



Diagramas Color Magnitud Sintéticos del Grupo Local para estudiar el ritmo de formación estelar

Enrique Galceran García

16 de diciembre de 2018

Resumen

One of the many things that can be analyzed from our Local Group is the Color Magnitude Diagrams (CMD). Our objective is to create several possible Star Formation Rates (SFR) and compare the CMD we obtain for each individual SFR with our problem galaxy. To compare our two CMDs, we will use a normalized density mesh with a convolutional layer on top of it and minimize our 'cost function'.

Índice

1. Introducción	1
2. Fundamentos Teóricos	2
3. Metodología	2
3.1. Mallas Estáticas	2
3.2. Capa de Convolución	3
3.3. Utilidad de la Capa de Convolución	3
4. Resultados	4
5. Conclusiones	5
5.1. Tiempo de ejecución	5
5.2. Posibles mejoras	5
5.3. Conclusión y Comentario	6
Referencias	6
A. Contenido Complementario	6
B. Imágenes Adicionales	1

1. Introducción

1

2

2

4

5

5

6

6

6

1

Los Diagramas Color Magnitud (CMD del inglés) son unas herramientas muy útiles que nos permiten inferir el ritmo de formación estelar de una galaxia. Las estrellas evolucionan en grupos formados por los diferentes brotes de formación estelar a lo largo de su vida. En una galaxia donde sólo hay un brote único corto se puede ajustar de tal forma que se pueda ver dónde se encuentran las diferentes estrellas en estos diagramas CMD.

Como se puede apreciar, las diferentes fases de las estrellas se pueden distinguir sin demasiada dificultad. Así pues, después de medir una CMD en teoría podríamos directamente delimitar las diferentes zonas, contar la cantidad de estrellas que medimos en cada lugar y de allí sacar la SFR de la galaxia.

Sin embargo eso no será del todo posible. Existe una gran cantidad de fuentes de error que van a dificultar hacer este recuento. En un caso realista las zonas no están tan limpiamente definidas (figura 1)^[3]. En esos casos tendremos que añadir estas posibles fuentes de error a nuestra simulación para que se tenga en cuenta y podamos adaptar nuestras CMDs sintéticas a los resultados que vamos a ver.

Sin embargo, estos diagramas dan una distribución generalizada de la edad, estando altamente degenerado. La dispersión debido a la metalicidad, función inicial de masas (IMF) o luminosidad hace que haya estrellas de diferentes edades situadas en la misma zona. Para poder discernir entre las diferentes estrellas individualmente habrá posteriormente que usar los diferentes trazadores para hacer un análisis más minucioso. Para el caso general de la galaxia, usar el CMD es suficiente.

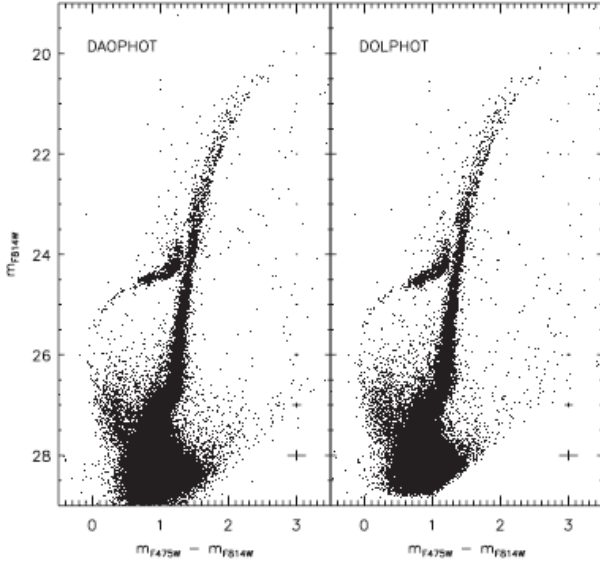


Figura 1 – Diagramas CMD no sintéticos. Se puede apreciar cómo las distribuciones no están tan bien definidas como en los casos sintéticos (fig. 3).

2. Fundamentos Teóricos

Para poder analizar las diferentes zonas del diagrama, debemos primero ver cómo evolucionan las estrellas desde que nacen hasta que mueren. Las estrellas jóvenes empiezan evolucionando a lo largo de la secuencia principal, pasando después en función de la masa que tengan a las ramas de las subgigantes, gigantes, enanas blancas, etc. (figura 2)^[4].

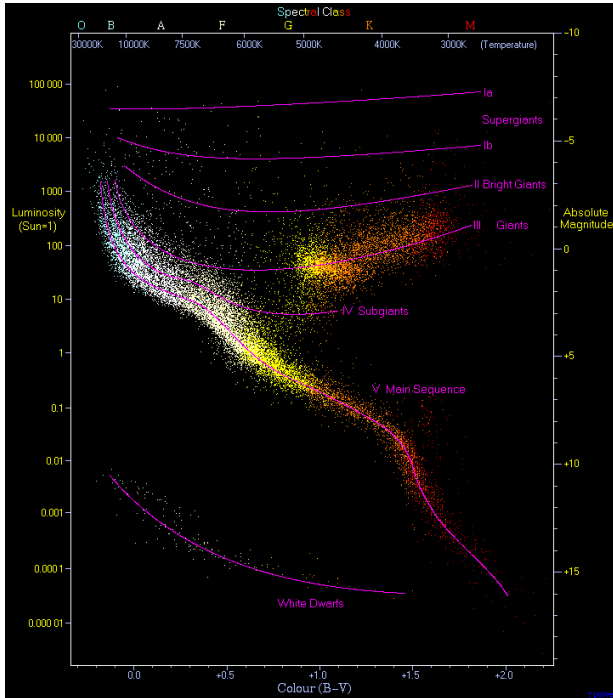


Figura 2 – Diagramas HR de la distribución de las estrellas en función de su clase.

Sabiendo cómo van a evolucionar las estrellas a lo largo de su vida, podemos generar un CMD con pequeños brotes de formación en diferentes momentos de la galaxia y separar así las diferentes zonas que poblarán en nuestro diagrama en función de su edad (figura 3)^[1].

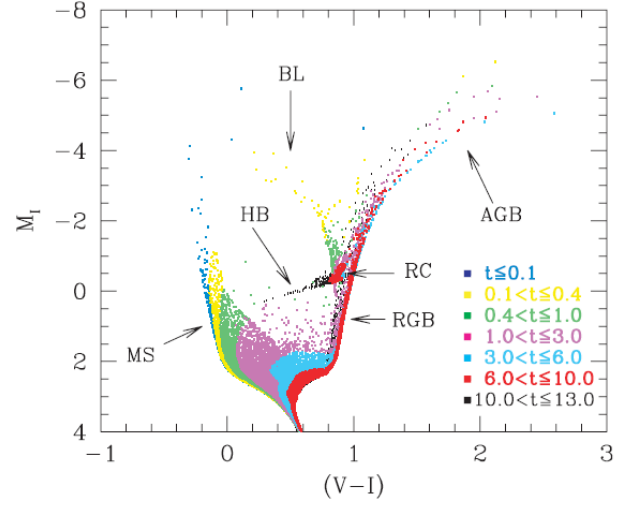


Figura 3 – Diagramas CMD con una formación estelar constante

Viendo cómo se han poblado las diferentes zonas, podremos hacer una primera aproximación de la formación estelar de las galaxias problema que queramos analizar. La densidad de estrellas en cada zona también representará la intensidad de los brotes de formación de la galaxia. Es por eso que no basta con simplemente sacar cuales son las zonas pobladas, si no que tendremos que decidir un método a usar para comparar los CMDs sintéticos que obtenemos con nuestra hipótesis con el CMD problema. Hay varias formas diferentes de hacer esta comparación y no existe un método que sea intrínsecamente mejor que otro, si no que variarán la rapidez, precisión y forma en la que compare (distribución de estrellas en diferentes zonas del diagrama, comparación de densidad por mallas estáticas o adaptativas, etc.).

3. Metodología

Como hemos mencionado en la sección 2, hay varios métodos diferentes para comparar los diferentes diagramas CMD. En nuestro caso, hemos empleado mallas estáticas con una capa de convolución encima para mejorar el ajuste.

3.1. Mallas Estáticas

Las Mallas estáticas generan una rejilla a lo largo de la zona que nos interesa con un tamaño fijo dada por la resolución. Después se reparten las estrellas en función de sus variables M_I y $V - I$ en las diferentes casillas. La forma más sencilla de realizar esto es ordenando las estrellas en función de una de estas variables e ir avanzando poco a poco, repartiendo en la otra dirección en función de la segunda variable. No es la versión más rápida ni la más eficaz, pudiendo hacer una interfaz que vaya peinando en función de la densidad de las zonas, pero cuando estamos utilizando *arrays* para las matrices con un programa que se aproveche del principio de la localidad secuencial (*Locality of reference* o *Principle of Locality*^[5]) es capaz de realizar estos cálculos a una velocidad muy rápida (del orden de 0.8s para una resolución de 800x800 y 500.000

estrellas)¹.

A una resolución baja vamos a tener pocas casillas en la malla con una gran cantidad de estrellas por cada casilla, luego se harán comparaciones a grosso modo. Estas comparaciones de baja resolución serán útiles al principio, para hacernos una idea general de dónde colocamos los brotes de formación. Sin embargo, llega un punto en el que si queremos comparar la cantidad de estrellas que hay en una zona u otra se vuelve imposible porque todas caerían en la misma casilla.

La solución a esto es aumentar la resolución. Sin embargo, aumentar la resolución mete una nueva fuente de error. Como estamos comparando la densidad de distribución de las estrellas en ambos CMDs, nos vamos a encontrar con que si una estrella ha caído justo en la casilla de al lado y cuenta como que están mal. Anteriormente no hacía falta preocuparnos de este problema porque al tener una resolución baja hay pocas casillas que comparar y a nivel estadístico eso no es muy importante.

Al aumentar la resolución, sin embargo, esto deja de ser un error despreciable. El problema reside en que se está comparando cada casilla con su análoga sin tener en cuenta las que están alrededor de ellas. Entra aquí la necesidad de incorporar una capa de Convolución.

3.2. Capa de Convolución

Las capas de convolución tienen muchas aplicaciones en un gran número de campos por su utilidad, desde redes neuronales para reducir la cantidad de nodos sin perder demasiada información, al desenfoque Gaussiano en edición de imágenes.

La idea principal consiste en darle un valor a cada punto en un mapa en función de los valores que se encuentren alrededor. El método que vamos a usar aquí utiliza la distribución normal como función de peso (se puede usar otro tipo de distribución o incluso un valor constante). Esto es porque el valor que se encuentra al lado de la coordenada central debe ser más importante que el que se encuentra al doble de la distancia. El efecto que conseguimos es sustituir cada estrella en nuestro diagrama CMD en una pequeña distribución normal (limitada a un radio máximo). El efecto que buscamos se puede apreciar mejor en un caso más pequeño (figura 4 - 6).

El radio es la cantidad de pixels alrededor del pixel central que se tiene en cuenta a la hora de expandir los puntos (i.e. un radio de 1 cada punto se reparte en el cuadrado de 3x3 centrado en el punto original y un radio de 2 se repartiría en un cuadrado de 5x5). Aumentar sigma aumenta el peso de los puntos más lejanos. Ahora hay que realizar pruebas y encontrar los valores más indicados para las diferentes resoluciones. A medida que aumentamos la resolución, debemos aumentar el radio de influencia de la convolución así como sigma.

3.3. Utilidad de la Capa de Convolución

Una buena forma de visualizar las razones por las que estamos haciendo este proceso en vez de simplemente comparar casilla a casilla la densidad de puntos es coger un

¹En casos como este, realmente no hace falta preocuparse demasiado por la velocidad de cálculo cuando tarda más en cargarse la interfaz gráfica para mostrar los resultados que en realizar los cálculos.

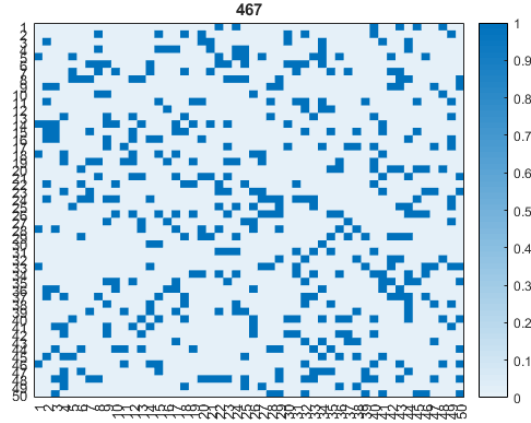


Figura 4 – Distribución aleatoria de puntos en un cuadrado de 50x50.

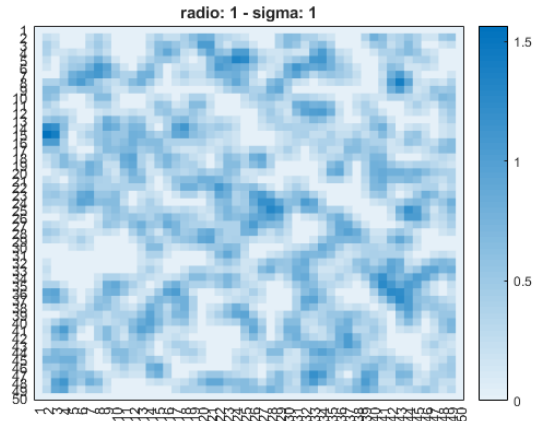


Figura 5 – Misma distribución después de pasarle un filtro de radio = 1 y $\sigma = 1$

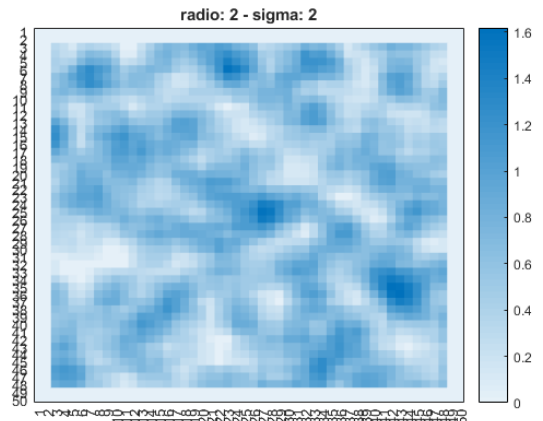


Figura 6 – Misma distribución después de pasarle un filtro de radio = 2 y $\sigma = 2$

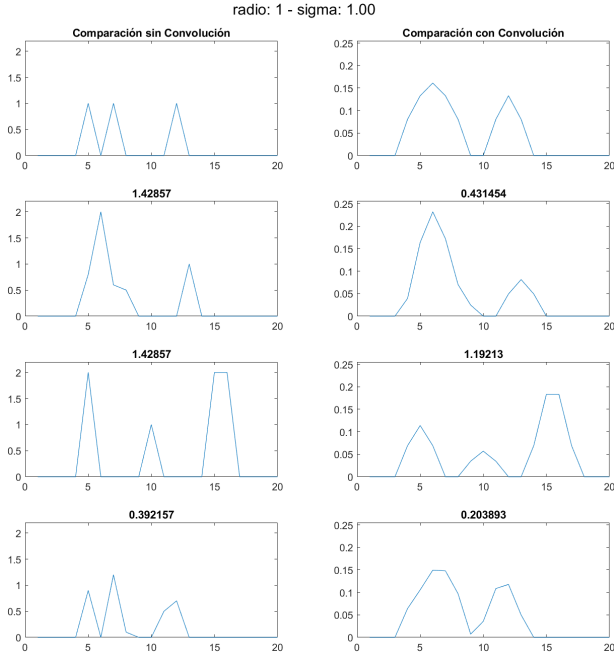


Figura 7 – Comparación entre no usar y usar una capa de convolución

ejemplo en una dimensión. En la figura 7 tenemos en la primera fila el caso que estamos buscando, siendo la columna de la derecha el resultado de hacer la convolución. En las filas 2, 3 y 4 podemos ver tres posibles valores. Los rótulos de dichas gráficas nos muestran el *valor de ajuste*; la suma del valor absoluto de la diferencia en cada uno de los puntos de la recta. Como se puede observar, aunque la fila 3 sea significativamente peor con respecto al caso original, tiene el mismo error que la fila 2.

La fila dos representa la unión del doblete, pero como se encuentra entre los dos puntos, no puntúa en absoluto y se considera que está mal. Aquí es donde entra la convolución. Después de tener en cuenta los valores vecinos, el valor de ajuste de la fila dos es tres veces menor que el de la fila 3. Por otro lado, si ponemos un valor más cercano al correcto, también se puede apreciar cómo mejora al hacer la convolución.

En definitiva, haber usado una convolución nos permite tener una comparación que tenga en cuenta también a los vecinos, siendo todavía capaz de distinguir entre un valor bueno y uno muy bueno.

4. Resultados

Las resoluciones que hemos utilizado se pueden ver en la tabla 1. Como se puede observar, aumentamos tanto el radio como la desviación típica junto a la resolución para que las casillas que se tienen en cuenta no se hagan demasiado pequeñas.

El código que hemos desarrollado nos muestra por pantalla os datos más importantes de nuestra simulación (las figuras 8 y 9 representan los dos casos que mejor se acercan de los SFH analizados): La zona de arriba a la izquierda nos muestra el SFH además del tiempo total que ha tardado la simulación en ejecutarse, la zona de la derecha nos muestra una superposición de los los CMDs para poder compararlos visualmente además del ID de la SFH, el nú-

resolución	radio	sigma
50	1	0.5
100	2	1.0
400	3	1.0
800	3	2.0
1000	4	2.5
1500	5	3.0
2000	7	5.0
2500	10	5.0

Tabla 1 – Resoluciones, radios y desviaciones típicas usados.

mero de partículas de la simulación y la resolución espacial que tenemos, y abajo a la izquierda tenemos la distribución de densidad para cada una de las casillas además del valor de ajuste.

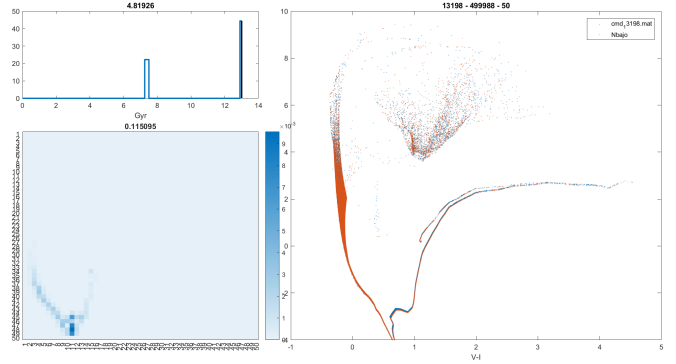


Figura 8 – Salida por pantalla del SFH con ID 13198

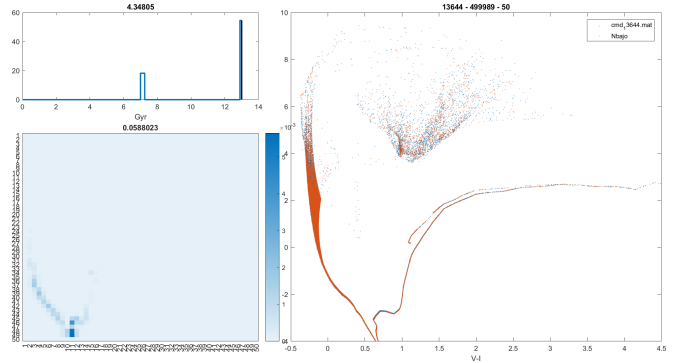


Figura 9 – Salida por pantalla del SFH con ID 13644

A medida que aumenta la resolución se puede distinguir mejor la diferencia entre cual de los dos SFH es mejor. Las gráficas de las Distribuciones de densidad y comparación se encuentran en el apéndice en las páginas I-II (figuras 14 y 15). Si realizamos el cálculo con los 14 SFH que hemos tomado, podemos representar la eficacia con respecto a la resolución. Como hemos calculado la eficiencia antes y después de incorporar la convolución, podemos comparar ambos resultados (figuras 10 y 11). Como se puede apreciar, Aumentar la resolución sin hacer la convolución mejora un poco la comparación, pero tiene la limitación de que no tiene en cuenta lo que hay a su alrededor. Una vez le metemos la convolución podemos ver como claramente hay un caso que destaca entre los demás. Además, Aumentar la resolución facilita separar entre la calidad de

los diferentes SFH.²

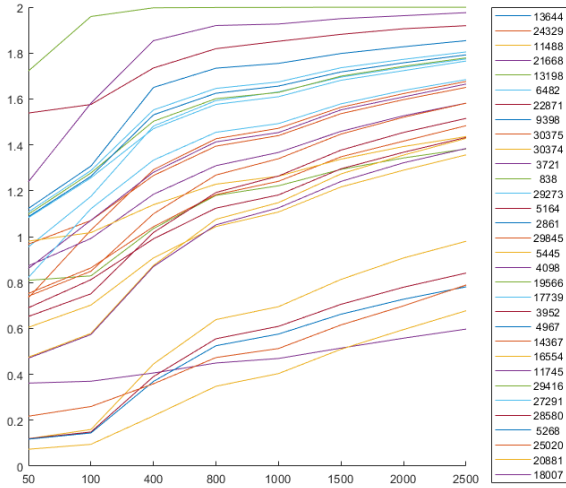


Figura 10 – Eficiencias antes de hacer la convolución.

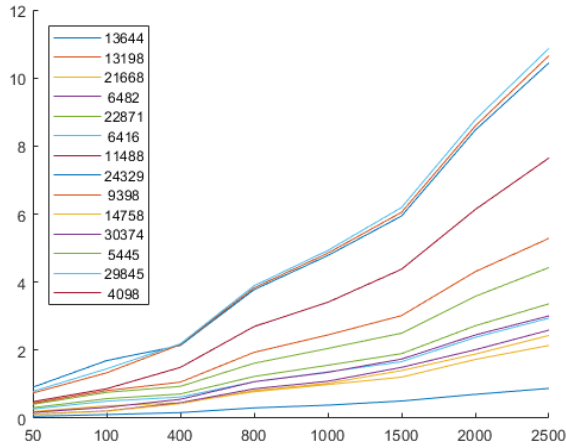


Figura 11 – Eficiencias después de hacer la convolución.

Con estas dos últimas figuras podemos llegar a la conclusión de que efectivamente el mejor caso es la SFH con ID:13644 y la segunda mejor será ID:13198; siendo la mejor con dos brotes de 0.1 Gyr de duración, uno en 7 Gyr y otro en 12.9 Gyr con el segundo brote siendo el triple de intenso que el primero.

5. Conclusiones

5.1. Tiempo de ejecución

Para seguir aumentando la resolución del resultado debemos tener en cuenta que la simulación que usamos utiliza una semilla pseudoaleatoria, luego debemos repetir todas las simulaciones que podrían ser las correctas unas cuantas veces para poder sacar un valor promedio de nuestra función de comparación.

²En la sección 5.1 se habla más en detalle sobre el tiempo de ejecución. Es por eso que para la parte de convolución se ha reducido las SFH a los 14 mejores.

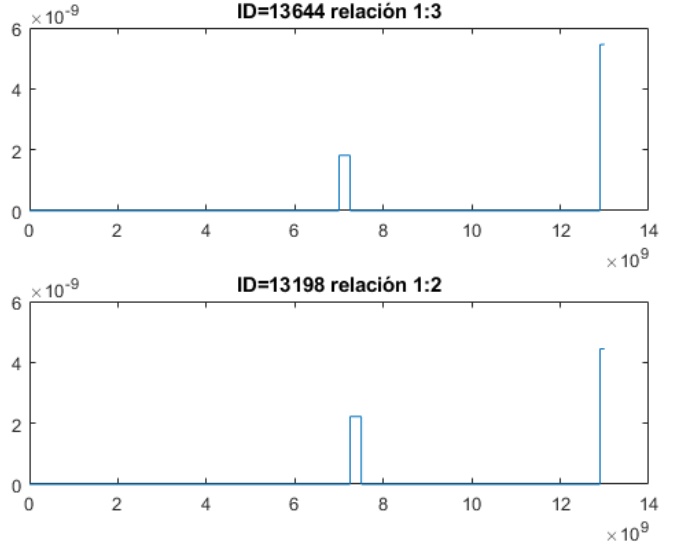


Figura 12 – SFH de las simulaciones 13644 y 13198.

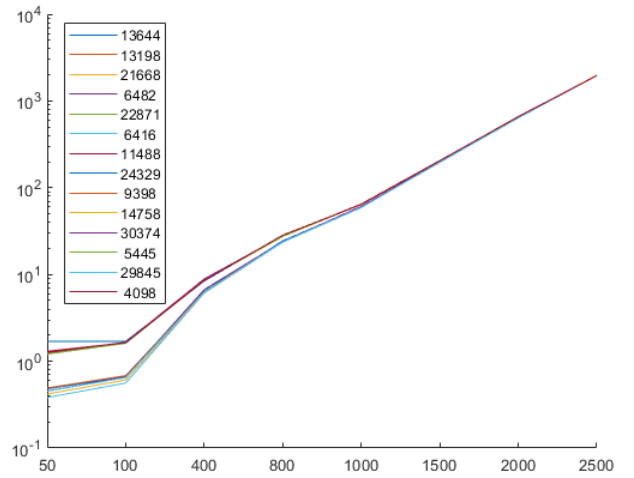


Figura 13 – Tiempo de ejecución en función de la resolución (resolución no a escala).

Como podemos ver, a medida que aumentamos la resolución de nuestra comparación nos vamos encontrando con un nuevo problema: el tiempo de ejecución. Los cálculos para resoluciones pequeñas tardan menos de un segundo, pero a medida que aumentamos la resolución y obtenemos soluciones más precisas el tiempo en realizar el cálculo crece de forma exponencial (figura 13).

Si nos fijamos en la figura 11, podemos ver que la distinción efectivamente se hace más clara y pronunciada a medida que aumentamos la resolución, pero en realidad no es necesario aumentarla tanto. El tiempo de procesamiento para las 14 SFHs a resolución=800 es de 6 minutos, mientras que tardó 7.5 horas en calcularse para resolución=2500. De cara al futuro, salvo en el caso donde los SFH sean extremadamente parecidos, no hace falta aumentar tanto la resolución.

5.2. Posibles mejoras

Uno de los mayores impedimentos de cara a la agilidad del código está en la obtención de la distribución de probabilidad gaussiana. A nivel de código es muy sencillo obtener la función de distribución gaussiana (`normpdf(d,0,sigma)`) donde d es la distancia al centro

y σ es la desviación típica de la distribución normal), pero la función en sí es muy lenta. Una posible mejora futura sería calcular con anterioridad la distribución de la cuadrícula de puntos que se tiene en cuenta en la convolución, pero a la hora de programarlo ha resultado dar problemas. Matlab permite ejecutar el código y cronometrar el tiempo de ejecución de cada línea (figura ??), luego podemos ver que para calcular 19 capas de convolución ha requerido $\sim 60\%$ del tiempo en realizar una y otra vez el cálculo de la gaussiana. Corregir esto, reduciría el tiempo de ejecución para resolución=2500 a algo menos de 3 horas.

Las mallas adaptativas ayudarían mucho a agilizar el código. Pese a que efectivamente haya estado el $\sim 60\%$ del tiempo obteniendo la distribución normal, hay que tener en cuenta que ha realizado la operación casi dos mil millones de veces, tardando sólo 2 microsegundos a cada vez. Si utilizamos mallas adaptativas las matemáticas de la capa de convolución se volverían mucho más complicadas y aún más difícil de programar, pero sería mucho más rápido y no haría falta precalcular la matriz de densidad mencionada en el párrafo anterior.

Otra forma de agilizar el código sería operar con la GPU. Para calcular la convolución se puede usar varios núcleos pues los resultados sólo dependen de los datos iniciales, pero mi nivel de programación no llega tan lejos.

Uno de los métodos que se empezó a utilizar muy al principio consistía en encontrar la distancia entre los vecinos más cercanos para cada estrella. Sin embargo, no precisábamos de suficiente tiempo como para desarrollar ese código. La idea consistía en formar parejas entre las estrellas más cercanas e ir ampliando la distancia entre las estrellas y encontrar así un valor que representara la distancia entre las estrellas cercanas. La idea al final se abandonó en parte porque iba a depender mucho de las estrellas colocadas en la zona de las gigantes rojas (zona superior del CMD), pues tenía una densidad muy baja de estrellas y estaban es muy improbable que encontrar una estrella cercana. Para corregir esto se podría poner una máscara de las estrellas que se tenían en cuenta y así obtener un valor de distancia para las diferentes zonas y analizarlas por separado.

5.3. Conclusión y Comentario

La forma de comparar estos datos se aprovecha de las dos cosas que mejor hace un ordenador: repetir muchas veces una acción y operaciones con números. 500.000 puntos para un ordenador moderno es poco, permitiendo así hacer rápidamente operaciones complicadas rápidamente. Una de las primeras acciones que se hicieron en este proyecto consistía en reescribir los datos a un formato más manejable. La simulación de CMDs saca una gran cantidad de información, pero genera por ello archivos de texto muy pesados (105MB), luego la primera parte consistía en reducir los datos a sólo los archivos que íbamos a necesitar (a archivos .mat, que usan la localidad secuencial^[5] y pesan 28MB).

A nivel de cálculo, como se ha mencionado en la sección 5.2, podríamos mejorar significativamente la rapidez del código para hacerlo más preciso. Sin embargo la falta de tiempo y complicación del código resultante ha hecho que finalmente nos decantemos a usar mallas estáticas. Paradójicamente, el tiempo de ejecución de estos códigos

depende poco del número de estrellas que hay, pues el tiempo en repartir los datos por la malla es de menos del 5 % para las resoluciones altas. Una vez repartidas las estrellas se operan con los valores de las celdas de la malla.

Independientemente del sistema que estemos usando para operar los datos, en realidad podríamos obtener la SFH de nuestra galaxia a mano: simplemente es cuestión de dibujar la distribución de las estrellas de la galaxia y comparar la cantidad de estrellas que hay en ambas ramas. El tiempo de los brotes se puede sacar usando la figura 3 y ensayo y error. Para la intensidad de cada brote ya sí que habrá que usar algún método para obtener la relación entre ambos.

Sin embargo, no seríamos capaces de obtener la SFH a mano en un caso realista. Hay una cantidad de error que no se está teniendo en cuenta en este trabajo, pues estamos trabajando en todo momento con CMDs sintéticos. Si tuviéramos un CMD de una galaxia real, el método de mallas estáticas de convolución sería imprescindible de usar y prácticamente imposible de realizar a mano con precisión.

Referencias

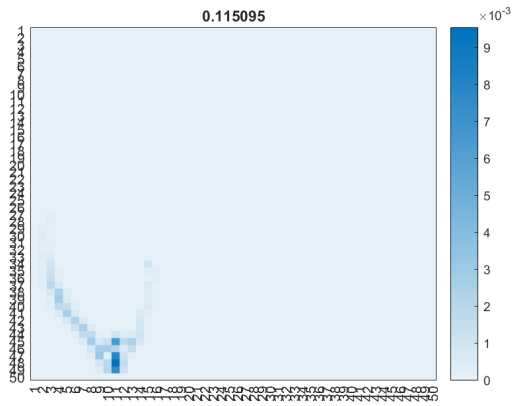
- [1] Antonio Aparicio y Carme Gallart *IAC-STAR: A Code for synthetic Color-Magnitude Diagram Computation*, The Astronomical Journal, 128:1465–1477, 2004 September
- [2] Antonio Aparicio y Sebastian Hidalgo *IAC-POP: Finding the Star Formation History of Resolved Galaxies*, The Astronomical Journal, 138:558–567, 2009 August
- [3] Springel, V., White, S. D. M., Jenkins, A., et al., *Simulations of the formation, evolution and clustering of galaxies and quasars*, 2005. Nature, 435 (Nature Publishing Group), 629.
- [4] Wikipedia, *Main sequence*, https://en.wikipedia.org/wiki/Main_sequence
- [5] Wikipedia, *Locality of reference*, https://en.wikipedia.org/wiki/Locality_of_reference
- [6] Roca, S. *Apuntes de clase*, 2018. Formación y evolución de galaxias, Universidad Complutense de Madrid.

A. Contenido Complementario

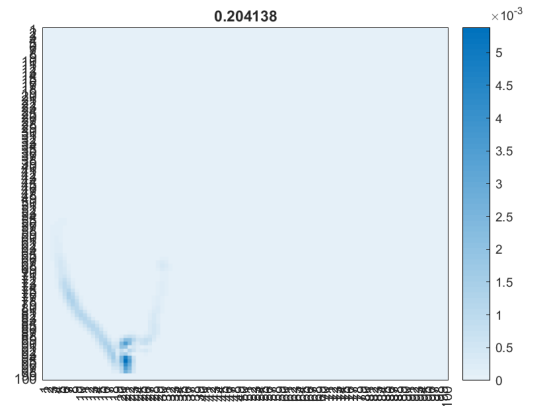
Material adicional (como los datos de las queries, los códigos o más figuras) se puede encontrar en el repositorio: <https://github.com/enriquegalceran/Formacion-Practica2>

B. Imágenes Adicionales

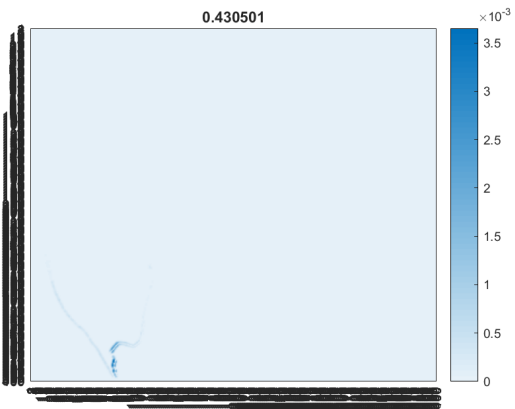
Figura 14 – Distribuciones de densidad para la SFH con ID:13198



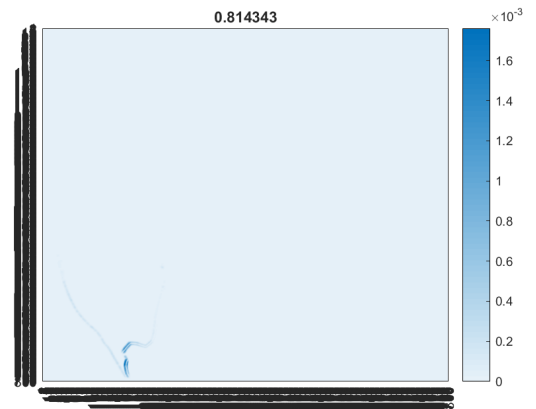
(a) resolución = 50



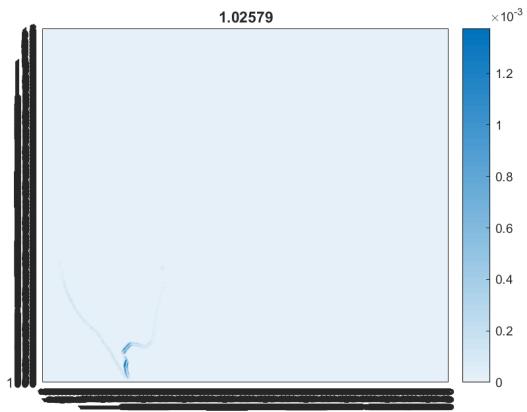
(b) resolución = 100



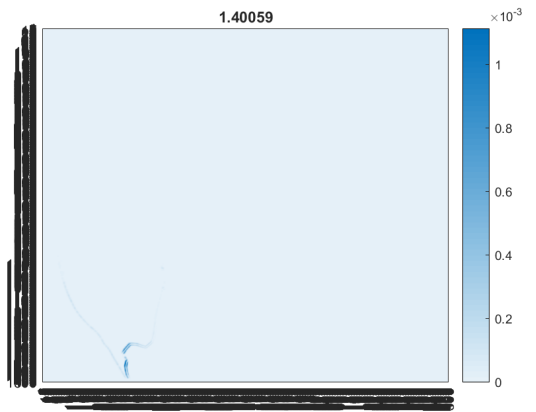
(c) resolución = 400



(d) resolución = 800

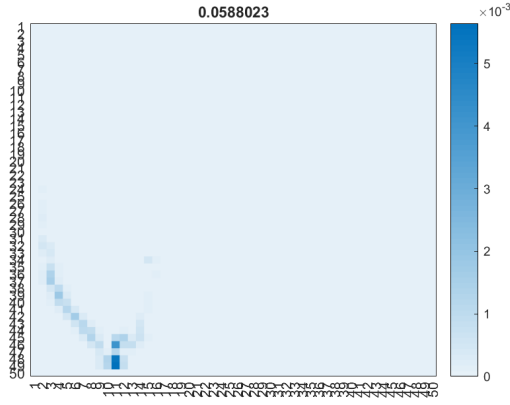


(e) resolución = 1000

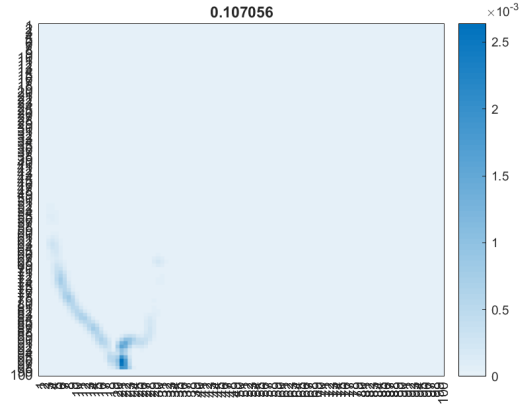


(f) resolución = 1500

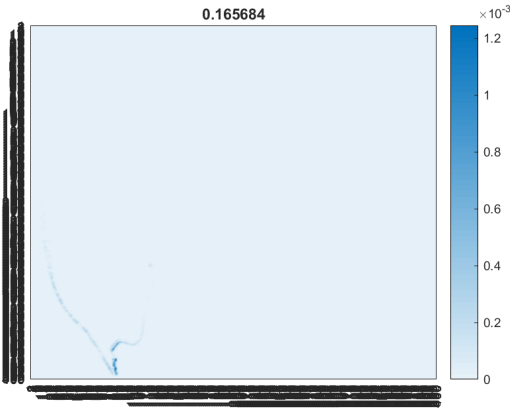
Figura 15 – Distribuciones de densidad para la SFH con ID:13644



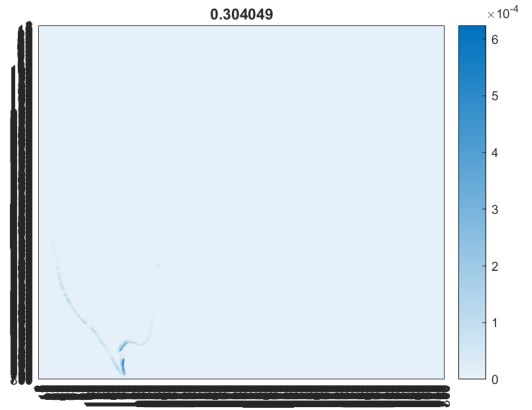
(a) resolución = 50



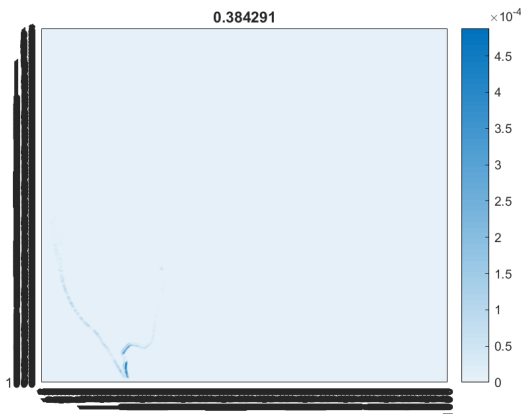
(b) resolución = 100



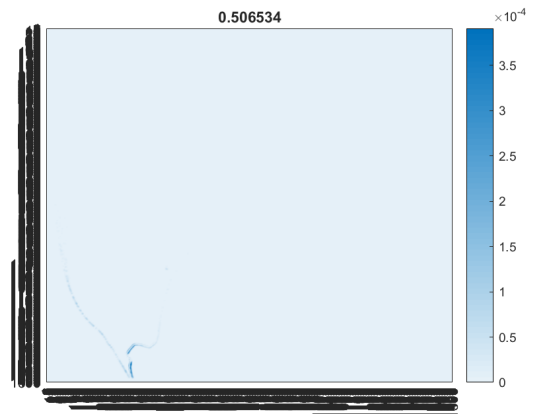
(c) resolución = 400



(d) resolución = 800



(e) resolución = 1000



(f) resolución = 1500