Debería encontrar algo similar a esto:

GPU

Versión	Versión de Python	Compilador	Construir herramientas	cuDNN	CUDA
tensorflow-2.11.0	3.7-3.10	CGC 9.3.1	Bazel 5.3.0	8.1	11.2
tensorflow-2.10.0	3.7-3.10	CGC 9.3.1	bazel 5.1.1	8.1	11.2
tensorflow-2.9.0	3.7-3.10	CGC 9.3.1	bazel 5.0.0	8.1	11.2

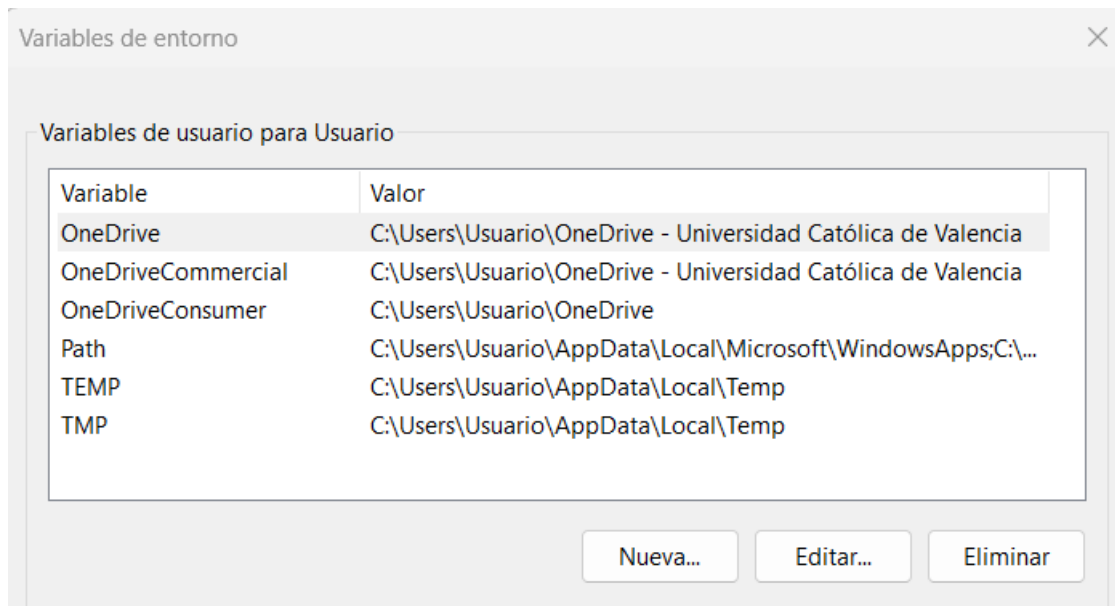
Como puede ver, la versión de *tensorflow* 2.11 requiere “Python” entre 3.7 y 3.10, “cuDNN” 8.1 y “CUDA” 11.2.

4. En este caso, se trabajará en la versión más actualizada hasta la fecha, *tensorflow* 2.11. Por ello, en el enlace de “CUDA Toolkit” debemos elegir “CUDA” 11.2 como se indica en la imagen, así como en el enlace proporcionado.
5. Una vez se encuentre, clique en él y seleccione el sistema operativo que su ordenador posee. Aparecerán dos opciones más: *architecture*, que tiene una sola opción, seleccione esta, y *versión*. En el caso de “CUDA” 11.2 seleccione la versión 10.
6. Una vez seleccionada la versión se le dará la opción de elegir el tipo de instalación, local o en línea. Cualquiera de estas dos opciones es adecuada. La versión local es más pesada (2.9 GB).
7. Presione *download*.
8. Una vez descargada, acuda a su carpeta de descargas y ejecute su instalación. En su pantalla aparecerán las instrucciones necesarias.

IMPORTANTE: si aparece una advertencia acerca de la incompatibilidad de “Visual Studio” con “CUDA”, ha instalado “Visual Studio” mal, cancele la instalación de “CUDA” y revise “Visual Studio”.

Instalación de “cuDNN”.

1. Ahora puede volver la página inicial de *tensorflow* para acceder al enlace que permite la descarga de “cuDNN”.
2. Entre en el siguiente enlace: [CUDA Deep Neural Network \(cuDNN\) | NVIDIA Developer](#)
3. Seleccione *download cuDNN*.
4. Se le pedirá que cree una cuenta.
5. Una vez creada, acepte los términos y condiciones de “cuDNN”.
6. Seleccione la versión de “cuDNN” compatible con el “CUDA” instalado.
7. Si no aparece clique *archived cuDNN releases*.
8. Se cargará una página con todas las versiones.
9. Seleccione aquella que indique su compatibilidad con el “CUDA” instalado (en este caso deseamos “cuDNN” 8.1, compatible con “CUDA” 11.2).
10. Aparecerá un desplegable. Dependiendo de su ordenador y sistema operativo deberá descargar una versión u otra.
11. Descargue el archivo.
12. En su carpeta de descargas, descomprima el archivo.
13. Entre en el archivo. Podrá ver que está lleno de diferentes carpetas. Copie todas.
14. Ahora busque en su terminal (todo el ordenador), la carpeta “CUDA” que debe haberse creado.
15. Entre en ella y pegue todas las carpetas.
16. Seleccione con clic derecho la carpeta “bin”. Copie la ruta de acceso.
17. Busque en su ordenador: “editar las variables de entorno”. Entre.
18. Aparecerá una pequeña ventana. Acceda a ella y presione “variables de entorno...”. Se abrirá una nueva ventana.



19. En esta nueva ventana clique en “Path”.
20. Abrirá otra pestaña. Clique en la derecha “nuevo” y pegue la ruta de acceso.
21. Vuelva a la carpeta “CUDA”.
22. Copie la ruta de acceso de “libvn” (el nombre puede variar ligeramente).
23. En la pestaña de “path” vuelva a clicar “nuevo” y pegue la ruta de acceso.
24. Cierre todo y reinicie el ordenador.

Instalar *tensorflow*.

1. Tras reiniciar el ordenador. Abra *Anaconda Navigator*.
2. Abra “CMD.exe Prompt”.
3. Escriba “conda create --name tf_2.11 python==3.10”.

*El tipo de *tensorflow* y de “Python” pueden cambiarse al compatible con lo instalado.

Consulte el enlace anteriormente proporcionado.

4. Se le indicará que dé permiso para continuar la instalación “y” (sí) o “n” (no).
Escriba “y” y presione enter.
5. Introduzca el comando: “conda activate tf_2.11”.
6. Ahora escriba: “pip install tensorflow”.
7. Se instalará *tensorflow* compatible con su GPU.

***Advertencia:** en ocasiones, las versiones supuestamente compatibles dan error.

En este caso, instale las siguientes versiones:

- cudnn==8.4.1.50
- cudatoolkit==11.8
- tensorflow==2.10.0

Siga los mismos pasos que indicados anteriormente, pero cambie las versiones que descarga.

Instalar DEEPLABCUT.

1. Descargue la “conda file” de la página oficial de DLC. [DeepLabCut — The Mathis Lab of Adaptive Motor Control \(mackenziemathislab.org\)](https://mackenziemathislab.org). Se encuentra al final de la página.
2. Vuelva al terminal (ventana negra en la que ha introducido los comandos). En el mismo ambiente en el que se encuentra escriba la ruta a descargas donde se encuentra su “conda file”, precedido de cd. Por ejemplo: “cd C:\Users\Usuario\Downloads”.
3. Ahora introduzca el siguiente comando: “conda env create -f DEEPLABCUT.yaml”.
4. Siguiendo comando: “conda activate DEEPLABCUT”:
5. Finalmente instale tensorflow en este nuevo ambiente: “pip install tensorflow==2.10.0”.

Para abrir DLC siga los siguientes pasos:

1. Introduzca: “ipython”.
2. Después inserte: “import deeplabcut”.
3. Finalmente: “deeplabcut.launch_dlc()”.

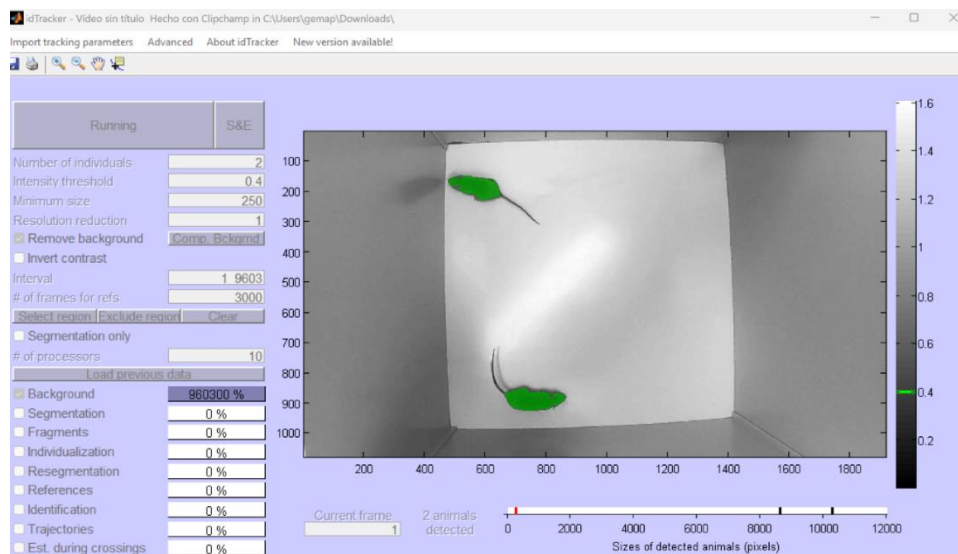
RECUERDE: cuando introduzca comandos presione siempre “enter” al terminar escribirlos para ejecutarlos.

Advertencia: en ocasiones, DLC también presenta errores cuando es abierto del modo anteriormente explicado.

Si encuentra errores, una versión alternativa con la que puede abrir el software es escribiendo “python -m deeplabcut” en el terminal, dentro del *environment* de DLC y presionando *enter*.

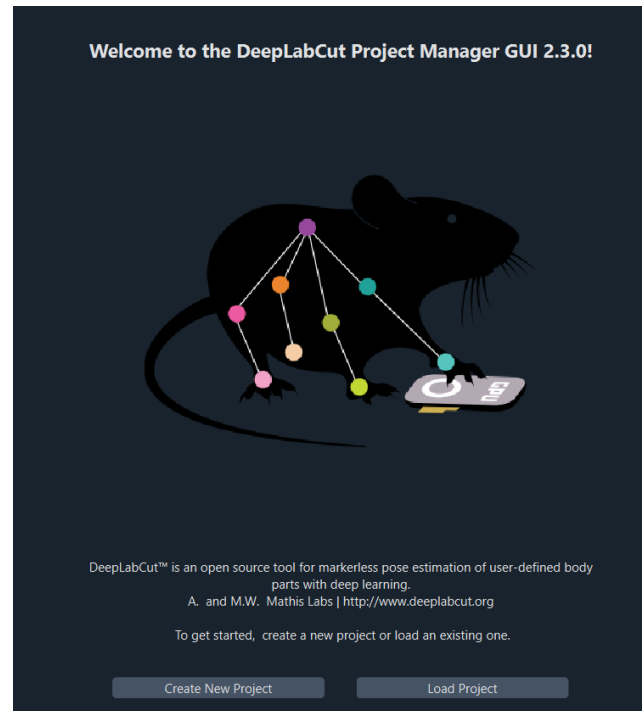
Anexo V. Cómo analizar un vídeo con “Id Tracker”.

1. Primero se debe adecuar el vídeo para las características de “Id Tracker”. Se deben recortar los márgenes, acortar las imágenes que no contengan los dos ratones al mismo tiempo (inicio y final).
2. Abra “Id Tracker”.
3. Seleccione el vídeo que se desea analizar.
4. Una vez subido, se debe ajustar el número de individuos.
5. También es necesario ajustar la intensidad del umbral. El objetivo de esto es que la zona “verde”, aquella que detecta el número de individuos, se encuentre tan sólo en los ratones.
6. Se puede aumentar el tamaño mínimo siempre que sea inferior al tamaño mínimo que llegan a mostrar los individuos.
7. Comparar el fondo con la imagen para distinguir adecuadamente los ratones del fondo: “comp. background”.
8. Tras esto, se puede presionar la zona superior izquierda “Start” para analizar el vídeo.

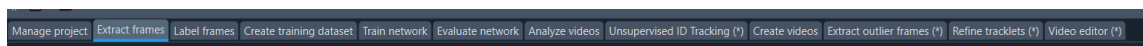
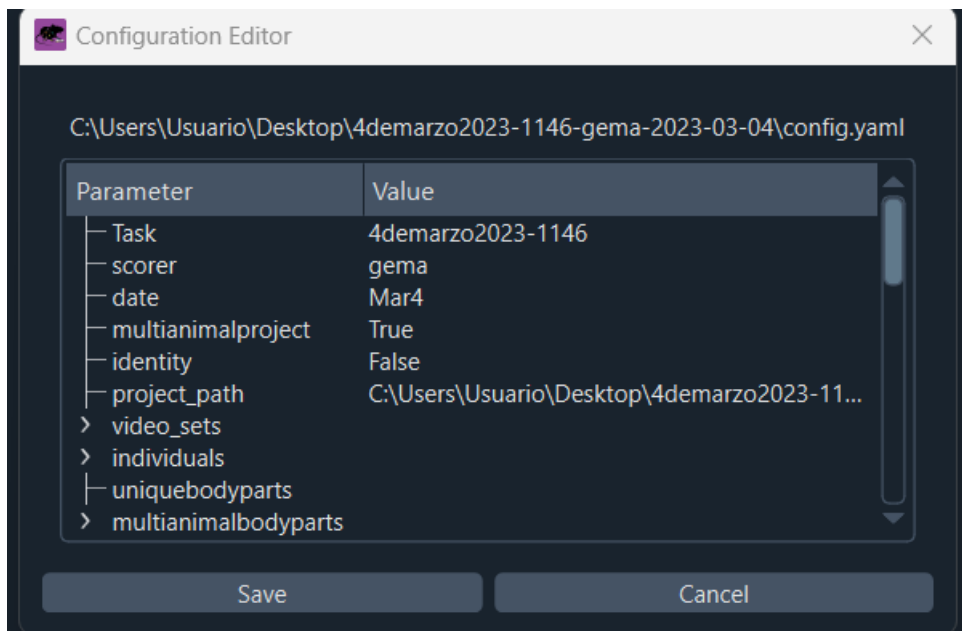


Anexo VI. Cómo analizar un vídeo con “DeepLabCut”.

1. Prepare el vídeo de forma que tan solo se vean los animales, esto facilitará el área de análisis.
2. Haga una copia o dos de seguridad ya que, en ocasiones, el vídeo desaparece al ser subido a DLC.
3. Ahora abra DLC.
4. Seleccione si desea crear un nuevo proyecto o cargar un proyecto ya existente.
5. En este caso, se debe dar a “create new project”.
6. Escriba el título del proyecto, así como la persona que lo realiza.
7. Si va a analizar un vídeo con más de un animal, asegúrese de seleccionar “multianimal project”. Podrá ver la opción en la primera pantalla.

The image shows the "New Project" dialog box in the DeepLabCut software. The dialog has a title bar that says "New Project" and a close button. Inside, there are several input fields: "Project:" (empty), "Experimenter:" (empty), and "Location:" (containing "C:\Users\Usuario\Desktop"). Below these is a checkbox labeled "Is it a multi-animal project?" which is currently unchecked. There is a "Number of cameras:" dropdown menu set to "1". Below that is a checked checkbox labeled "Files". A large empty rectangular area is present below the "Files" checkbox. At the bottom of the dialog, there are two buttons: "Browse videos" and "Clear". Below these buttons is another checkbox labeled "Copy videos to project folder" which is unchecked. In the bottom right corner, there is a "Create" button.

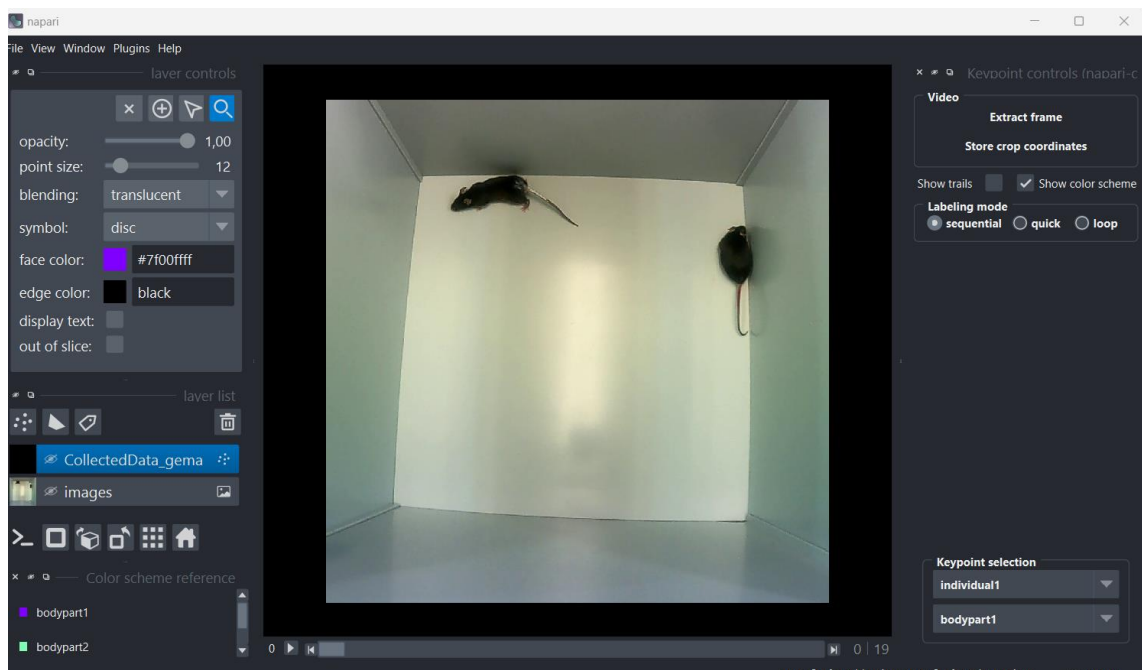
8. Ahora cargue el vídeo seleccionado pinchando “browse videos”. Si no le deja subir el vídeo directamente, métalo en una carpeta que tan solo contenga el vídeo.
9. Seleccione “copy videos to project folder” para asegurarse de que se trabaja con una copia y no el original (por el problema de borrado de vídeos).
10. Puede ver que puede elegir el lugar en el que se guardará el proyecto. Seleccione el lugar deseado.
11. Ahora seleccione “create”.
12. Si ha realizado todo bien, aparecerá una pequeña ventana indicándole que se ha creado un proyecto.
13. **IMPORTANTE:** en la pestaña de “manage project”. Seleccione “edit conf. yaml”. Aquí debe introducir, el número de individuos, partes del cuerpo que se van a evaluar, etc. Si no se realiza correctamente no podrá analizar el vídeo.



***Durante el resto de la explicación se hablará acerca de las siguientes pestañas. En general, se seguirá el orden de izquierda a derecha.**

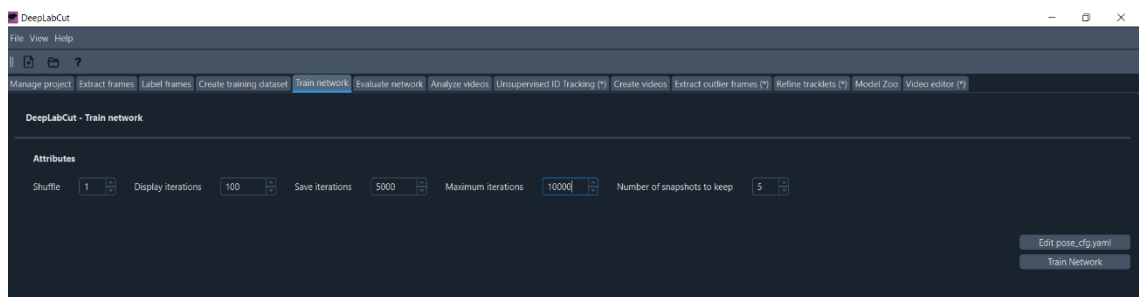
14. Pase a la siguiente pestaña: “Extract frames”. En ella se le permite elegir el tipo de extracción, el algoritmo, los cortes, etc. Si no tiene un conocimiento profundo acerca del funcionamiento de cada uno de ellos, no los toque. El análisis que se proporciona por defecto es adecuado.
15. Presione “extract frames”.

16. Si bien se suele encontrar en la GUI (pestaña interactiva en la que se ha estado trabajando). Es importante que consulte continuamente el terminal ya es donde realmente están ocurriendo los cambios. Entre al terminal.
17. Podrá ver cómo se están extrayendo los “frames” en este momento. Dependiendo de la velocidad de su ordenador esto podrá tomar más o menos tiempo. Es importante que consulte el terminal ya que la GUI se bloquea cuando se está ejecutando una tarea.
18. Cuando terminen de extraerse los “frames” aparecerá una pequeña ventana que le indicará su finalización.
19. Ahora debe etiquetarlo. Pase a la pestaña “label frames”.
20. Pinche en “label frames”. Se abrirá una nueva ventana. Espere si no se abre.
21. Seleccione la carpeta que aparece cuando presiona “label frames”. Ahora se abre una nueva ventana.

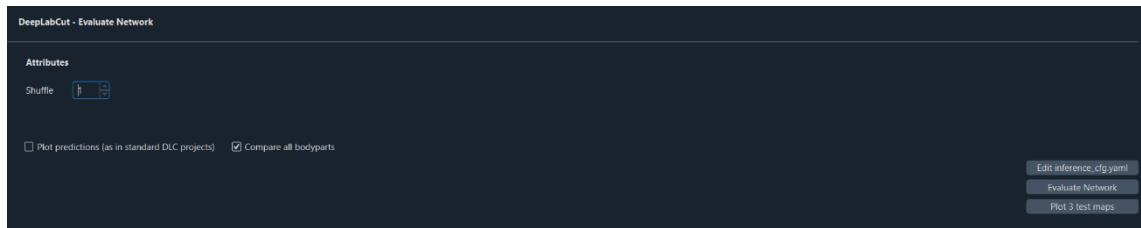


22. En esta parte del proceso es muy recomendable hacer uso de un ratón para poder etiquetar cómodamente las partes.
 1. Con el ratón verá que, si desliza la rueda se acerca o se aleja de la imagen.
 2. Si presiona el círculo con el símbolo de más en el interior, podrá marcar las partes del cuerpo de interés en el ratón.

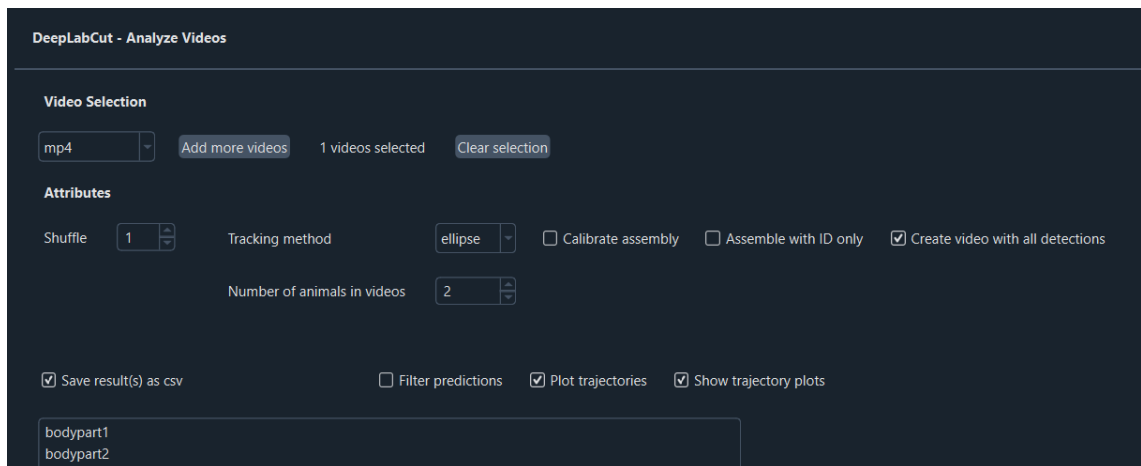
3. Si desea mover alguna, puede presionar en la flecha que se encuentra junto a este.
 4. Si desea volver a ver la imagen en su posición original puede clicar en la imagen de la casa que se encuentra en la zona izquierda.
23. Cuando termine de etiquetar todas las partes del cuerpo de interés, puede guardarlo seleccionando “file”, “save selected layer”.
 24. Tras esto, cierre la ventana.
 25. Ahora puede pasar a la siguiente pestaña, “create training dataset”. Esta es la última parte que puede realizarse en una CPU.
 26. Si no conoce el resto de las opciones que aparecen, puede usar las proporcionadas por defecto. Seleccione “create training data set”. Aparecerá una pequeña ventana avisándole de que ya ha sido creado.
 27. Ahora puede pasar a la siguiente etapa, **SÓLO** si tiene GPU.
 28. Clique en “train network”.
 29. En este paso, va a ser esencial la potencia que tenga su GPU, así como lo bien configurada que se encuentre para su uso. Si su GPU tiene una memoria dedicada inferior a 8 GB es recomendable que mantenga las iteraciones relativamente bajas. Por ejemplo, en una GPU de 6 GB de memoria dedicada pueden tomar 40 minutos en ser analizados con 10.000 itinerancias. El ideal son más de 100.000 itinerancias, lo que equivaldría a más de 6 horas y media.



30. Una vez finalizado el análisis, pase a la siguiente pestaña “evaluate network”. Evalúe la red creada. Este paso suele ser rápido. También puede crear figuras que representen el movimiento, aunque, en este caso, no son útiles ya que no proporcionan la información deseada. Puede también ver cómo serían las figuras en “plot 3 test maps”.



31. Ahora es el momento de analizar y crear un vídeo de ser deseado. En general, se pueden indicar las siguientes órdenes:



- Recuerde seleccionar el vídeo de la carpeta que se ha creado.
 - Recuerde seleccionar la cantidad de animales que se encuentran en el vídeo.
 - En este caso se ha seleccionado que se desea crear un vídeo con las detecciones.
 - También se ha seleccionado “plot trajectories”, “show trajectory plots”. Esto creará gráficos con las trayectorias de los ratones.
 - Finalmente, también se ha pedido que se guarden los resultados como un csv. Un tipo de archivo. Este último no es necesario en sí, depende de las demandas de la persona.
32. Existe una pestaña en la que se puede crear el vídeo. Dependiendo del interés se puede seleccionar las características deseadas como se puede ver en la imagen.

DeepLabCut - Create Videos

Video Selection

mp4

Attributes

Shuffle Overwrite videos Color keypoints by:

Video Parameters

Specify the number of trail points

Plot all bodyparts

Draw skeleton

Use filtered data

Plot trajectories

High quality video (slow)

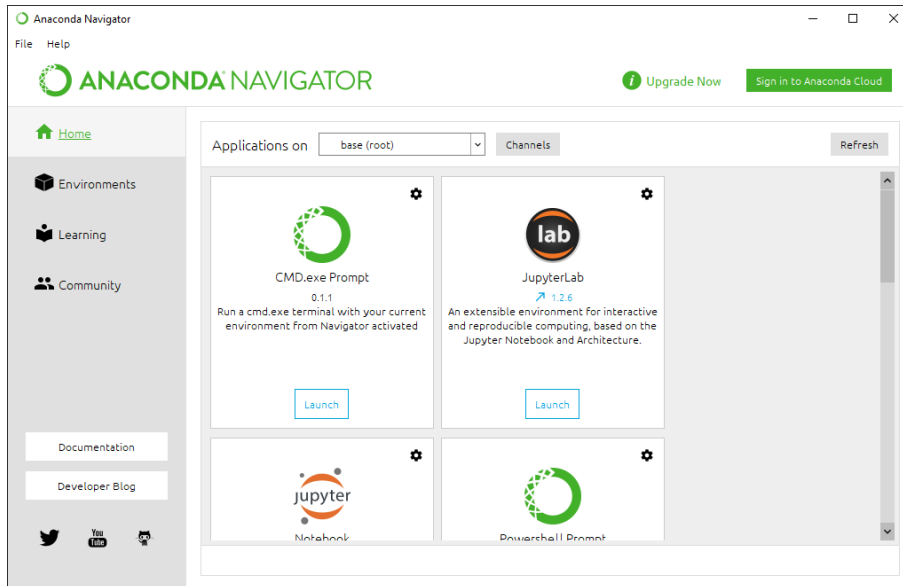
33. Sin embargo, hasta este punto, ya se han obtenido todos los datos necesarios para el análisis.

Anexo VII. Cómo instalar simBA.

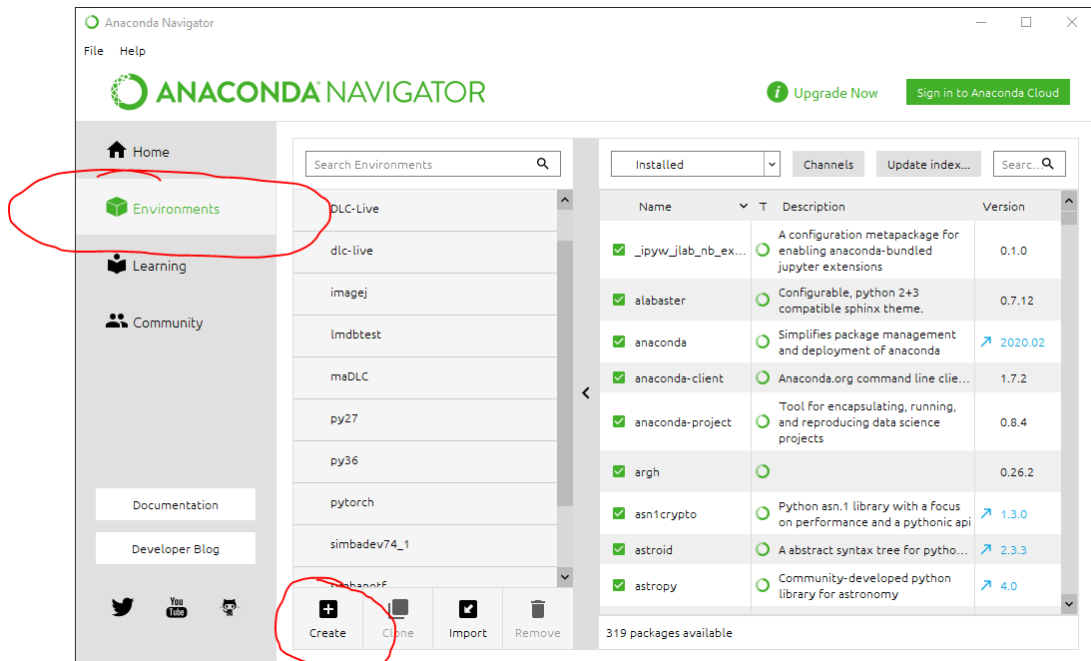
Instalar simBA es más fácil que instalar DLC.

Los pasos son los siguientes:

1. Instale Anaconda. En este caso, ya se tenía dado que era necesario para DLC.
2. Abra *Anaconda Navigator*.

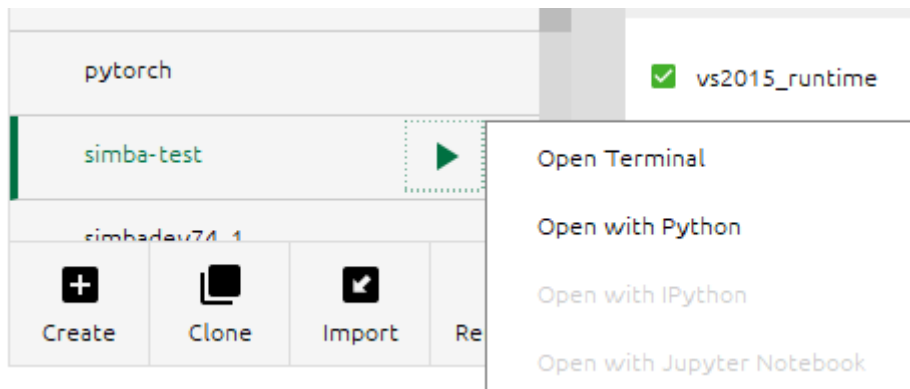


3. Vaya a la barra lateral izquierda y clique en *enviroments*.
4. Cree un nuevo *enviroment*.



5. Ponga nombre al *enviroment* y pulsar “create”.

6. Abra el terminal.



7. Una vez dentro introduzca los siguientes comandos en su respectivo orden:

- conda install python=3.6
- pip install simba-uw-tf-dev
- pip uninstall shapely
- conda install -c conda-forge shapely

8. Si se detiene la descarga en algún momento para que autorice la instalación o desinstalación simplemente clique la letra “y”, pulse *enter* y continuará.

9. Introduzca: “simba”.

10. Pulse *enter* para entrar.

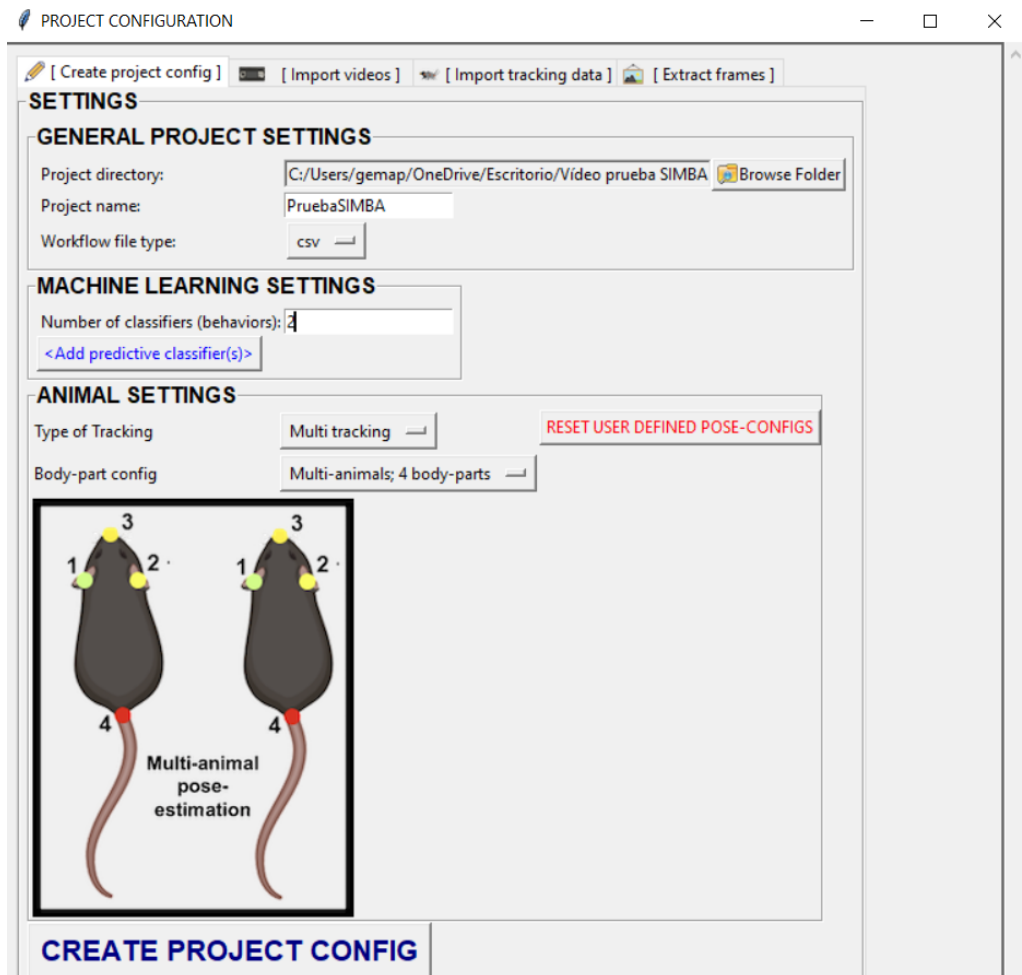
11. Se abrirá la GUI.

Anexo VIII. Cómo utilizar simBA.

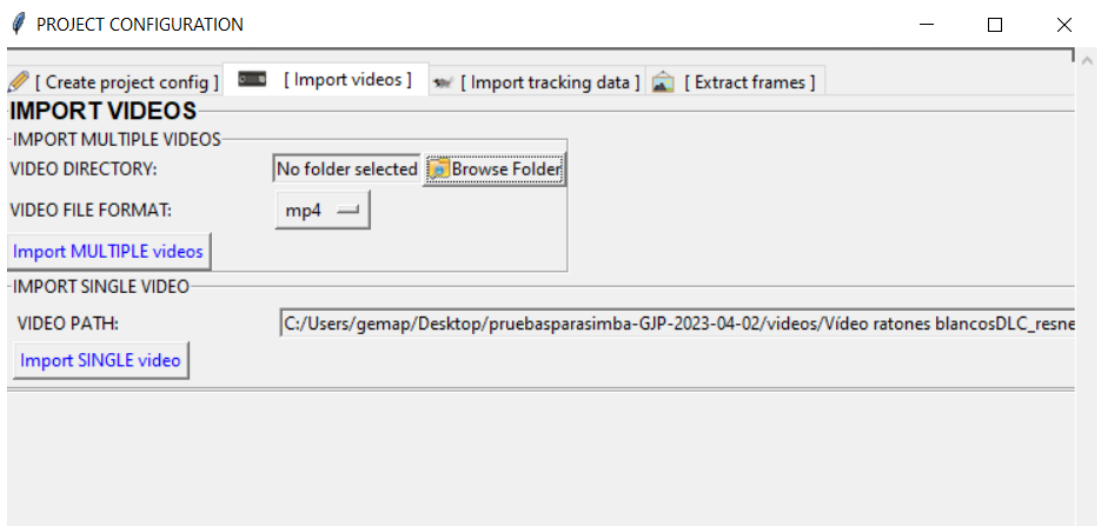
1. Escriba “simba” y pulse *enter* en el *environment* “SIMBA” de Anaconda.
2. Se abrirá la siguiente GUI.



3. Pulse en “file”, seguidamente “create new project”. Al clicar “create new project” se verá la siguiente pantalla.



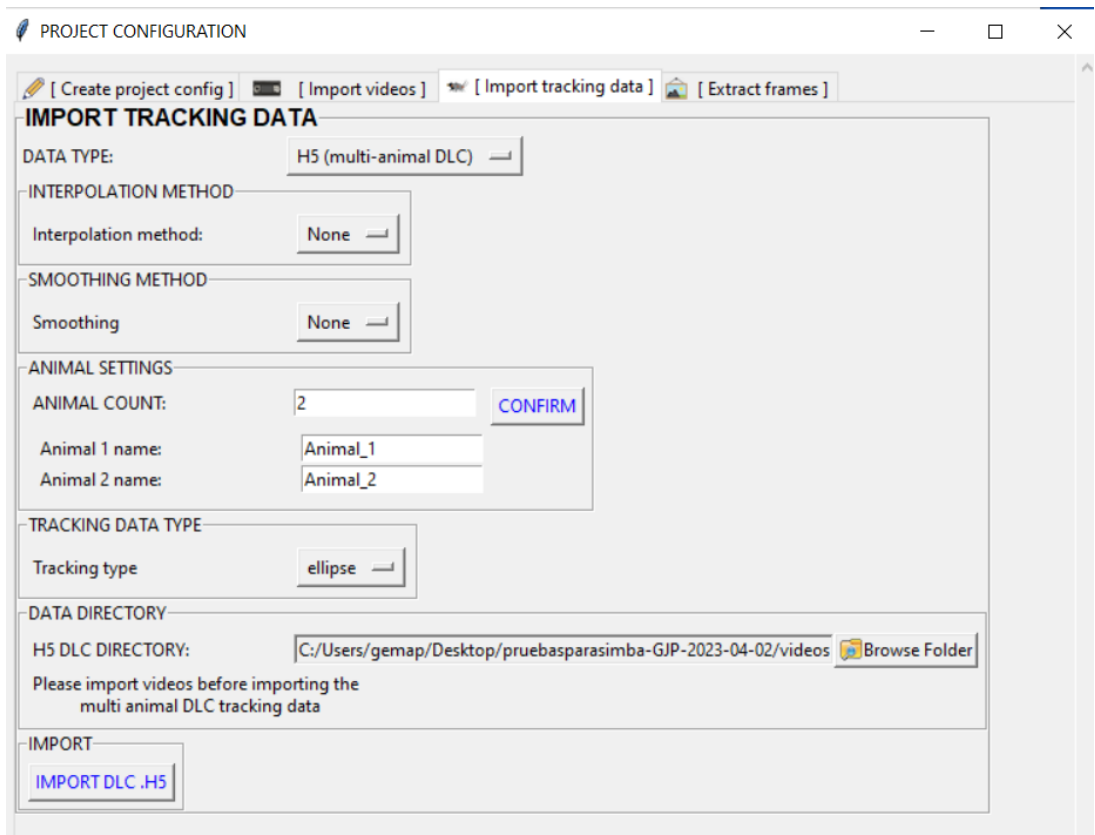
4. En este caso los parámetros seleccionados se pueden observar en la imagen.
5. El siguiente paso es importar el vídeo que desea utilizar para el análisis:



6. Seleccione el lugar en el que se encuentra el vídeo, así como el tipo de archivo.

***Importante:** la primera opción está diseñada para la selección de varios vídeos. La segunda para un solo vídeo.

***Importante:** “simBA” no es compatible con caracteres de otros idiomas (como tildes). Si alguno de los archivos tiene caracteres de otros idiomas dará error. **En cualquiera de los pasos.**

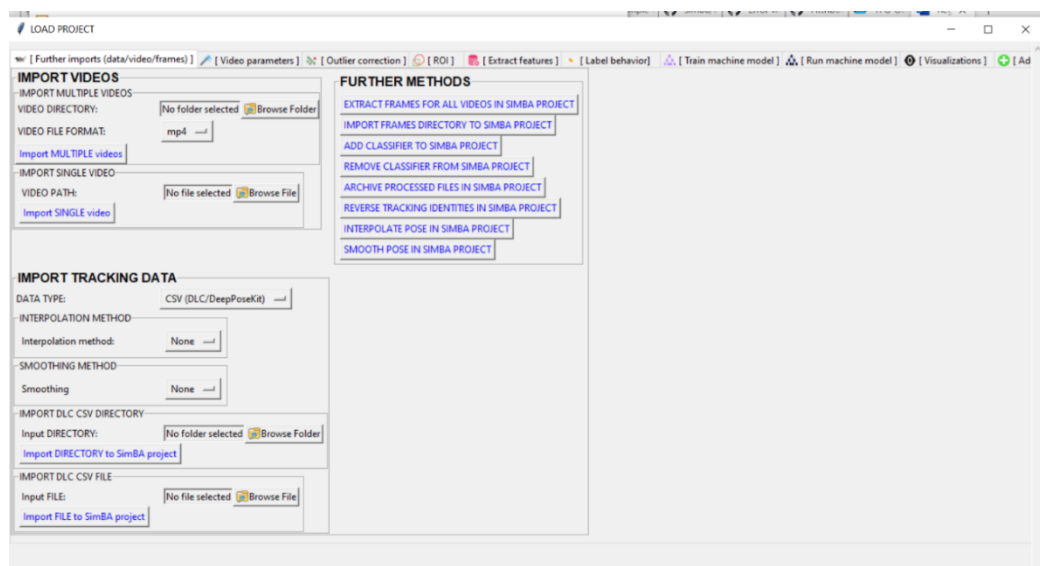
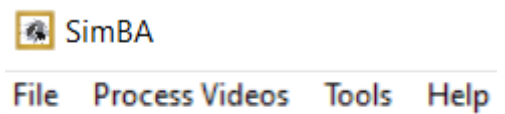


7. Pase a “import tracking data”.

8. En este paso, debe seleccionar el tipo de archivo que contiene la información. Se encuentra en la carpeta creada por “DeepLabCut” donde se encuentran los vídeos y el análisis.
9. Seleccione la cantidad de animales y el tipo de tracking que se ha realizado.
10. Finalmente, en “browse” folder, seleccione la carpeta donde se encuentre un archivo H5. **Recuerde:** los caracteres españoles no los reconoce.
11. Una vez seleccionado todo esto clique en “import DLC .H5”.
12. Ahora aparecerá una pantalla con una imagen del vídeo.
13. Presione “c” si la imagen mostrada es adecuada para identificar las identidades de los ratones. Presione “x” si no es así.
14. Si ha presionado “c” ahora debe hacer doble clic en los ratones por orden de identidad. Primero, dos clics en el primer ratón, luego dos clics en el segundo ratón.
15. Ya está todo. Ahora puede cerrar la pestaña.

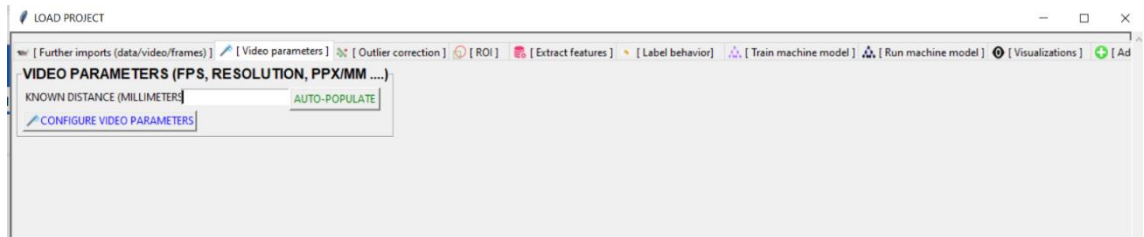
TRABAJAR CON EL PROYECTO CREADO.

1. Ahora en “file” seleccione, “load project”.

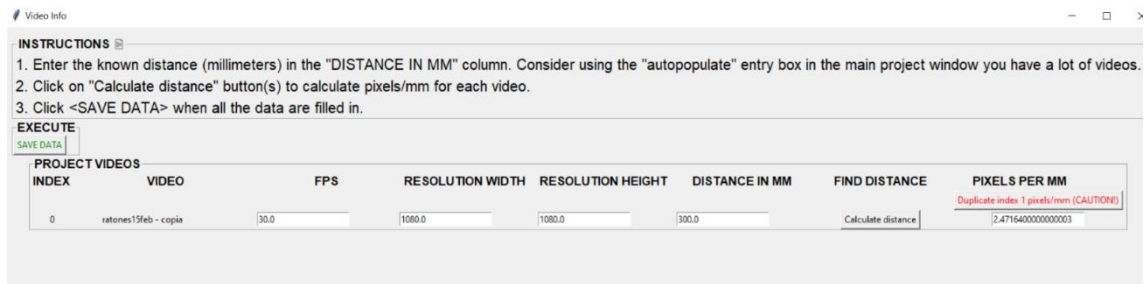


2. Deberá buscar el lugar en el que creó el proyecto y seleccionar el archivo de datos. Una vez lo abra, aparecerá la siguiente pantalla:

3. Si desea añadir datos, puede hacerlo en esta primera pestaña.
4. El siguiente paso, es seleccionar los parámetros del vídeo. Todos los pasos son muy importantes para evitar que dé error.



5. Por ejemplo, en este caso se han puesto 450 mm, considerando que la caja mide aproximadamente 45x45 cm. Clique en “auto-populate”.
6. Seguidamente, clique en “config. video parameters”.
7. Se abrirá una nueva pestaña.



8. Seleccione “calculate distance”.
9. Se abrirá una imagen de su vídeo. Presione dos puntos laterales:



10. Una vez terminado, presione “ESC” para guardar su selección y salir.
11. Volverá a la pestaña anterior. Si tiene más de un vídeo, repita los pasos. Si no, presione “save data” y cierre la pestaña.
12. La siguiente pestaña es “outlier correction”. Seleccione “settings”.
13. Se abrirá una nueva pestaña:

OUTLIER SETTINGS

LOCATION CORRECTION

Choose Animal_1 body part 1:

Choose Animal_1 body part 2:

Choose Animal_2 body part 1:

Choose Animal_2 body part 2:

Location criterion:

MOVEMENT CORRECTION

Choose Animal_1 body part 1:

Choose Animal_1 body part 2:

Choose Animal_2 body part 1:

Choose Animal_2 body part 2:

Location criterion:

AGGREGATION METHOD

Aggregation method:

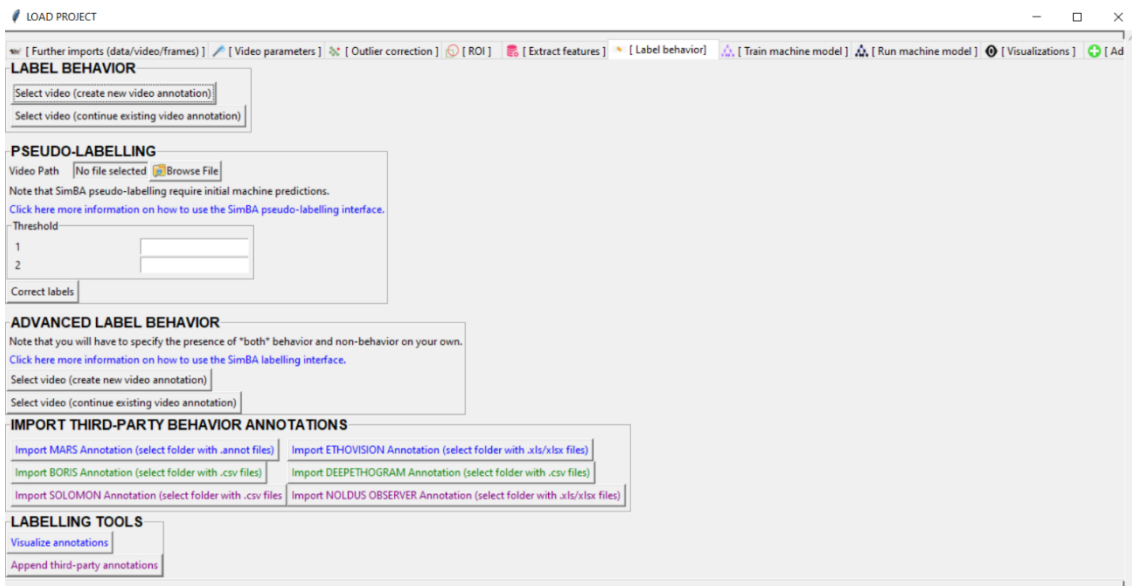
CONFIRM

14. Seleccione las partes del cuerpo de cada animal que más le interesen. En “location correction” escriba un valor entre 0.7 y 1.5. Cuanto más grande, más demandante será su selección.
15. En “movement correction” se recomienda un valor superior a 1.5.
16. Clique “confirm” y cierre la pestaña.
17. Ahora puede pulsar “run outlier correction”.
18. Una vez finalizado, pase a la pestaña de “extract features”. Si todo ha sido realizado correctamente. Este paso se realizará de forma fácil y automática.

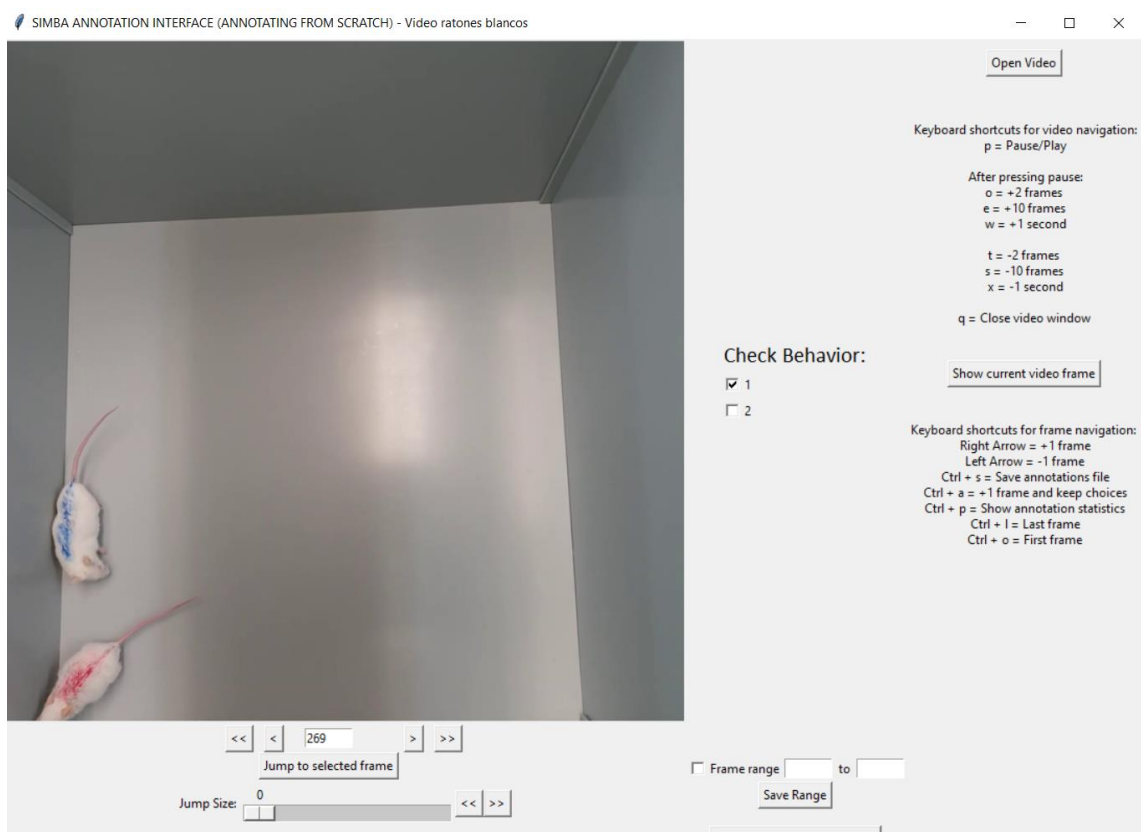
Dependiendo del caso en el que uno se encuentre los siguientes pasos serán diferentes.

ESCENARIO EN EL QUE SE ENTRENA POR PRIMERA VEZ LA RED.

1. Clique en “label behavior”.



2. Seleccione “select video (create new video anotation)”.
3. Se abrirá una ventana en la que deberá seleccionar el vídeo con el que se ha estado trabajando.
4. La siguiente pestaña será con la que trabajará.



5. Deberá ir seleccionando, “frame” por “frame” o por conjunto de “frames” el comportamiento que se esté dando y que se desea evaluar. **Recuerde:** los

comportamientos que se desea evaluar y su nombre se establecen al inicio, en la creación del proyecto.

6. Recuerde, cada poco, seleccionar “control” + “s” para guardar sus anotaciones. La GUI tiene fallos cada cierto tiempo y puede cerrarse de forma inesperada. Cuando termine, vuelva a pulsar las teclas “ctrl”+ “s”. Ahora cierre la pestaña.
7. Pase a la siguiente pestaña “train machine model”. Clique “settings”.
8. Aparecerá la pestaña de la imagen.
9. Se recomienda insertar los valores que se muestran en la imagen. Cuando termine, guarde en “save settings into global enviroment”. Después, presione “train single model (global enviroment)”.
10. Cuando termine, abra de nuevo, “settings” y seleccione, en esta ocasión, otro comportamiento que desee entrenar. Inserte los parámetros anteriores y repita el proceso tantas veces como comportamientos necesite entrenar.
11. Una vez terminado, puede pasar a “run machine model”. Baje hasta encontrar la siguiente imagen:

HYPER-PARAMETERS

Random forest estimators: 2000

Max features: sqrt

Criterion: gini

Test Size: 0.2

Train-test Split Type: BOOTS

Minimum sample leaf: 1

UNDER-sample setting: None

UNDER-sample ratio:

OVER-sample setting: SMOTE

OVER-sample ratio: 1

CLASS WEIGHTS

1 PRESENT: 2

1 ABSENT: 1

MODEL EVALUATION SETTINGS

Create model meta data file

Create Example Decision Tree (requires "graphviz")

Create Fancy Example Decision Tree (requires "dtreeviz")

Create Classification Report

Create Features Importance Bar Graph

Features: 15

Compute Feature Permutation Importances (Note: CPU intensive)

Create Learning Curves (Note: CPU intensive)

Learning Curve Shuffle K Splits:

Learning Curve Shuffle Data Splits:

Create Precision Recall Curves

Calculate SHAP scores

target present:

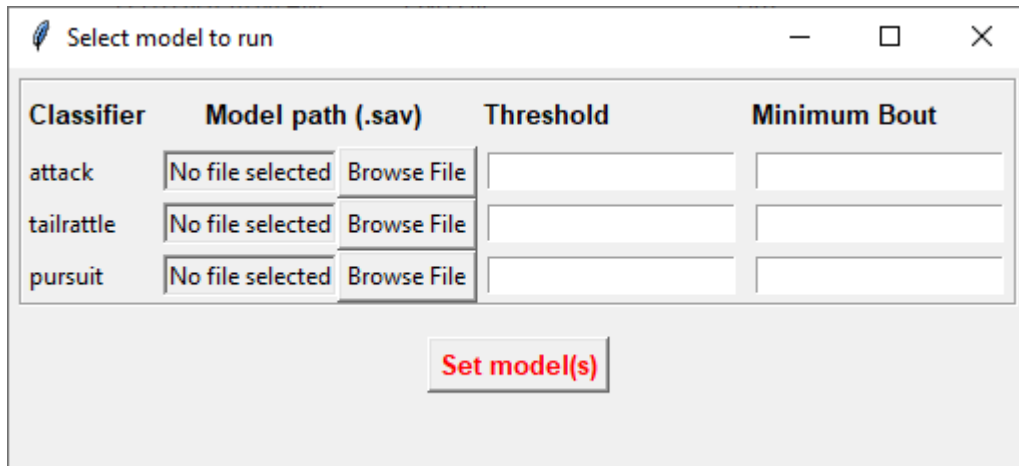
target absent:

SAVE

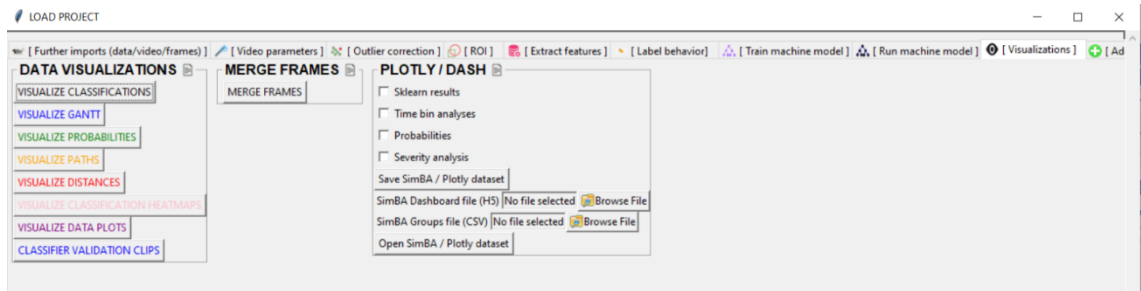
Run Machine Model

Model Settings

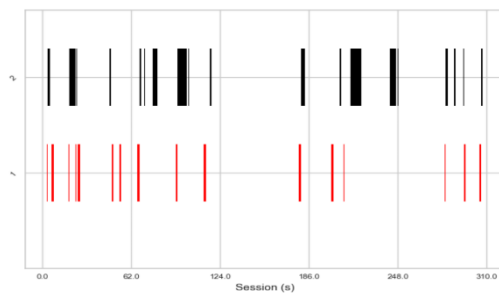
Run RF Model



12. Una vez la encuentre, seleccione “model settings”. Ahora deberá utilizar los datos entrenados anteriormente.
13. Le aparecerá una pestaña similar a la siguiente:
14. Seleccione “browse file” y busque en su carpeta del proyecto el modelo generado. Es un archivo terminado en “.sav”. Debería encontrarse en la carpeta del proyecto, dentro de la carpeta “models”.
15. Asocie cada uno de los comportamientos a su debido archivo.
16. Para “discrimination threshold” seleccione un valor entre 0 y 1. Si eligiera, por ejemplo, 0.7, todos los “frames” en los que exista una probabilidad de 0.7/1 de contener ese comportamiento serán clasificados como si lo contuvieran.
17. “Minimum behavior bout length” es más complicado. Para comprenderlo se debe tener en cuenta que en ocasiones el mismo comportamiento no ocurre durante 30 “frames” seguidos, sino que puede ocurrir 15, parar durante 2 y continuar otros 15. Por ello, en este apartado se puede seleccionar cuánta separación entre “frames” consideramos como dos comportamientos individuales en lugar de un comportamiento largo. El valor está en milisegundos.
18. Puede poner el valor que desee en cada comportamiento o todos con los mismos valores. Al finalizar, pulse “Set models”.
19. Cierre la pestaña.
20. Seleccione “run RF model”.
21. Una vez finalizado, ya puede pasar a la pestaña donde crear los resultados del análisis: “visualizations”.



22. Dependiendo de los resultados en los que esté interesado deberá seleccionar una de las opciones en “data visualizations”. Cada pestaña abrirá un comando diferente para crear bien sea un vídeo o “frames” individuales de acuerdo a sus necesidades.
23. En este caso, para los objetivos del TFG, los más útiles son “visualize classifications” y “visualize gantt”. El primero permite crear un vídeo con un contador de la duración de cada comportamiento, el segundo, un vídeo con las detecciones de cada uno de los comportamientos de interés:

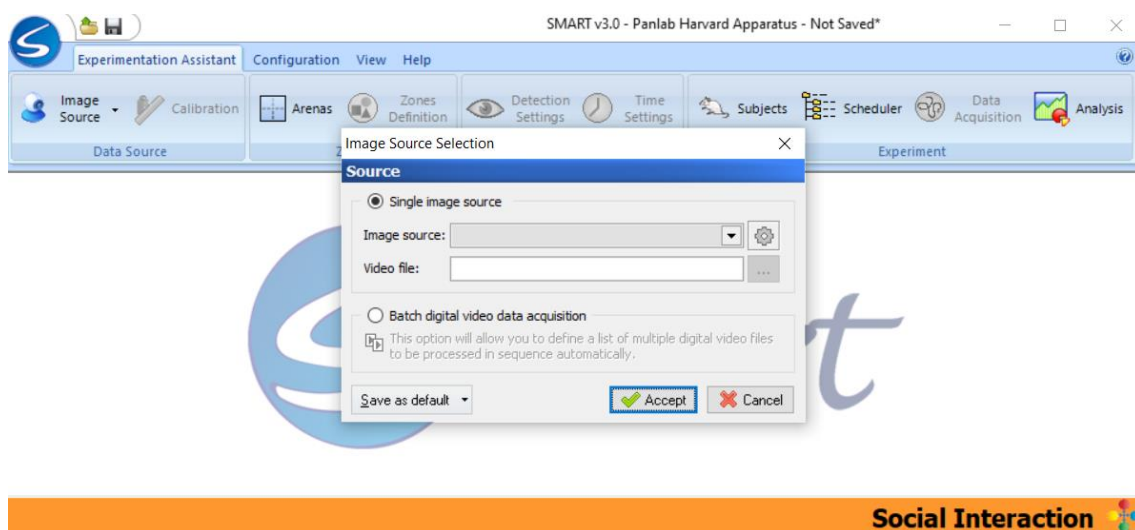


Anexo IX. Análisis de video con “SMART”.

1. Abra la aplicación.
2. Dependiendo de la versión que tenga, las opciones que aparecerán serán diferentes. En este caso, se muestran las opciones para la versión completa:
3. Seleccione “social interaction”.
4. Se abrirá una nueva pestaña con la información. Seleccione “aceptar”.

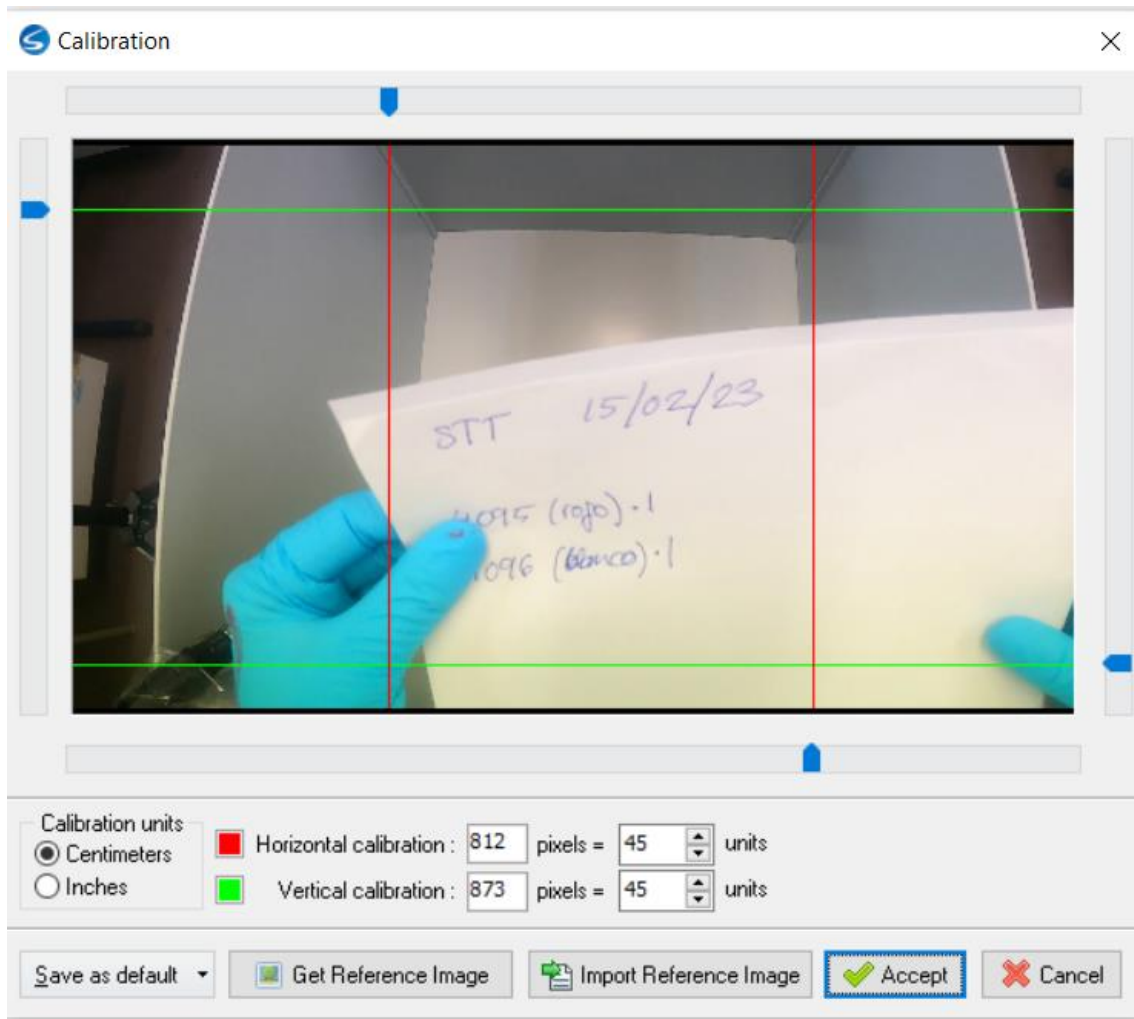
*Recuerde seleccionar “accept” cada vez que termine una tarea.

5. Seguidamente, pinche en “image source”.
6. Aparecerán diversas opciones. En este caso, seleccione en el primer desplegable (“image source”: “video file”.
7. Después, en “video file”, clique sobre los tres puntos para buscar el vídeo que desea utilizar.

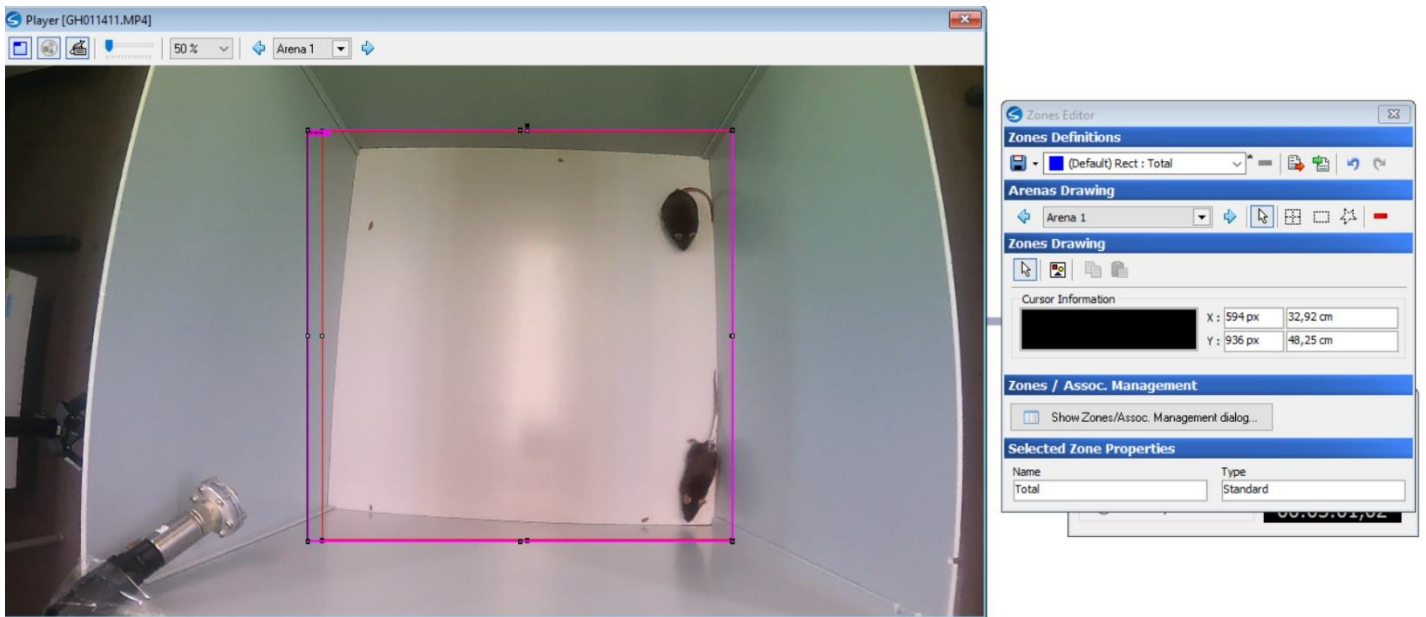


8. Clique “accept”.
9. Se abrirá el vídeo con el que desea trabajar.
10. Ahora clique en “calibraton”.

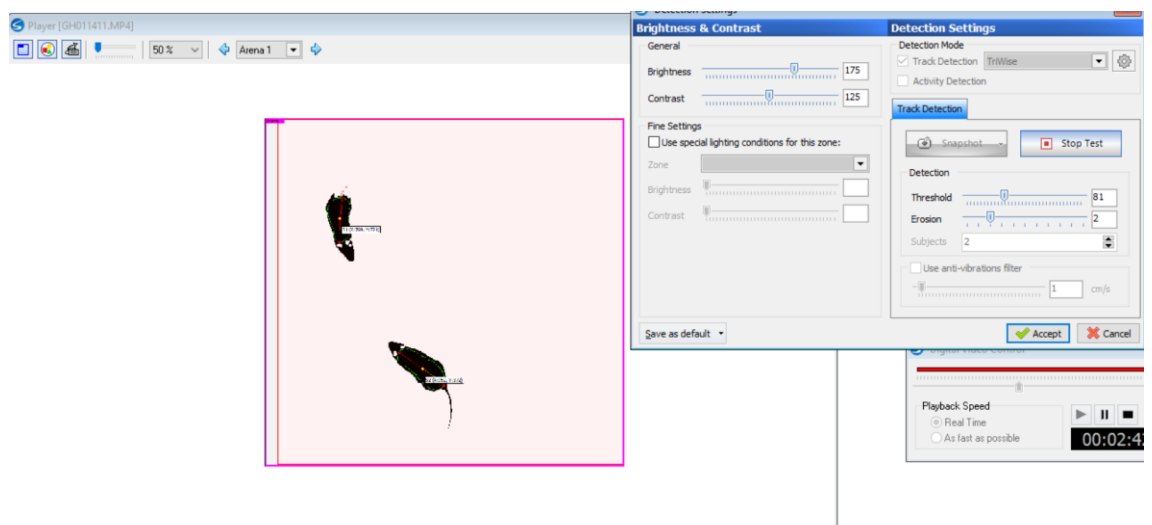
11. Aparecerá una pestaña donde deberá seleccionar los márgenes de la arena. Además, deberá marcar las dimensiones de esta. En este caso, 45x45 cm.



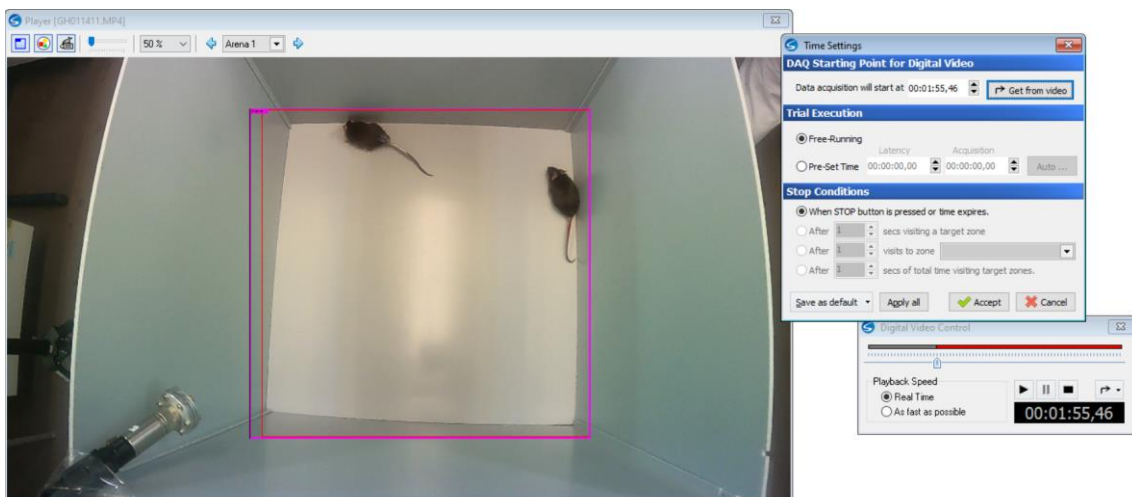
12. Clique “accept”
13. En “arenas”, seleccione la arena en la que se encuentra o cree una nueva con la forma utilizada en su experimento.
14. Pase ahora a “zones definition”. En la pantalla en la que se muestra el vídeo, ajuste los bordes de modo que coincidan con los bordes de su arena.



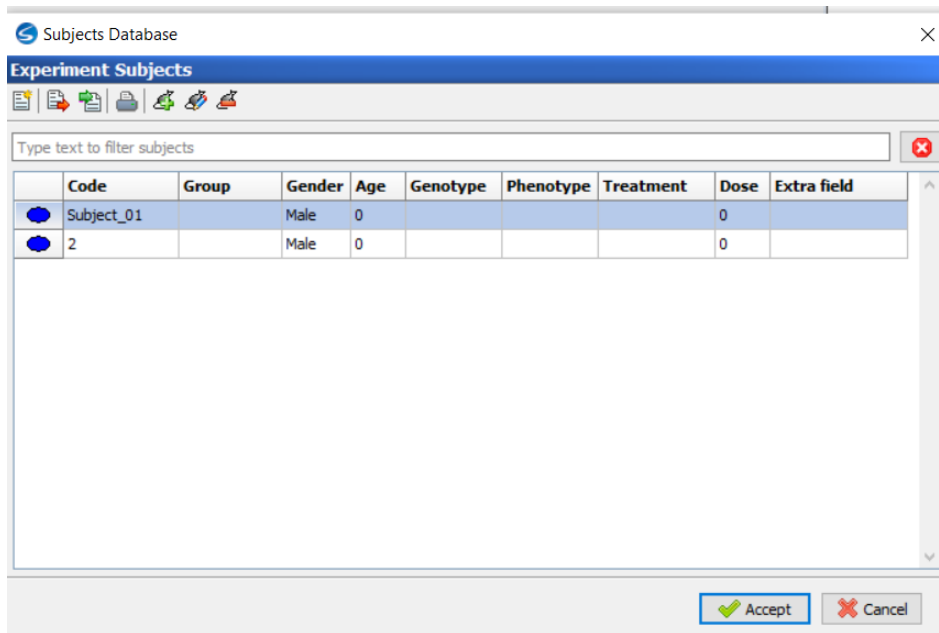
15. Seleccione guardar (pequeña caja al lado del cuadrado azul en este caso).
16. Pase a “detection settings”.
17. Ahora debe ajustar los diferentes parámetros del vídeo para asegurar la correcta detección.
18. El primer paso es realizar una captura de la arena sin ninguno de los ratones. De este modo, se puede determinar el fondo.
19. Busque un momento del vídeo en el que no haya ningún ratón y seleccione “snapshot”.



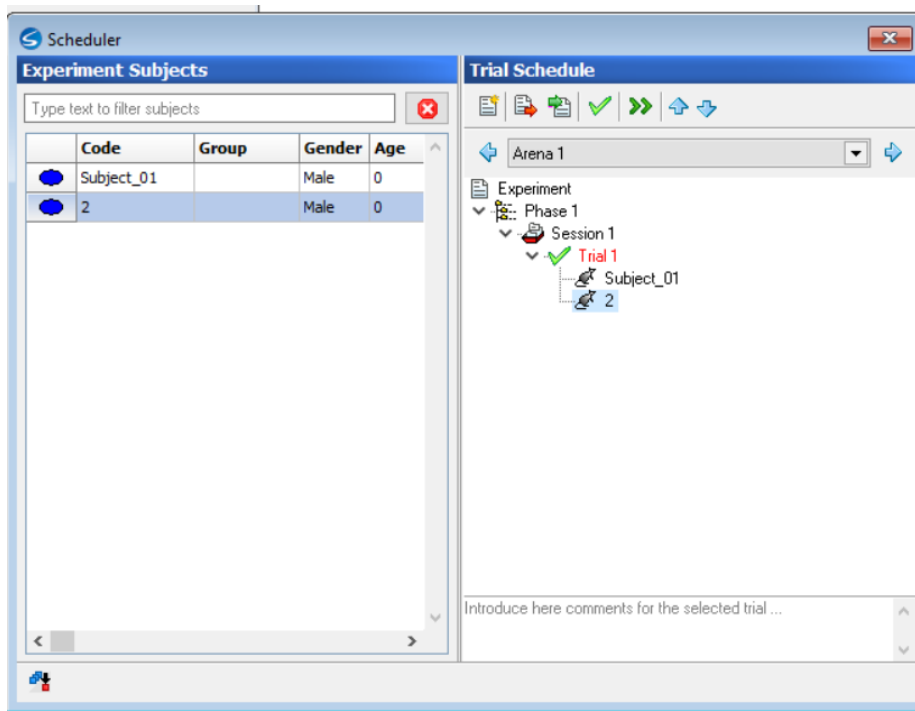
20. Ahora, si dispone de la versión completa, se recomienda seleccionar “TriWise” como método de detección (“track detection”). De este modo se consiguen mejores resultados.
21. Después, ajuste el “threshold” (umbral) de forma que se asegure de detectar sólo a los ratones. Un posible problema es la detección de heces, por ejemplo). En este caso, se ha visto 81 como el nivel más adecuado.
22. Ajuste también la “erosion” y el número de sujetos. En este caso, en erosión, lo más adecuado fue 2 y el número de sujetos también fueron 2.
23. Ahora, seleccione “time settings”. Busque en el vídeo el primer momento en el que se encuentran los dos ratones en el vídeo.
24. Clique “get from the video”.



25. Pase a “subjects”. Si tan sólo tiene un ratón, clique sobre el dibujo de un “+”. Agregue un segundo ratón o los que sean precisos.



26. En “scheduler”. Añada el segundo ratón al “Trial 1”. De este modo, se tendrán en cuenta ambos ratones para el análisis:



27. Asegúrese de que está en verde, si no, no se podrá realizar el análisis.

28. Pase a “data acquisition”, seleccione “start”.

29. Espere a que termine el análisis.

30. Cuando finalice el análisis puede presionar sobre el pequeño icono (se encuentra redondeado en la imagen) con dos ratones. Aquí podrá obtener el número de contactos por minuto y su duración.

Runtime Viewer



Arena Selector

Arena 1

Subject Data

	Subject 1	Subject 2
Subject Code:	Subject_01	2
Travelled Distance:	114,95 cm	183,73 cm
Current Zone:	Total	Total
Speed:	7,47 cm/s Status: >	4,10 cm/s Status: >
Rearing:	2	0
Rotations:	0 CW 1 CCW	0 CW 0 CCW
Stretchings:	2	2

Trial Data

Contacts:   3

Protocol Specifics

Permanency Time

	Total
Time S1	21,24
T. % S1	100,00
Time S2	21,24
T. % S2	100,00

31. Los datos se mostrarán de la forma siguiente:

Social Interaction Details

Arena Selector

Arena 1

Relative Position Events

Nr.	Start Time	Subjects	Cont. Pnts.	Rel. Pos.	Duration
1	00:00:00,00	S1-S2	Center-Center	Far	3,76
2	00:00:03,76	S1-S2	Center-Center	Close	0,12
3	00:00:03,88	S1-S2	Center-Center	Contacting	4,84
4	00:00:08,72	S1-S2	Center-Center	Far	10,20
5	00:00:18,92	S1-S2	Center-Center	Close	0,14
6	00:00:19,06	S1-S2	Center-Center	Contacting	7,24
7	00:00:26,30	S1-S2	Center-Center	Far	18,02