



FACULTAD
DE
CIENCIAS

**Selección de fuentes extragalácticas a alto redshift
mediante una herramienta estadística de
cross-identificación de catálogos de galaxias.**

(Selection of extragalactic sources at high redshift using a statistical tool for
cross-identification of galaxy catalogs.)

Trabajo de Fin de Grado
para acceder al

GRADO EN FÍSICA

Autor: Javier Gutiérrez Solórzano

Director: Diego Herranz Muñoz

Febrero, 2018

Agradecimientos

Quiero agradecer a mi director de proyecto, Diego Herranz su dedicación y disponibilidad para solucionar muchos de los problemas que han ido surgiendo durante el largo tiempo que ha supuesto la elaboración de este trabajo. Extiendo mi agradecimiento a Joaquín González-Nuevo porque su ayuda ha sido imprescindible para implementar correctamente el método para la estimación del desplazamiento al rojo de las galaxias submilimétricas.

También quiero mostrar mi especial gratitud a mis padres por todo el tiempo que ellos invierten para que yo pueda hacer esto y muchas otras cosas más.

Índice

Resumen	4
1 Introducción	8
1.1 Galaxias con alto desplazamiento al rojo.	9
1.1.1 Características de las galaxias con alto desplazamiento al rojo.	9
1.2 Desplazamiento al rojo.	10
1.2.1 Desplazamiento al rojo cosmológico.	10
1.3 Técnicas para determinar el desplazamiento al rojo.	14
1.3.1 <i>Redshift</i> espectroscópico.	14
1.3.2 <i>Redshift</i> fotométrico.	14
1.4 Lentes gravitatorias	15
1.4.1 Radio de Einstein.	16
1.5 Inferencia bayesiana.	18
1.5.1 Teorema de Bayes y teorema de las probabilidades totales.	18
1.5.2 El factor de Bayes	20
2 Selección de las muestras	22
2.1 Proyecto <i>H</i> -ATLAS	22
2.1.1 Observatorio espacial <i>Herschel</i>	22
2.1.2 Catálogo <i>H</i> -ATLAS DR1.	23
2.2 Proyecto GAMA	26
2.2.1 Catálogo GAMA I	27
2.3 Región de solapamiento de ambos catálogos	29
3 <i>Redshift</i> fotométrico de ETGs a partir de SPIRE	31
3.1 Selección de la SED de referencia	31
3.1.1 La galaxia SMM J2135-0102.	33
3.2 Algoritmo	34
3.3 Confrontación del método	37
4 Método probabilístico de Cross-Identificación	39
4.1 Factor de Bayes posicional propuesto por Budavári & Szalay	41
4.2 Factor de Bayes fotométrico propuesto por Budavári	42
5 Identificación sistemas candidatos a lente gravitatoria	44
5.1 Criterio propuesto por Negrello et al.	44
5.2 Criterio propuesto por González-Nuevo et al.	44
5.3 Propuesta para encontrar SLGs	44
6 Resultados y análisis	47
6.1 Candidatos según Negrello et al. y González-Nuevo et al.	47
6.2 Candidatos según el criterio bayesiano	47
7 Discusión y conclusiones	49
A Redshifts fotométricos de los candidatos	51
B Selección de objetos del catálogo GAMA	52

C Código en Python	53
C.1 Esquema del módulo <code>main_halos.py</code>	54
C.2 Módulo <code>main_halos.py</code>	55
C.3 Módulo <code>main_gama.py</code>	61
C.4 Módulo <code>main_hatlas.py</code>	63
C.5 Módulo <code>main_matching_gh.py</code>	67
C.6 Módulo <code>main_matching_gh_simulacion.py</code>	82
C.7 Módulo <code>factor_bayes.py</code>	86
C.8 Módulo <code>rojo.py</code>	88
C.9 Módulo <code>get.py</code>	93
C.10 Módulo <code>main_smm.py</code>	108
C.11 Módulo <code>main_main_radio_Einstein_D_L.py</code>	110
C.12 Módulo <code>main_ajuste_error_espec.py</code>	111
Bibliografía	114

Resumen

Se ha propuesto un método para la búsqueda de galaxias en fases tempranas de su formación suponiendo que la imagen observada ha sido magnificada por el efecto de una lente gravitatoria. El método consiste en seleccionar un subconjunto de emparejados que surgen al aplicar un criterio estadístico de cross-identificación basado en el factor de Bayes a los catálogos *H*-ATLAS DR1 y GAMA DR1. El factor de Bayes utilizado consta de un término basado en la separación angular entre dos observaciones y otro basado en el corrimiento al rojo. En los casos en los que se cuenta con una medida fiable del corrimiento al rojo de la fuente perteneciente a *H*-ATLAS, se utilizarán esos valores. En caso de que no se disponga, se obtendrá su *redshift* fotométrico a partir de un ajuste por mínimos cuadrados de la distribución espectral de energía de una galaxia modelo a tres medias del flujo espectral de la fuente a tres longitudes de onda concretas. Si el factor de Bayes da una alta probabilidad de que el par de observaciones no pertenecen a un mismo objeto, pero el término que depende de la distancia angular sugiere que si lo son, entonces tenemos un buen candidato para ser una lente gravitatoria.

El método desarrollado para estimar el corrimiento al rojo ha dado buenos resultados al compararlo con trabajos anteriores. Por su parte los resultados obtenidos mediante el método bayesiano para la identificación de lentes muestra una pureza relativamente baja pero resulta prometedor como fase previa a un estudio más detallado porque permite identificar de una forma rápida, sencilla y directa, basada en un criterio estadístico riguroso, entorno al tres por ciento de los candidatos con mayor probabilidad de ser sistemas lente de entre las galaxias del catálogo *H*-ATLAS DR1.

Palabras clave: Alto desplazamiento al rojo – galaxias submilimétricas – lente gravitatoria fuerte – factor de Bayes – cross-identificación

Abstract

A method has been proposed for the search of galaxies in early phases of their formation supposing that the observed image has been magnified by the effect of a gravitational lens. The method consists of selecting a subset of paired that arise by applying a statistical criterion of cross-identification based on the Bayes factor to the catalogs *H*-ATLAS DR1 and GAMA DR1. The Bayes factor used consists of a term based on the angular separation between two observations and another based on the redshift. In those cases in which a reliable measure of the redshift of the source belonging to *H*-ATLAS is available, these values will be used. If it is not available, its photometric redshift will be obtained from a least squares fit of the spectral energy distribution of a model galaxy to three measures of the spectral flux of the source at three specific wavelengths. If the Bayes factor gives a high probability that the pair of observations do not belong to the same object, but the term that depends on the angular distance suggests that if they are, then we have a good candidate to be a gravitational lens.

The method developed to estimate the redshift has given good results when compared to previous works. On the other hand, the results obtained by the Bayesian method for the identification of lenses show a relatively low purity but it is promising as a previous phase to a more detailed study because it allows to identify in a fast, simple and direct way, based on a rigorous statistical criterion, around the three percent of the candidates most likely to be lens systems among the galaxies in the *H*-ATLAS DR1 catalog.

Key words: high-redshift – submillimeter galaxies – strong gravitational lensing – Bayes factors – cross-identification

1 Introducción

Los objetos con alto desplazamiento al rojo o alto *redshift*¹ en inglés, (en lo sucesivo utilizaremos el término *redshift* por brevedad) son muy importantes para entender cómo se formaron las estructuras que observamos en el universo actual. En su gran mayoría se trata de galaxias en estadios tempranos de su formación que se caracterizan por una magnitud aparente muy débil y una emisión electromagnética principalmente en la zona infrarroja (IR) del espectro. Debido a que la generación actual de telescopios infrarrojos (en especial aquellos que operan en el infrarrojo lejano, $\lambda \sim 100 - 500 \mu\text{m}$) tienen una resolución espacial bastante limitada en comparación con los telescopios ópticos (la resolución angular de un telescopio con apertura circular es proporcional al inverso de su diámetro y los telescopios que operan en el IR-lejano deben ser telescopios espaciales pues la atmósfera es opaca en esa zona del espectro) las medidas realizadas sobre las galaxias tempranas en formación² (ETGs) tienen una gran contaminación debido a la presencia de otras fuentes débiles en sus proximidades y son especialmente difíciles de estudiar.

A veces, la luz procedente de los objetos lejanos resulta desviada debido a la presencia de objetos muy masivos que se interponen entre estas fuentes y nosotros (generalmente galaxias elípticas y grupos de galaxias) y se producen intensas magnificaciones (entendiéndose como un incremento sobre el brillo original) de la imagen del objeto fuente. Este fenómeno, conocido como lente gravitatoria, lejos de ser un problema está resultando ser una importante herramienta para el estudio de las propiedades individuales y estadísticas de las galaxias submilimétricas (SMGs). Eso ha motivado la aparición de distintas propuestas para la identificación de galaxias lejanas lensadas³ (SLGs) entre los cartografiados de galaxias actuales. En particular, los trabajos de [Negrello et al. \(2010\)](#) y [González-Nuevo et al. \(2012\)](#) proponen unos criterios para seleccionar este tipo de galaxias a partir de las medidas del telescopio espacial infrarrojo *Herschel*. Ambos criterios proponen la identificación de los candidatos a partir de la selección de distintos flujos de corte para un conjunto de frecuencias consideradas. Estos métodos, como cualquier otro, tienen sus propios sesgos de selección, por lo que resultaría interesante disponer de otros métodos radicalmente diferentes para seleccionar SLGs.

En este trabajo planteamos la búsqueda de SLGs identificando sistemas lente gravitatoria completos, es decir, encontrando no solo el objeto fuente que ha sido lensado, sino también el objeto lente cuya masa es responsable de la desviación de la luz procedente de la fuente. Al igual que los autores anteriores, nosotros proponemos la búsqueda de SLGs en el catálogo *H-ATLAS* porque es sensible a las longitudes de onda en el IR-lejano en el que se produce el máximo de emisión de las ETGs y porque se ha estimado que la sección eficaz de detección de lentes a alto *redshift* por el instrumento SPIRE es muy favorable ([Negrello et al., 2010](#)). En este caso, debido a que existe una zona de solapamiento entre los cartografiados *H-ATLAS* y GAMA y también debido a que la distribución en *redshift* de los objetos del catálogo GAMA es la adecuada, consideramos el catálogo GAMA como una buena elección para encontrar en él los objetos lente. Nuestra idea consiste en aplicar un criterio estadístico de cross-identificación de catálogos de galaxias para identificar lentes gravitatorias en las que participa como objeto fuente una ETG. El método consta de dos factores de Bayes propuestos por [Budavári and Szalay \(2008\)](#) y [Budavári \(2011\)](#);

¹A lo largo del trabajo haremos referencia a objetos con un *redshift* alto, medio y bajo. La distinción típica que se encuentra en la literatura es: alto *redshift* $z > 1$, *redshift* intermedio $0.1 < z < 1$ y *redshift* bajo $z < 0.1$

²También nos referiremos a estos objetos como “galaxias lejanas”, “galaxias submilimétricas” (en referencia al rango del espectro en el que son mas brillantes), o “galaxias tempranas”. Debe quedar claro que en este trabajo cuando utilizamos el término “galaxia temprana” nos referimos siempre a un orden cronológico y no a la morfología de la galaxia.

³En las publicaciones realizadas en lengua inglesa, se refieren a estos objetos con el nombre de “*gravitationally lensed galaxies*”, sin embargo resulta difícil traducir correctamente esta terminología al castellano. A falta de una traducción mejor usaremos la palabra “lensar” en lugar del término *lensed*.

uno de ellos depende de la separación angular entre las observaciones⁴ y otro de la diferencia entre los *redshift*. Cada emparejado considerado está formado por dos observaciones, una de ellas perteneciente al catálogo *H*-ATLAS y la otra a GAMA. En los casos en los que se cuenta con una medida fiable del corrimiento al rojo de la observación realizada por *H*-ATLAS se utilizarán esos valores; en caso de que no se disponga de dicha información, se realizará un ajuste por mínimos cuadrados de la distribución espectral de energía (SED) de una galaxia modelo a tres medidas del flujo espectral en tres longitudes de onda diferentes realizadas por el instrumento SPIRE. Si el ajuste se considera adecuado proporcionará el *redshift* fotométrico de la fuente. En el caso de que el factor de Bayes conjunto dé una alta probabilidad de que el par de observaciones no pertenezcan a un mismo objeto, pero el término que depende de la distancia angular sugiere que si lo son, el par de observaciones es considerado un sistema lente gravitatoria.

La memoria está dividida en varias secciones: la Sección 1 contiene una explicación básica de algunos de los conceptos más importantes que serán usados a lo largo del trabajo; la Sección 2 pretende dar una explicación general de los catálogos utilizados y de los instrumentos con los que se realizaron; la Sección 3 describe el método utilizado en este trabajo para calcular el *redshift* fotométrico de las ETGs a partir de las medidas del instrumento SPIRE, se explicarán también los fundamentos en los que se basa y confrontaremos el *redshift* obtenido para un conjunto de galaxias con los obtenidos por otros autores; en la Sección 4 describiremos en detalle el método de cross-identificación utilizado; en la Sección 5 se describirán las distintas propuestas para identificar los sistemas que conforman lentes gravitatorias, incluida la nuestra; en la Sección 6 se mostrarán los resultados obtenidos y finalmente, Sección 7, expondremos nuestras conclusiones. Los apéndices contienen tablas y el código de los programas escritos en lenguaje de programación *Python*.

1.1 Galaxias con alto desplazamiento al rojo.

Las galaxias con alto desplazamiento al rojo son galaxias muy lejanas, cuya luz ha tardado miles de millones de años en recorrer la distancia que nos separa de ellas y que ahora, vistas desde la Tierra, muestran el aspecto que tuvieron cuando la luz partió de ellas. Estos objetos juegan un papel fundamental en cosmología, porque su estudio permite comprender cómo se formaron las galaxias actuales. Una de las predicciones fundamentales de la teoría del *Big Bang* es que el universo temprano consistía en materia y radiación distribuida de forma homogénea y en equilibrio termodinámico. Hoy día el universo es altamente heterogéneo, formando estructuras como estrellas, galaxias, cúmulos y supercúmulos de galaxias. Entender cómo se produjo esta etapa de transición entre el universo homogéneo a la formación de las estructuras que se observan en el universo actual, es uno de los objetivos de la cosmología y de la física fundamental.

1.1.1 Características de las galaxias con alto desplazamiento al rojo.

Se especula que las primeras galaxias, se formaron a partir de nubes de gas compuestas por hidrógeno y pequeñas cantidades de helio y litio. En estas condiciones se formaron un gran número de estrellas, en un periodo de tiempo relativamente corto, conocido como *starburst* (estallido estelar). Para una galaxia del tamaño de la Vía Láctea con $\sim 10^{11}$ estrellas, se cree que el proceso duró $\sim 10^8$ yr, lo que implica una tasa de formación estelar (SFR) $\sim 10^3 M_{\odot} \text{ yr}^{-1}$, una cifra enorme si se compara con la tasa de formación estelar actual de la misma $\sim 1 M_{\odot} \text{ yr}^{-1}$ (Murdin, 2001). Estas primeras estrellas, forman la primera generación de estrellas en el Universo, llamada Población III y tenían unas características distintas a las actuales. Se cree que eran muy masivas, llegando a poseer masas de hasta $100 M_{\odot}$. Actualmente, las nebulosas donde se forman las nuevas

⁴En este trabajo se entiende por observación al conjunto de dos medidas: posición y desplazamiento al rojo. Cada catálogo realiza sus propias observaciones de forma independiente.

estrellas contienen metales⁵, lo cual incrementa su opacidad; la radiación emitida durante el colapso de la nebulosa interacciona con el material y parte de él no pasa a formar la estrella, porque que es expulsado, lo cuál limita la masa de las estrellas actuales. Las estrellas muy masivas quemar muy rápidamente su combustible nuclear (los modelos indican que una estrella como el Sol está en la secuencia principal $\sim 10^{10}$ años, mientras que las estrellas masivas con $> 25 M_{\odot}$ tienen una vida media de menos de 7×10^6 años, [Murdin 2001](#)), y finalizan su vida como supernovas que expulsan grandes cantidades de residuos nucleares que enriquecen el medio interestelar con elementos pesados.

Las estrellas pertenecientes a estas primera etapas del Universo debieron ser grandes emisoras de luz y radiación ultravioleta (UV). Las grandes nubes de gas y polvo que se encontraban en las galaxias que las albergaban absorbían una parte esta radiación y la remitían en el IR. Las temperaturas de equilibrio térmico para esas nubes se encuentran el rango de $10 - 100$ K, lo que da lugar la emisión IR en la región $\lambda \sim 30 - 300 \mu\text{m}$ ([Murdin, 2001](#)). Los halos galácticos de las galaxias espirales actuales de tipo *Sa* están compuestos por poblaciones de estrellas viejas, con edades $\gtrsim 8 - 9$ Gyr (lo que corresponde a $z \gtrsim 1 - 1.5$, [Lapi et al. 2011](#)), las cuales se formaron a partir de los residuos nucleares de las estrellas que las precedieron y conforman la llamada Población II; por tanto, las ETGs muestran desplazamientos al rojo $z \gtrsim 1$. La radiación IR procedente de una nube de polvo y gas interestelar perteneciente a una galaxia con un desplazamiento al rojo $z = 1$ debería encontrarse en la zona del espectro con $\lambda \sim 60 - 600 \mu\text{m}$ (estas cifras se obtienen fácilmente a partir de la Ecuación 2 y el rango de emisión de las nubes al que se ha hecho referencia anteriormente). Estos datos suponen un punto de partida para la búsqueda de ETGs a partir de su observación en la zona IR del espectro.

1.2 Desplazamiento al rojo.

El desplazamiento al rojo (o desplazamiento al azul) es el desplazamiento del espectro electromagnético de un objeto hacia longitudes de onda mayores (o más cortas) que puede deberse a movimientos cinemáticos entre el emisor y el receptor (z_{mov} , efecto Doppler), a efectos gravitatorios (z_{grav}) o a la expansión de la métrica del universo a grandes escalas (z_H). Dado un desplazamiento al rojo z , podemos descomponerlo como la suma

$$z = z_{mov} + z_{grav} + z_H, \quad (1)$$

de las distintas contribuciones citadas anteriormente. La definición general de corrimiento al rojo, siempre válida, independientemente de la causa que lo originó, es

$$z = \frac{\lambda_o - \lambda_e}{\lambda_e} = \frac{\lambda_o}{\lambda_e} - 1, \quad (2)$$

donde los subíndices hacen referencia a la longitud de onda emitida e y observada o .

1.2.1 Desplazamiento al rojo cosmológico.

Cuando se habla de desplazamiento al rojo en el contexto de la cosmología, nos estamos refiriendo al hecho de que la distribución espectral de energía de las galaxias lejanas se encuentra desplazada hacia longitudes de onda mayores. Este hecho fue observado ya en el siglo XIX por aquellos que comenzaron a identificar líneas espectrales en los espectros emisión de los cuerpos celestes que en

⁵ Se entiende por metales todos los núcleos atómicos con masa atómica superior al He. Estos elementos no se encontraban en el universo antes de la aparición de las primera estrellas.

aquel momento se conocían como nebulosas⁶. Entre 1912 y 1925 el astrónomo italiano Vesto Slipher midió la posición de las líneas de emisión de unas 40 de estas nebulosas y constató por primer vez y para su sorpresa que, en todas ellas, las líneas se encontraban desplazadas hacia longitudes de onda inferiores con respecto a las mismas líneas espectrales medidas en el laboratorio (Cepa, 2007). Posteriormente Edwin Hubble y su ayudante Milton Humanson, que disponían del mejor telescopio construido hasta entonces en el monte Palomar, Arizona, dieron un paso más en la comprensión del fenómeno. En primer lugar extendieron la lista elaborada por Slipher y probaron que en ningún caso el espectro se encontraba desplazado hacia el azul. En segundo lugar, Hubble identificó estrellas variables cefeidas en varios de estos cuerpos nebulosos⁷ y gracias a ello pudo medir por primera vez la distancia a esos objetos. Interpretó el desplazamiento al rojo como un efecto Doppler y halló su velocidad a partir de la expresión

$$v \approx cz, \quad (3)$$

e hizo una representación gráfica de la velocidad asociada a ese desplazamiento al rojo frente a la distancia que el midió. Se dio cuenta de que había una relación lineal entre la velocidad que y la distancia,

$$v = H_0 D \implies z = (H_0/c)D. \quad (4)$$

Esta última igualdad es conocida como la ley de Hubble y H_0 como constante de Hubble.

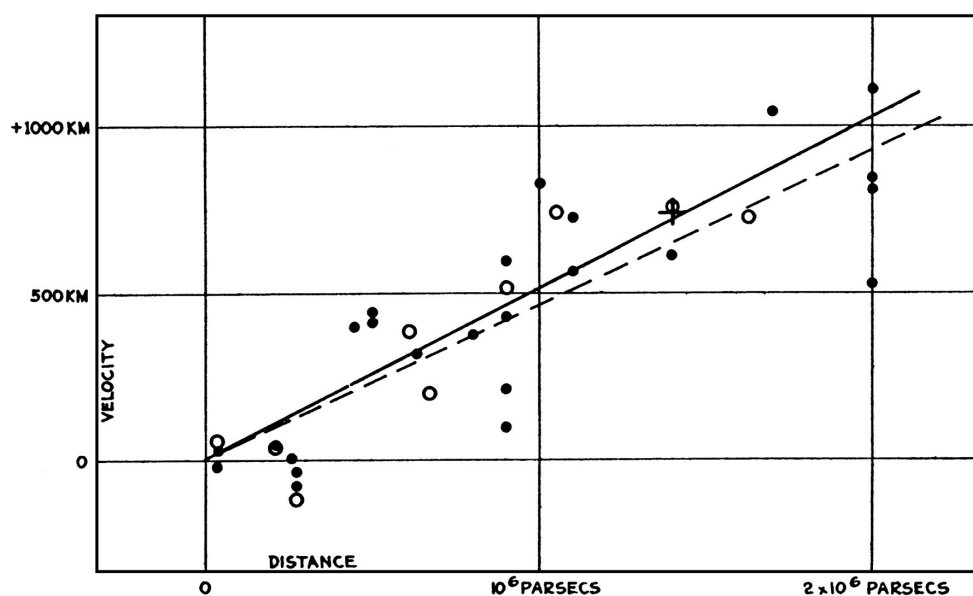


Figura 1: Gráfica de una de las publicaciones originales de Hubble. Pone en evidencia que Hubble interpretó el desplazamiento al rojo como debido al efecto Doppler. “Una Relación entre la Distancia y Velocidad Radial en las Nebulosas Extra-Galácticas”, Los Procesos de la Academia Nacional de Las Ciencias, Volumen 15, Edición 13, 1929: p. 172. ©Huntington Library, San Marino, CA. Fuente: [https://www.visionlearning.com/es/library/Proceso-de-la-Ciencia/49/La-Naturaleza-del-Conocimiento-Cient% c3%adfico/185](https://www.visionlearning.com/es/library/Proceso-de-la-Ciencia/49/La-Naturaleza-del-Conocimiento-Cient%c3%adfico/185)

⁶En aquel momento se conocían como nebulosas todos aquellos cuerpos celestes cuya luz no parecía provenir de una fuente puntual y su apariencia era similar a una nube de luz difusa. No estaba todavía establecida la diferencia entre nebulosas y galaxias.

⁷Fueron precisamente estudios como los que llevó a cabo Hubble para determinar las distancias, los que pusieron en evidencia que las galaxias y las nebulosas son objetos muy diferentes.

Es conveniente aclarar dos puntos. En primer lugar la ley de Hubble no relaciona en realidad velocidades con distancias, sino desplazamientos al rojo con distancias. Esta confusión es debida a que inicialmente el desplazamiento al rojo se achacó a un efecto Doppler tal y como se deduce de las publicaciones originales de Hubble (Figura 1). En segundo lugar, la ley de Hubble no es un modelo físico, es un hecho observable. Actualmente se ha medido del orden de 10^6 desplazamientos y solo ~ 10 de ellos muestran desplazamientos hacia el azul, siendo estos últimos, todos ellos, objetos del Grupo Local. Por tanto cualquier interpretación de este fenómeno debe ser coherente con los siguientes puntos:

- El desplazamiento siempre se produce hacia longitudes de onda mayores, a excepción de los objetos que se encuentran en el Grupo Local.
- El desplazamiento no depende de la frecuencia, afecta a todo el espectro electromagnético de la misma forma.
- El desplazamiento es lineal con la distancia al objeto, al menos para el caso de los objetos más próximos, cuya distancia se ha podido determinar por métodos directos.
- Los desplazamientos son isótropos, es decir, no dependen de la dirección en la que se encuentren en la bóveda celeste, únicamente de la distancia que nos separa de los mismos.

Además se impone que todas las condiciones anteriores deben cumplirse en cualquier lugar del Universo. Es obvio que no hay evidencia experimental de que esto se cumpla; se trata de una suposición basada en lo que se conoce como principio anti-anropocéntrico, es decir, el observador no se encuentra en un lugar privilegiado. Actualmente, la explicación más ampliamente aceptada es la conocida como evolución de la métrica, por ser prácticamente la única explicación disponible. Esta explicación acepta que existe un factor de escala que cambia con el tiempo cosmológico, aumentando la distancia radial que nos separa de los objetos. Dado que la velocidad de la luz es finita y las distancias que nos separan incluso de las galaxias más próximas es enorme, los intervalos de tiempo que tarda la luz en alcanzarnos son lo suficientemente grandes como para que el factor de escala haya cambiado sensiblemente afectando a la longitud de onda de la radiación electromagnética mientras se transmite por el espacio⁸. Los desplazamientos hacia el azul que se observan entre los objetos del Grupo Local se deben a que tienen un movimiento cinemático dirigido hacia nosotros y a que se encuentran relativamente próximos. En este caso, la contribución a z debida al efecto Doppler resulta más importante que la debida a la expansión de la métrica, que siempre contribuye a un desplazamiento al rojo. Para objetos muy lejanos la expansión de la métrica resulta dominante siempre.

A continuación se muestra cómo afecta el cambio del factor de escala a las propiedades de la luz que viaja por el espacio, partiendo de la métrica utilizada en el modelo cosmológico estándar (Cepa, 2007). En un universo en expansión, el *redshift* está directamente relacionado con la distancia. En el modelo cosmológico estándar de Friedmann-Lemaître, las distancias están definidas a partir de la métrica de Robertson-Walker⁹

$$ds^2 = c^2 dt^2 + R^2(t) \left(\frac{dr^2}{1 - \kappa r^2} + r^2 d\theta^2 + r^2 \sin^2 \theta d\varphi^2 \right), \quad (5)$$

⁸Es muy interesante observar que el factor de escala solo afecta al universo a grandes escalas como la distancia entre galaxias. Si el factor de escala afectase a escala de laboratorio de la misma forma que a las escalas astronómicas, cambiaría también nuestros patrones de medida incrementándose en la misma proporción y haciendo indetectable la expansión del universo.

⁹Se demuestra en Weinberg (1972) que es la métrica más general que describe un espacio de tetra-dimensional, con tres dimensiones espaciales y una temporal y cuyo subespacio espacial tiene curvatura arbitraria y es maximalmente simétrico.

siendo $R(t)$ el factor de escala, κ el signo de la curvatura que puede tomar los valores -1, 0 o 1, t el tiempo cosmológico y r, θ y φ las coordenadas comóviles expresadas en un sistema de coordenadas esféricas. Debido a la isotropía del espacio, la distancia propia solo depende de la coordenada radial comóvil r y es constante con θ y φ , con lo cual, la métrica se simplifica reduciéndose a

$$ds^2 = c^2 dt^2 + R^2(t) \frac{dr^2}{1 - \kappa r^2}.$$

Supongamos que se emiten dos pulsos de luz. El primero se emite desde un punto coordenada radial comóvil $r = r_e$ y tiempo cosmológico $t = t_e$. El segundo con las mismas coordenadas espaciales y un tiempo cosmológico $t_e + T_e$. Estos pulsos se propagan por el espacio durante un tiempo suficientemente grande, como el tiempo necesario para cruzar el espacio entre dos galaxias próximas, tras el cual alcanzan a un observador. El primer pulso lo alcanza en coordenadas $r = 0$ y $t = t_o$, mientras que el segundo alcanza al observador en $r = 0$ y $t = t_o + T_o$. Teniendo en cuenta que las ondas luminosas viajan por geodésicas ($ds = 0$) y dado que los dos pulsos fueron emitidos y finalmente recibidos en puntos con las mismas coordenadas espaciales, tenemos que:

$$\int_{t_e}^{t_o} \frac{dt}{R(t)} = \int_{t_e+T_e}^{t_o+T_o} \frac{dt}{R(t)} = -c^{-1} \int_{r_e}^0 \frac{dr}{\sqrt{1 - \kappa r^2}}.$$

Centraremos la atención en la igualdad establecida por las integrales en las que participa el factor de escala. En principio no podemos realizar las integrales porque desconocemos la función $R(t)$; sin embargo podemos realizar una manipulación sobre los limites de integración:

$$\begin{aligned} \int_{t_e+T_e}^{t_o+T_o} \frac{dt}{R(t)} &= \int_{t_e}^{t_o} \frac{dt}{R(t)} \equiv \int_{t_e}^{t_e+T_e} \frac{dt}{R(t)} + \int_{t_e+T_e}^{t_o} \frac{dt}{R(t)} \implies \\ \int_{t_e}^{t_e+T_e} \frac{dt}{R(t)} &= \int_{t_o}^{t_o+T_o} \frac{dt}{R(t)} + \int_{t_e+T_e}^{t_o+T_o} \frac{dt}{R(t)} \equiv \int_{t_o}^{t_o+T_o} \frac{dt}{R(t)}. \end{aligned}$$

De esta forma, si los tiempos de integración T_e y T_o han sido mucho más pequeños que el tiempo de propagación de los pulsos por el espacio, el factor de escala puede considerarse constante, teniendo distintos valores cuando los pulsos fueron emitidos y cuando fueron recibidos:

$$\frac{1}{R(t_e)} \int_{t_e}^{t_e+T_e} dt = \frac{1}{R(t_o)} \int_{t_o}^{t_o+T_o} dt \implies \frac{T_e}{R(t_e)} = \frac{T_o}{R(t_o)}.$$

Por comodidad podemos considerar que T_e y T_o son los periodos de la onda luminosa cuando fue emitida y cuando fue observada, respectivamente. Utilizando la ecuación $T = \lambda/c$ y la definición general de corrimiento al rojo (Ec. 2) obtenemos que

$$\frac{R(t_o)}{R(t_e)} = \frac{\lambda_o}{\lambda_e} = 1 + z. \quad (6)$$

El desplazamiento al rojo no se debe por tanto a un efecto Doppler, si no porque el incremento del factor de escala, afecta de la misma forma a la longitud de onda de la radiación electromagnética durante el tiempo de propagación que a las distancias propias.

Desarrollando en serie de Taylor el inverso del factor de escala en torno a $t = t_o$,

$$z = \frac{R(t_o)}{R(t_e)} - 1 = \frac{\dot{R}(t_o)}{R(t_o)}(t_o - t) + \left[\left(\frac{\dot{R}(t_o)}{R(t_o)} \right)^2 - \frac{1}{2} \frac{\ddot{R}(t_o)}{R(t_o)} \right] (t_o - t)^2 + \dots$$

podemos recuperar la ley de Hubble al identificar el término $\dot{R}(t_o)/R(t_o)$ con el valor de la constante de Hubble en el tiempo cosmológico actual H_o y truncándolo en el término de segundo orden

$$z = H_o(t_o - t) + H_o^2 \left[1 + \frac{1}{2}q_o \right] (t_o - t)^2 + \dots$$

La ley de Hubble en el contexto del modelo estándar debe considerarse como una aproximación a la Ecuación 6 (Cepa, 2007).

1.3 Técnicas para determinar el desplazamiento al rojo.

En este trabajo estaremos haciendo referencia continuamente a “*redshift* espectroscópicos” y “*redshift* fotométricos”. Esos nombres hacen referencia a la técnica con la que se obtuvo esa medida. A continuación se explicará brevemente en que consisten estas técnicas. Cabe mencionar que el método propuesto en la Sección 3 de esta memoria forma parte de las técnicas fotométricas.

1.3.1 *Redshift* espectroscópico.

Cuando se habla de *redshift* espectroscópico nos estamos refiriendo a una técnica para determinar el *redshift* de los objetos celestes. El *redshift* espectroscópico consiste en identificar algunas de las líneas espectrales de absorción o emisión de algún elemento químico en la luz proveniente de un objeto celeste y comparar la longitud de onda a la que se encuentran con la que debería tener si proviniesen de una muestra en reposo. Este es el primer método que se utilizó para determinar el desplazamiento al rojo de los objetos celestes y permite determinar el *redshift* de forma muy precisa; el problema que tiene es que requiere de largos tiempos de exposición (~ 3 horas en un telescopio de 4 m para un objeto de magnitud aparente ~ 22) y aunque existen técnicas para obtener varios espectros simultáneamente, es imposible hacerlo de un número grande de objetos en un tiempo razonable (Serrano, 2010).

1.3.2 *Redshift* fotométrico.

Se trata de otra técnica para determinar el *redshift*. En este caso se utilizan varios filtros espectrales que permiten el paso de luz en una determinada región electromagnética muy estrecha. Después se utiliza una base de datos donde se almacena una colección de las SEDs de varios modelos de galaxias y se obtiene un espacio de parámetros (*redshift*, tipo espectral, magnitud, flujo...) que permite el mejor ajuste posible entre las medidas y los modelos. Estas técnicas se conocen con el nombre genérico de *SED-fitting procedure*.

Estos métodos surgen con la aparición de las cámaras de gran campo y los cartografiados (*surveys*) con los que es posible cubrir grandes áreas de cielo con tiempos de exposición más cortos que en el caso espectroscópico. Tienen la ventaja de ser mucho más rápidos que las técnicas espectroscópicas ya que como hemos dicho que obtener el espectro de cada objeto de un cartografiado es un trabajo muy lento. Como desventaja cabe señalar que la medida resultante tiene una mayor imprecisión que el método anterior. Además se hace necesario tener una base de datos lo suficientemente amplia (que represente a los objetos que se estén estudiando) con la que comparar y cobertura fotométrica amplia. Aún en ese caso, siempre puede haber objetos que tengan sus propias singularidades espectrales y por tanto sean difíciles de clasificar. En cuanto a las medidas es importante la cobertura espectral y la precisión fotométrica. Un factor limitante para la aplicación de este método suele ser que la cobertura espectral disponible para un determinado objeto suele ser pequeña, es decir, no se dispone de un número significativo de medidas sobre una región suficientemente amplia del espectro electromagnético; como es imposible disponer de un único

aparato que realice medidas en todo el espectro electromagnético, puede ser necesario disponer de varios instrumentos.

Los errores asociados a estas medidas dependen significativamente tanto del algoritmo utilizado, como del intervalo de z al que pertenece la medida. Por ese motivo, lo que se suele hacer es comparar las medidas espectroscópicas y fotométricas disponibles sobre una población suficientemente grande de objetos y se estima el error a partir de las dispersión de estas diferencias. Formalmente se calcula $\sigma = \sqrt{\langle (z_{\text{phot}} - z_{\text{spec}})^2 \rangle}$ (siendo z_{spec} el *redshift* espectroscópico y z_{phot} el fotométrico) para cada uno de los objetos; después se obtiene un valor promedio mediante un ajuste. La dispersión típica que se suele obtener mediante estos métodos es $\sigma \simeq 0.1 \times (1 + z)$ (Serrano, 2010). En el caso concreto de ANNZ,¹⁰ muestra una dispersión media cuadrática de $\sigma = 0.023$ al aplicarlo sobre los objetos del SDSS 1 (*Sloan Digital Sky Survey Data Release 1*), en el rango de $0 \lesssim z \lesssim 0.7$ (Collister and Lahav, 2004).

1.4 Lentes gravitatorias

Se denomina lente gravitatoria a los efectos que produce la gravedad sobre la luz, en particular la desviación de la trayectoria de luz procedente de objetos los objetos fuente debido a la presencia de objetos muy masivos llamados lente. El fenómeno que se produce varía dependiendo de la masa y forma del objeto lente y la posición relativa entre la fuente, la lente y el observador. Típicamente las lentes gravitatorias se clasifican en dos tipos, en base únicamente a los efectos cuantitativos que producen; las “lentes gravitatorias débiles” producen magnificaciones débiles (magnificación¹¹ $\mu < 2$) y distorsiones moderadas, mientras que las “lentes gravitatorias fuertes”, producen magnificaciones más intensas ($\mu \sim 2$ o más) y pueden producir imágenes múltiples muy distorsionadas de un mismo objeto. Al depender sus efectos de factores geométricos y gravedad, no vienen mezclados con otros fenómenos físicos, lo cual hace de las lentes un fenómeno particularmente interesante para estudiar, por ejemplo, distribuciones de materia oscura. También es reseñable que la distorsión actúa por igual sobre todo el espectro electromagnético, por lo cual las lentes gravitatorias carecen de aberración cromática.

El estudio de los efectos que produce la gravedad sobre la luz puede resultar una tarea muy complicada, por ello es común recurrir a una serie aproximaciones; generalmente se llevan a cabo las siguientes:

- Campos gravitatorios suficientemente débiles¹², en los que las partículas con masa siguen la dinámica newtoniana.
- La extensión de la masa de la lente a lo largo del eje de observación es despreciable en comparación con la distancia entre la fuente-lente y lente-observador (aproximación de lente delgada).
- La lente y la fuente se encuentran aproximadamente en la línea de observación, de forma que la separación angular entre ambos, $\theta \simeq \sin(\theta)$.
- Los efectos de difracción son despreciables, porque incluso la longitud de onda de las ondas de radio, es demasiado pequeña frente a las escalas de los elementos que forman la lente gravitatoria típica.

¹⁰ Se trata de un paquete de software, disponible de forma gratuita, para la estimación del *redshift* fotométrico utilizando redes neuronales artificiales (de ahí el nombre ANNZ, *Artificial Neural Networks*).

¹¹ Es suficiente interpretar este valor como el incremento del brillo de la fuente por efecto de la lente gravitatoria. La definición formal de la magnificación es complicada y no tiene cabida en este trabajo (para una explicación extensa se puede consultar Murdin 2001).

¹² La distinción entre lentes gravitatorias débiles y fuertes no depende de la intensidad del campo gravitatorio.

Consideremos un haz de luz en presencia de punto¹³ de masa M a una distancia R perpendicular a él. Teniendo en cuenta las aproximaciones anteriores, el efecto debido a la gravedad sobre los fotones es que estos se desvían un ángulo θ hacia la masa puntual,

$$\theta = \frac{4GM}{c^2 R} \quad (7)$$

donde G es la constante de gravitación universal y c la velocidad de la luz. Para campos gravitatorios débiles, la aproximación se aplica siempre que $R \gg \frac{2GM}{c^2} \equiv r_s$ (radio de Schwarzschild, r_s , que se corresponde con el radio aparente del horizonte de sucesos en un agujero negro). Para el caso de la masa típica de una estrella ($M \sim 1 M_\odot$), $r_s \sim 10$ km, para la masa típica de una galaxia ($M \sim 10^{11} M_\odot$) $r_s < 1$ pc, mientras que para un cúmulo de galaxias ($M \sim 10^{13} M_\odot$) $r_s < 10^3$ pc, por lo que en astronomía esta condición se cumple fácilmente (Murdin, 2001). La medida del ángulo de desviación de la luz procedente de una estrella debido a la masa del Sol es considerado como el primer test de la teoría de la Relatividad General. La Ecuación 7 predice el valor correcto; en contraste con la dinámica de Newton que predice la mitad de ese valor.

Las lentes gravitatorias debidas a estrellas que se encuentran en la Vía Láctea, se da en una de cada 10^6 estrellas, por lo que resultan un fenómeno raro si se compara con el número de lentes gravitatorias producidas por las galaxias y cúasares que se produce aproximadamente en uno de cada 10^3 (Murdin, 2001).

1.4.1 Radio de Einstein.

Se considerará ahora el caso particular en el que la fuente y la masa que hace de lente se encuentran exactamente sobre la línea de visión.

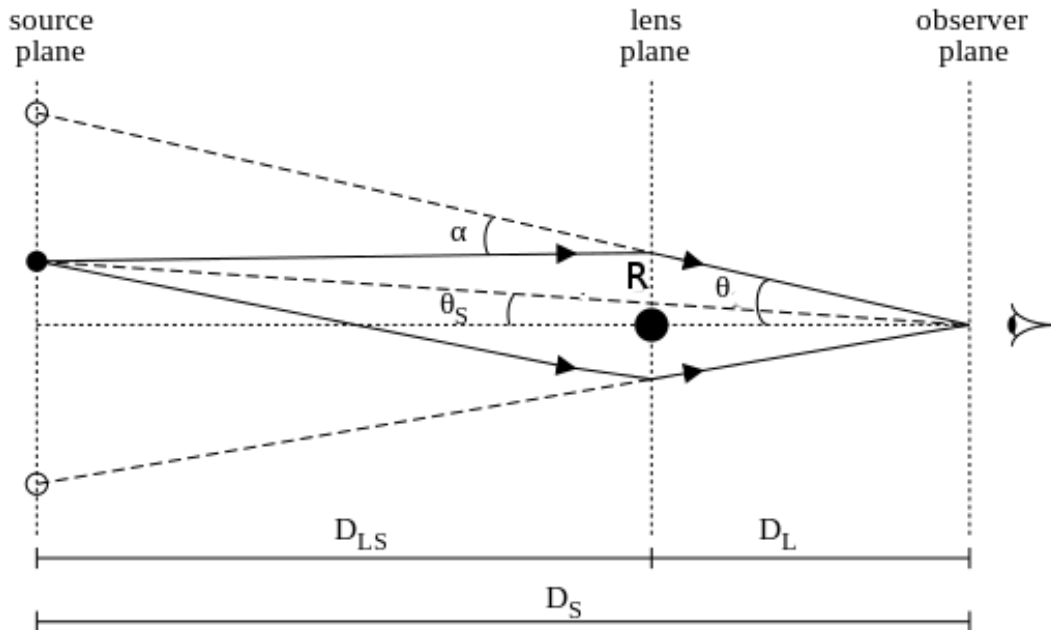


Figura 2: Distribución geométrica de los elementos que conforman una lente gravitatoria. (Fuente: https://en.wikipedia.org/wiki/Einstein_radius#/media/File:Gravitational_lens_geometry.svg)

¹³Si la masa que se interpone tiene una distribución de masa con simetría esférica, la luz fuera de la distribución es desviada como si se tratase de un objeto puntual (ley de Gauss).

Para realizar el estudio utilizaremos la expresión 7 junto con argumentos geométricos basados en la Figura 2. A partir del dibujo

$$D_S \theta = D_S \theta_S + D_{LS} \alpha,$$

que podemos reescribir despejando α :

$$\alpha = (\theta - \theta_S) \frac{D_S}{D_{LS}}. \quad (8)$$

La distancia entre el haz luminoso y la fuente puntual es R . Si R es suficientemente pequeño en comparación con D_L , se puede realizar la aproximación

$$\frac{R}{D_L} = \sin \theta \approx \theta \implies R \approx \theta D_L$$

y el ángulo de desviación de los fotones debido a la masa gravitatoria será:

$$\alpha = \frac{4GM}{c^2 \theta D_L}. \quad (9)$$

El último paso consiste en igualar la Ecuación 8 y la Ecuación 9:

$$\frac{4GM}{c^2 \theta D_L} = (\theta - \theta_S) \frac{D_S}{D_{LS}}$$

y notar que para el caso en que la masa se encuentra justo detrás de la lente, $\theta_S = 0$; el ángulo θ recibe entonces el nombre de radio de Einstein, que se denota mediante θ_E ,

$$\theta_E^2 = \frac{4GM}{c^2} \frac{D_{LS}}{D_L D_S}. \quad (10)$$

Este caso particular de lente gravitatoria da lugar al fenómeno conocido como anillos de Einstein, en la que la imagen de la fuente forma un círculo centrado en el objeto lente. Los anillos de Einstein han sido detectados en varias ocasiones, pero son un fenómeno muy difícil de observar ya que los requisitos necesarios para que tengan lugar son muy improbables. Sin embargo, el concepto es importante, porque los fenómenos asociados a lentes gravitatorias fuertes se producen a escalas de θ_E . Cuando $D_S \gg D_L$ entonces se cumple también $D_S \simeq D_{LS}$ y la Ecuación 10 se suele aproximar como

$$\theta_E \simeq 0.1 \times \left(\frac{M \text{ en } M_\odot}{D_L \text{ en pc}} \right)^{1/2} \text{ en arcsec}, \quad (11)$$

que es más adecuada para hacer cálculos teniendo en cuenta las unidades más utilizadas en astrofísica (Murdin, 2001).

1.5 Inferencia bayesiana.

La inferencia bayesiana es un método racional para la actualización de creencias. Se trata de un método de razonamiento aproximado, es decir, no se considera que una determinada información sea cierta o falsa con carácter absoluto; en su lugar, se parte de una hipótesis inicial a la que se le asigna un número como medida de la credibilidad que se tiene sobre la misma y con cada nueva observación se calcula de nuevo el factor numérico con el que se cuantifica el credibilidad de la hipótesis. La herramienta matemática que se utiliza para la actualización de creencias es el Teorema de Bayes.

1.5.1 Teorema de Bayes y teorema de las probabilidades totales.

Uno de los conceptos fundamentales en inferencia bayesiana es el concepto de probabilidad condicional. La probabilidad condicional, es la probabilidad de que ocurra un suceso A , dado que también tiene lugar otro suceso B . La probabilidad condicional se escribe $P(A|B)$ y se lee «*probabilidad de A dado B* ».

Definición 1.1 (Probabilidad condicional) Sean dos sucesos A y B tales que la la probabilidad de que ocurra B no sea nula, $P(B) > 0$, se define la probabilidad de A dado B como

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \quad (12)$$

siendo $P(A \cap B)$ la probabilidad conjunta de los dos sucesos, o dicho de otra forma, la probabilidad de que se den los dos sucesos simultáneamente.

A partir de la definición de probabilidad condicional podemos escribir que

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A).$$

Si los sucesos son independientes entre si, $P(A|B) \equiv P(A)$ y $P(B|A) \equiv P(B)$, y por tanto la probabilidad de ocurrencia de ambos sucesos de forma simultanea es igual al producto de la probabilidad de ocurrencia de los sucesos de forma independiente, esto es,

$$P(A \cap B) = P(A)P(B).$$

En el caso de que los sucesos no sean independientes, la probabilidad de que tenga lugar A dado B será:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)},$$

que es lo que se llama *Teorema de Bayes*¹⁴ para los sucesos A y B .

Teorema 1.1 (Teorema de Bayes) Sea Ω un conjunto de sucesos compuesto por un conjunto n de particiones mutuamente excluyentes¹⁵ ($\Omega = \{H_1, H_2, H_3 \dots H_n\}$, $H_i \cap H_j = \emptyset$ siendo $i \neq j$) y sea B un suceso tal que $B \subset \Omega$, la probabilidad de que el suceso B sea consecuencia de H_i viene dada por

¹⁴El teorema de Bayes es un resultado incontrovertible partiendo de la definición de probabilidad condicional y los axiomas de Kolmogórov.

¹⁵Tanto el Teorema de Bayes como el Teorema de la probabilidad total, son también válidos cuando existe un conjunto infinito numerable de causas disjuntas dos a dos.

$$P(H_i|B) = \frac{P(B|H_i) P(H_i)}{P(B)}, \quad (13)$$

donde el divisor $P(B)$ representa la probabilidad de que tenga lugar el suceso B .

Teorema 1.2 (Teorema de la Probabilidad Total) Sea Ω un conjunto de sucesos compuesto por un conjunto n de particiones mutuamente excluyentes ($\Omega = \{H_1, H_2, H_3 \dots H_n\}$, $H_i \cap H_j = \emptyset$ siendo $i \neq j$) y sea $B \subset \Omega$ un suceso cualquiera del que se conocen las probabilidades condicionales $P(B|H_i)$, entonces la probabilidad del suceso B viene dada por

$$P(B) = \sum_{i=1}^n P(B|H_i)P(H_i). \quad (14)$$

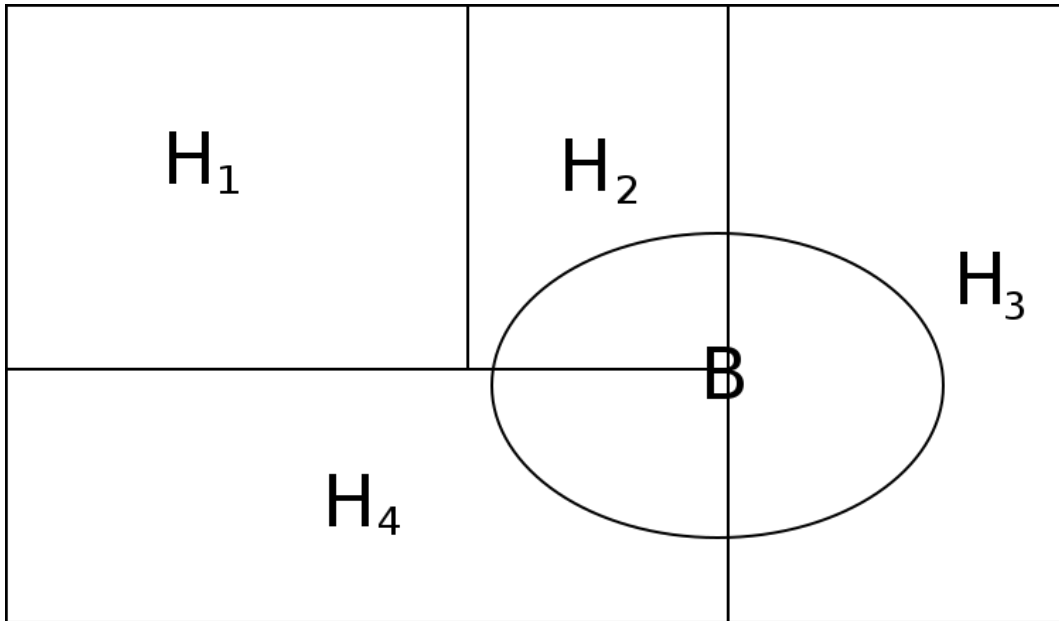


Figura 3: Representación de un espacio muestral Ω de área unidad, constituido por cuatro particiones mutuamente excluyentes, ($\Omega = \{H_1, H_2, H_3, H_4\}$, $H_i \cap H_j = \emptyset$ siendo $i \neq j$). En esta representación un conjunto de suceso B equivale a una región del espacio muestral con una forma y una superficie características.

Tanto el Teorema de Bayes como el Teorema de la Probabilidad Total admiten una interpretación gráfica. El espacio muestral Ω puede representarse por una superficie de área unidad, donde cada subconjunto X tiene una probabilidad de ocurrir igual al área que ocupa $P(X)$, (Figura 3). Las probabilidades condicionales se representan por las regiones que forman parte simultáneamente de dos subconjuntos del espacio muestral. En la Figura 3, el subconjunto B es el único que solapa con otros subconjuntos. Las áreas de las regiones de B que forman parte de uno de los subconjuntos H_i vienen representadas por $P(B|H_i)P(H_i)$, que es equivalente al área del subconjunto H_i que forma parte de B y que viene representado por $P(H_i|B)P(B)$.

1.5.2 El factor de Bayes

Supongamos que disponemos de un conjunto de datos $D = \{x_1, x_2 \dots x_n\}$ y dos hipótesis alternativas H_1 y H_2 para las que suponemos la función verosimilitud $p(D|H_1)$ y $p(D|H_2)$ y una *probabilidad a priori* $p(H_1)$ y $p(H_2)$. Debido a la evidencia de los datos, se produce una transformación de la *probabilidad a priori* en *probabilidad a posteriori* $p(H_k|D)$ a través del teorema de Bayes,

$$p(H_k|D) = \frac{p(D|H_k)p(H_k)}{p(D)} = \frac{p(D|H_k)p(H_k)}{p(D|H_1)p(H_1) + p(D|H_2)p(H_2)} \quad (k = 1, 2)$$

donde el subíndice k indica de qué hipótesis se trata. Para elegir cuál de las dos es la hipótesis es más verosímil, se realiza el cociente de la *probabilidad a posteriori* de cada hipótesis,

$$\frac{p(H_1|D)}{p(H_2|D)} = \frac{p(D|H_1) p(H_1)}{p(D|H_2) p(H_2)}.$$

Al cociente

$$B_{12} = \frac{p(D|H_1)}{p(D|H_2)}$$

se le denomina factor de Bayes.

Definición 1.2 (Factor de Bayes) *El factor de Bayes se define como el cociente de la función verosimilitud de dos hipótesis alternativas*

$$B_{12} = \frac{p(D|H_1)}{p(D|H_2)}, \quad (15)$$

donde los subíndices hacen referencia a la hipótesis 1 y 2 respectivamente.

Si el valor de B_{12} es superior a la unidad, el valor de $p(D|H_1)$ es superior al de $p(D|H_2)$ y la confianza sobre H_1 es mayor que sobre H_2 . En caso de ser inferior a la unidad, los datos refuerzan la confianza de H_2 sobre H_1 . En adelante, tomaremos los valores de la Tabla 1 como escala referencia.

B_{12}	$\log_{10}(B_{12})$	Confianza sobre H_1
<1	<0	H_1 es refutada
1 — 3.2	0 — $\frac{1}{2}$	Débil
3.2 — 10	$\frac{1}{2}$ — 1	Sustancial
10 — 31.6	1 — $\frac{3}{2}$	Fuerte
31.6 — 100	$\frac{3}{2}$ — 2	Muy fuerte
>100	>2	Absoluta

Tabla 1: Escala de referencia que nos relaciona el valor del factor de Bayes con la confianza que tenemos sobre la hipótesis H_1 . Se trata de la tabla de referencia utilizada por Harold Jeffreys en 1961 (consultar [Kass and Raftery 1995](#)).

En el caso particular del método de cross-identificación que se va a utilizar en este trabajo, lo que pretendemos es contrastar la hipótesis H_1 de que los emparejados considerados están formados por observaciones de un mismo objeto astronómico frente a la hipótesis H_2 de que se trata de observaciones pertenecientes a dos objetos diferentes.

Si las medidas de un parámetro pertenecen a un único objeto astronómico la función densidad de probabilidad conjunta $p(D|\theta, H_k)$ se obtiene mediante el producto de las funciones densidad de probabilidad asociadas para cada medida $p_i(x_i|\theta, H_k)$ que representan la probabilidad de que una medida x_i se corresponda exactamente con su verdadero valor θ y la función verosimilitud para H_1 resulta ser

$$p(D|H_1) = \int p(\theta|H_1)p(D|\theta, H_1) d\theta = \int p(\theta|H_1) \prod_{i=1}^n p_i(x_i|\theta, H_1) d\theta, \quad (16)$$

siendo $p(\theta|H_1)$ la densidad de *probabilidad a priori* de la hipótesis H_1 . Para el caso de la hipótesis H_2 las medidas x_i pertenecen a distintos valores verdaderos θ_i por lo que la función verosimilitud de H_2 se calcula como

$$p(D|H_2) = \prod_{i=1}^n \left[\int p(\theta_i|H_2)p_i(x_i|\theta_i, H_2) d\theta_i \right]. \quad (17)$$

En los casos en que disponemos de q conjuntos de medidas D_s diferentes, asociados a lo distintos parámetros, dispondremos de q factores de Bayes. Suponiendo cada uno de los parámetros que hemos elegido son igualmente válidos para determinar la verosimilitud de nuestras hipótesis, el factor de Bayes conjunto se obtiene como el producto de los factores de Bayes asociados a cada parámetro,

$$B_{12} = B_{12}^1 \times B_{12}^2 \dots B_{12}^q = \prod_{s=1}^q \frac{p(D_s|H_1)}{p(D_s|H_2)}. \quad (18)$$

De esta manera, cada nueva observación cambia el conjunto de valores D_s lo que cambia la verosimilitud de cada hipótesis y en consecuencia el factor de Bayes. En esto consiste la actualización de creencias basada en el razonamiento bayesiano.

2 Selección de las muestras

En esta sección se explica qué es el proyecto *H-ATLAS* y qué es el proyecto GAMA, de qué instrumentos se sirven estos proyectos para realizar las medidas y que datos nos proporcionan. Por último, se mostrará la zona en la que se produce un solapamiento de ambos cartografiados que es la zona en la podemos aplicar nuestra propuesta para la identificación del SLGs.

2.1 Proyecto *H-ATLAS*

H-ATLAS (*Herschel* Astrophysical Terahertz Large Area Survey) es el nombre de uno de los proyectos astronómicos que se fundamenta en las medidas del telescopio espacial *Herschel*. Este proyecto cubre un área total de 550 deg^2 (la octogésima parte del cielo) requiriendo unas 600 horas de observación, lo que implica un tiempo cuatro veces mayor que todas las demás prospecciones extragalácticas de *Herschel* combinadas. Con ello se espera detectar unas $2,5 \times 10^5$ galaxias con desplazamientos al rojo de hasta $z \sim 4$, cuando el Universo tenía apenas unos pocos miles de millones de años ([H-ATLAS, 2018](#)).

2.1.1 Observatorio espacial *Herschel*.

El observatorio espacial *Herschel*¹⁶ es un proyecto de la ESA (European Space Agency), lanzado el día 14 de mayo de 2009 junto con el satélite espacial Planck. El periodo de vida útil terminó el día 29 de Abril de 2013 cuando agotaron los 2200 litros de helio superfluido que utilizaba como refrigerante. Se encontraba situado en el punto L_2 de Lagrange, a unos $1.5 \times 10^6 \text{ km}$ de la Tierra; este es un punto de especial interés para colocar observatorios espaciales, puesto que los cuerpos situados ahí mantienen la misma posición relativa entre el Sol y la Tierra. Se trata del mayor telescopio infrarrojo espacial que se ha construido hasta el momento. Dispone de un único espejo de 3.5 metros de diámetro y de varios instrumentos diseñados para realizar observaciones en el rango espectral que va desde el IR lejano hasta la banda submilimétrica, $55 - 672 \mu\text{m}$.

Los tres instrumentos del observatorio espacial son:

- SPIRE (Spectral and Photometric Imaging Receiver) [SPIRE \(2018\)](#).

Se trata de una cámara fotométrica y un espectrómetro de transformada de Fourier. Ambos instrumentos operaban refrigerados por He líquido superfluido a una temperatura de $\sim 0.3 \text{ K}$. SPIRE se construyó con dos objetivos, el estudio de la formación de estrellas y la formación de galaxias. En ambos casos está involucrado un proceso similar de absorción de la luz visible y UV procedente de las estrellas y posterior remisión de radiación IR por el gas y polvo interestelar, con una longitud de onda de unos $100 \mu\text{m}$.

La cámara opera simultáneamente en tres bandas del espectro electromagnético, centradas en $250 \mu\text{m}$, $350 \mu\text{m}$, y $500 \mu\text{m}$, con una resolución angular de $20 - 30 \text{ arcsec}$ y un campo de visión de 4×8 minutos de arco. El sensor cuenta con 270 píxeles.

El espectrómetro tiene una resolución espectral de 300 km s^{-1} , cubre el rango de frecuencias entre $200 - 670 \mu\text{m}$ y su resolución angular es de $20 - 50 \text{ arcsec}$. El campo de visión es de 2.6×2.6 minutos de arco. En cuanto al detector, dispone de 56 píxeles.

El instrumento ha sido construido por un consorcio internacional de 18 instituciones, de 8 países, liderado por el Dr. Matt Griffin del Cardiff Institute (AIG) ([Herschel, 2018](#)).

¹⁶El satélite fue nombrado en honor al astrónomo británico William Herchel conocido entre otras cosas, por el descubrimiento del planeta Urano y del espectro infrarrojo. No confundir con el telescopio óptico e IR-cercano llamado William Herschel Telescope (WHT) que se encuentra en el Observatorio del Roque de los Muchachos en la isla de La Palma.

- PACS (Photodetecting Array Camera and Spectrometer) ([PACS, 2018](#)).

Al igual que SPIRE, PACS está formado por dos instrumentos independientes; una cámara y un espectrómetro integral de campo.

La cámara toma imágenes en dos bandas de forma simultánea, a $60 - 85 \mu\text{m}$ o $85 - 130 \mu\text{m}$ y a $130 - 210 \mu\text{m}$ mediante dos sensores bolométricos. Los sensores bolométricos están formados por un array de 32×16 y de 64×32 respectivamente. La resolución angular es de 5 arcsec y el campo de visión es de 1.75×3.5 minutos de arco. El límite de detección está en 3 mJy ¹⁷.

Al espectrómetro también trabaja simultáneamente en dos frecuencias, en $57 - 105 \mu\text{m}$ y $105 - 210 \mu\text{m}$. Los detectores están compuestos por un array de 5×5 y de 16×25 sensores de Ge-Ga. La resolución espectral es de $150 - 200 \text{ km s}^{-1}$, la resolución espacial es de 10 arcsec y el campo de visión de 50×50 . Para $\frac{\lambda}{\Delta\lambda} \sim 1500$ la sensibilidad es de $5 \times 10^{-18} \text{ W m}^{-2}$.

El instrumento ha sido construido por un consorcio internacional de 12 instituciones pertenecientes a 6 países, liderado por el investigador Dr. Albrecht Poglitsch del Max-Planck-Institute([Herschel, 2018](#)).

- HIFI (Heterodyne Instrument for the Far Infrared) ([HIFI, 2018](#)).

Se trata de un espectrómetro de muy alta resolución espectral (0.02-0.7 km/s) que trabaja entre los $157 - 625 \mu\text{m}$. Tiene una resolución espacial de $13 - 40$ arcsec. La temperatura de trabajo se encuentra entre los $2 - 10 \text{ K}$.

Al igual que los instrumentos anteriores, este ha sido construido por un consorcio internacional de 26 instituciones pertenecientes a 11 países diferentes. El líder principal del proyecto en este caso es Thijs de Graauw del Stichting Ruimte Onderzoek Nederland (SRON)([Herschel, 2018](#)).

2.1.2 Catálogo *H*-ATLAS DR1.

El lanzamiento de datos *H*-ATLAS DR1 (Data Release 1) fue publicado el 28 de junio de 2016. Los detalles sobre su contenido se encuentran en las publicaciones [Valiante et al. \(2016\)](#) y [Bourne et al. \(2016\)](#). Este lanzamiento de datos incluye varios mapas y archivos adicionales; en este trabajo se utilizará únicamente el fichero HATLAS_DR1_CATALOGUE_V1.2.FITS¹⁸. Se compone de tres campos centrados en ascensión recta 09h, 12h y 14.5h sobre el ecuador celeste (denominadas por el proyecto Bloques, Figura 4) que cubren cubren un área de total de 161 deg^2 con 120230 fuentes identificadas en 3 bandas fotométricas.

A continuación se muestra una breve descripción de las columnas del catálogo²⁰ que son relevantes en este trabajo:

- HATLAS_IAU_ID: Contiene el identificador de la fuente astronómica del catálogo *H*-ATLAS asignado por la IAU (International Astronomy Union).
- RA: Ascensión recta, en grados, de la fuente astronómica obtenida a partir de los datos de la banda de $250 \mu\text{m}$.
- DEC: Declinación, en grados, de la fuente astronómica obtenida a partir de los datos de la banda de $250 \mu\text{m}$.

¹⁷El Jansky (Jy) es una unidad de densidad de flujo espectral o irradiancia, especialmente utilizada en astronomía que no pertenece al Sistema Internacional de Unidades (SI). Es equivalente a $10^{-26} \text{ W m}^{-2}\text{Hz}^{-1}$.

¹⁸Dirección de descarga: http://www.h-atlas.org/public_data/DR1/HATLAS_DR1_CATALOGUE_V1.2.FITS

¹⁹Página web del proyecto que desarrolla TOPCAT: <http://www.star.bris.ac.uk/%7Embt/topcat/>

²⁰Para una descripción completa de cada una de las columnas del fichero consultar la dirección:

http://www.h-atlas.org/public_data/DR1/HATLAS_DR1_CATALOGUE.COLUMNS

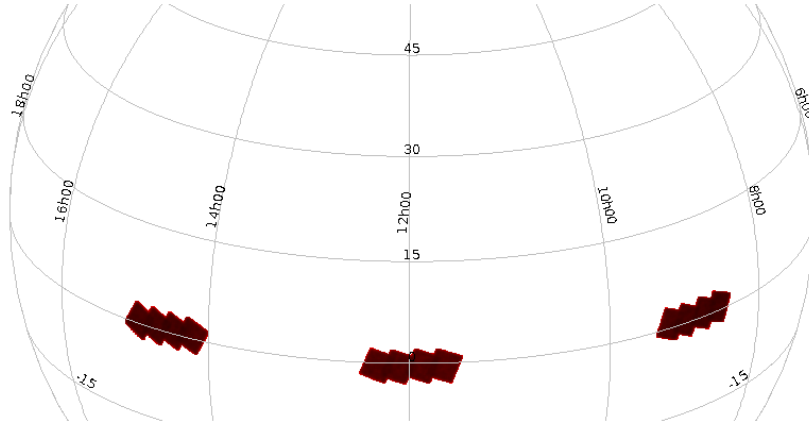


Figura 4: Representación de los objetos publicados en el catálogo *H*-ATLAS Data Release 1 (DR1) en coordenadas ecuatoriales utilizando el programa TOPCAT¹⁹(Tool for OPERations on Catalogues And Tables). De derecha a izquierda, cada una de las tres regiones recibe el nombre de Bloque 2, Bloque 3 y Bloque 4. (ver: <http://www.h-atlas.org/survey/fields>)

- F250: Flujo de la fuente, en Jy, de la banda de 250 μm . El límite de detección 5σ ²¹ de esta banda se encuentra en 33.5 mJy (González-Nuevo et al., 2012)
- F350: Flujo, en Jy, de la banda de 350 μm . El límite de detección 5σ , 37.7 mJy.
- F500: Flujo, en Jy, de la banda de 500 μm . El límite de detección 5σ , 44.0 mJy.
- E250: Desviación estándar (1σ) del flujo para la observación de un objeto en la banda 250 μm .
- E350: Desviación estándar del flujo en la banda 350 μm .
- E500: Desviación estándar del flujo en la banda 500 μm .
- GSQ_FLAG: Se hace una clasificación del tipo de objeto basada en diagramas de color g-i / J-K además de los criterios de PSF / magnitud de modelo de banda r (Bourne et al., 2016). La columna contiene un entero con tres posibles valores, cuyo significado es el siguiente: 0=galaxia, 1=estrella, 2 y 3=cuásar.
- Z_SPEC: Contiene los valores del *redshift* de algunos de los objetos del catálogo *H*-ATLAS. Aquellos cuyo valor es desconocido y no ha podido calcularse fotométricamente aparecen con valor -99.
- Z_QUAL: Etiqueta que indica el nivel de confianza Q_h que se tiene sobre el valor del *redshift* que aparece en la columna Z_SPEC. Los objetos cuya etiqueta tiene un valor ≥ 3 , tienen *redshifts* espectroscópicos considerados de confianza. El proyecto indica que los *redshifts* fotométricos han sido calculados mediante ANNz; por ese motivo hemos tratado a aquellas medidas con un factor de calidad 1 o 2 como si hubieran sido obtenidos mediante ANNz.
- Z_SOURCE: Indica la fuente de procedencia de los valores del *redshift* del catálogo. El código es el siguiente: 1- SDSS DR7, 2- 6dFGS, 4- 2SLAQ-QSO, 8- 2SLAQ-LRG, 16- GAMA HATLAS filler targets, 32- GAMA Main Survey, 64- 2dFGRS, 128- SDSS DR10, 256- Wigglez, 512- GAMA (que no se encuentran actualmente en TilingCat de GAMA-II)

²¹Una desviación estándar de 1σ significa que si asumimos una distribución normal de los valores posibles que puede tomar una medida respecto a su valor verdadero x , hay una probabilidad de en torno al 68% de que esta se encuentre en el intervalo $x \pm \sigma$. El límite de detección 5σ es un criterio para determinar cuándo se ha detectado una fuente sobre la señal de ruido de fondo.

De las 120230 fuentes del catalogo DR1 solo 28389 tienen un *redshift* con factor de calidad $Q_h \geq 3$ y 931 factor de calidad $Q_h = 1 \vee 2$ (ver fig: 5). Al resto de fuentes, cuyo *redshift* es desconocido, les aplicaremos el algoritmo descrito en la sección 3. Si estos resultan ser galaxias (según la clasificación de la columna GSQ_FLAG) y el ajuste proporciona un valor comprendido entre $1 < z < 3.5$ aceptaremos el valor del ajuste como medida válida de z .

En cuanto a la desviación estándar de las medidas del *redshift* el proyecto no proporciona su valor directamente. Para conocer su valor tendríamos que conocer los detalles de cómo se ha obtenido el *redshift* dependiendo de la procedencia de las medidas, lo cual implica mucho tiempo, que no está justificado emplear en este trabajo de final de grado. Una alternativa hubiera sido buscar los objetos en otra base de datos como NED (NASA/IPAC Extragalactic Database: <http://ned.ipac.caltech.edu/>) o SIMBAD (Set of Identifications, Measurements, and Bibliography for Astronomical Data: <http://simbad.u-strasbg.fr/simbad/>) ya que existen rutinas ya implementadas en Python para este tipo de tareas. Sin embargo, nos podemos encontrar en la situación de que estas bases de datos tampoco dispongan de las desviaciones estándar de todos los objetos en los que estamos interesados²². Por ese motivo, la solución que se ha adoptado en este trabajo ha sido dividir las medidas proporcionadas por *H-ATLAS* en dos grupos. Por una parte tenemos los objetos cuyo factor de calidad es $Q_h = 1 \vee 2$. Estos objetos suponemos que se han obtenido todos ellos mediante ANNZ, por lo que si las medidas se encuentran en el rango $0 < z < 0.7$, es razonable suponer para todas ellas $\sigma^z = 0.023$ a partir del estudio realizado por Collister and Lahav (2004). En el caso de que el factor de calidad sea $Q_h \geq 3$ la desviación estándar de las medidas se obtendrá a partir del ajuste que se muestra en la Figura 8 (asignar la misma desviación estándar a los desplazamientos al rojo espectroscópicos de los proyectos GAMA y *H-ATLAS* está justificado porque la procedencia de estas medidas es la misma en muchos casos). El hecho de que una medida tenga un factor de calidad más alto que otra, no quiere decir que el valor de las desviaciones estándar sea diferente. El factor de calidad indica la confianza subjetiva que tienen los autores del catálogo sobre las medidas del *redshift* (Driver et al., 2011). La confianza puede ser baja por cualquier motivo.

En resumen, para asignar los errores a las medidas del *redshift* de los objetos de *H-ATLAS* se ha decidido lo siguiente:

- Si el factor de calidad de la medida del *redshift* por el proyecto *H-ATLAS* es ≥ 3 , el *redshift* es espectroscópico y la desviación estándar se obtiene a partir de la expresión $\sigma^z = 1.1 \times 10^{-4} \times (1 + z)$.
- Si el factor de calidad $Q_h = 1 \vee 2$, se tratará a la medida como si hubiera sido obtenida mediante ANNZ. Si su valor pertenece al intervalo $0 < z < 0.7$ se le asignará una desviación estándar $\sigma^z = 0.023$.
- Si el *redshift* es desconocido por el proyecto *H-ATLAS* y el ajuste obtenido mediante el algoritmo descrito en la Sección 3 se encuentra en el intervalo $1 < z < 3.5$ consideraremos válido este valor y le asignaremos una desviación estándar $\sigma^z = 0.115 \times (1 + z)$.

Por último, otro dato que es de gran importancia es el valor de la resolución angular. El valor de la FWHM²³ para el catálogo *H-ATLAS* es de 17.98 arcsec en la banda de $250 \mu\text{m}$ ²⁴. Nosotros obtenemos el valor de la desviación típica como $\sigma_h^p = \text{FWHM} \times (2\sqrt{2 \ln 2})^{-1} \sim 7.63 \text{ arcsec}$.

²²Se ha realizado una búsqueda individual de unos cuantos objetos del catálogo *H-ATLAS* (30 objetos), cuyo *redshift* tiene $Q_h \geq 3$ a partir del identificador asignado por la IAU en la base de datos NED y se ha encontrado el valor de z y σ^z solamente para unos pocos casos (3).

²³FWHM es la anchura a media altura (Full Width at Half Maximum) de la dispersión de las medidas suponiendo que se trata de una distribución normal.

²⁴El dato se encuentra en: http://www.h-atlas.org/public_data/HATLAS_SDP_catalogue.README

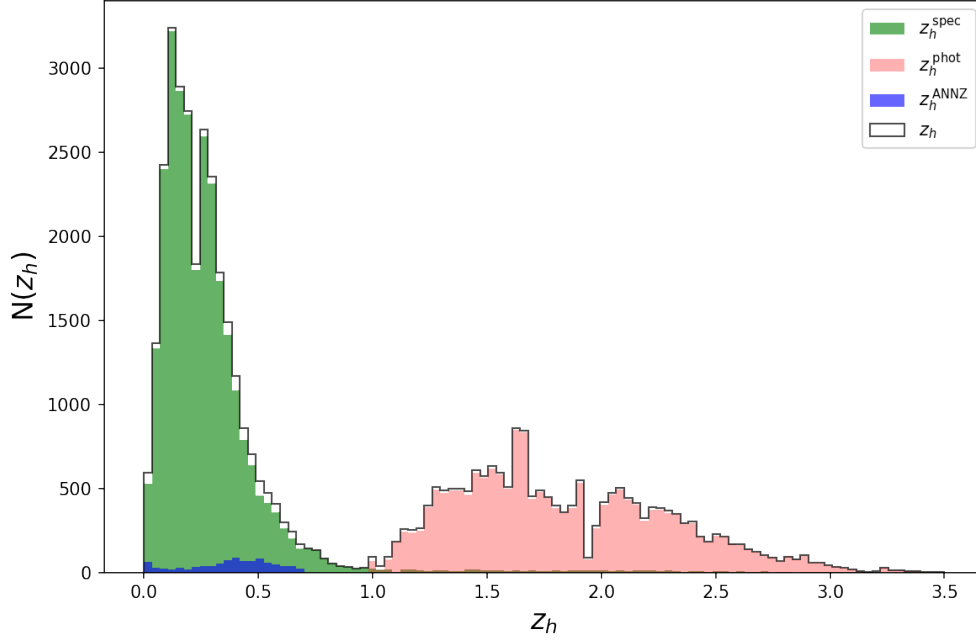


Figura 5: Histograma del *redshift* de los 47844 fuentes del proyecto *H-ATLAS* de los que el disponemos medidas o estimaciones razonables del *redshift* z_h de un total de 120230 para el catálogo completo. Estas fuentes se encuentran representadas en tres categorías; objetos de los que se dispone de una medida espectroscópica de confianza $N(z_h^{spec}) = 28389$, objetos cuyo *redshift* tiene un nivel de confianza $Q_h = 1/2$ (tratados como si hubieran sido obtenidos mediante ANNZ) $N(z_h^{annz}) = 931$ y objetos cuyo *redshift* ha sido obtenido mediante el método descrito en la Sección 3 $N(z_h^{phot}) = 18524$.

2.2 Proyecto GAMA

GAMA (Galaxy And Mass Assembly) es un proyecto internacional que hace uso de los más modernos observatorios espaciales y terrestres con el objetivo principal de estudiar estructuras de entre 1 kpc a 1 Mpc, lo cual incluye escalas que abarcan desde la estructura interna de las galaxias hasta cúmulos de galaxias (GAMA, 2018). Concretamente se pretende mejorar en tres asuntos clave respecto a otros estudios:

- Mejora de la eficiencia espectroscópica, permitiendo el muestreo integral desde galaxias para *redshift* intermedios y mostrar esa información en un mismo estudio.
- Mejorar la resolución espacial para estudiar la estructura de las galaxias próximas y los procesos de formación galáctica.
- Mejorar el rango de la cobertura espectral.

Es destacable el amplio rango espectral que cubre el proyecto que abarca desde la región de Rayos X, hasta la región de radio de alta frecuencia (90 cm). GAMA, hace uso de una amplia variedad de medidas procedentes de otros proyectos, entre los cuáles destacan:

- Cartografiados públicos: Sloan Foundation 2.5m SDSS
- United Kingdom Infrared Telescope (UKIRT) UKIDSS-LAS
- Campañas GAMA: Galaxy Evolution Explorer (GALEX) GALEX-GAMA
- Giant Metrewave Radio Telescope (GMRT) GMRT-GAMA
- Cartografiados relacionadas con GAMA: VLT Survey Telescope (VST) KiDS
- Visible and Infrared Survey Telescope for Astronomy (VISTA) VIKING
- The Canada France Hawaii Lensing Survey (CFHTLenS)

- Observatorio Espacial Herschel H-ATLAS
- Australian Square Kilometre Array Pathfinder (ASKAP) DINGO
- X-ray Spectroscopy Mission and the X-ray Multi-Mirror Mission (XMM-Newton) XMM-XXL
- Wide-Field Infrared Survey Explorer (WISE)

El estudio espectroscópico de GAMA cubre aproximadamente 3×10^5 galaxias de hasta magnitud 19.8, repartidas en un área de 286 deg^2 . Estas medidas son fruto de 210 noches de observación en un periodo de 7 años, desde 2008 hasta 2014, realizadas con el espectrógrafo AAOmega en el telescopio Anglo-Australiano (AAT) por miembros del equipo de GAMA. Estos datos han sido completados por datos procedentes de estudios previos como el Sloan Digital Sky Survey (SDSS), el 2dF Galaxy Redshift Survey (2dFGRS) y el Millennium Galaxy Catalogue (MGC).

2.2.1 Catálogo GAMA I

El proyecto GAMA ha publicado dos bases de datos denominadas GAMA DR1 y GAMA DR2. La primera, fue publicada el 25 de junio de 2010 y contiene el *redshift* y otra información adicional de 114441 objetos repartidos por tres regiones, denominadas G09, G12 y G15 (Estas solapan parcialmente con las regiones que cubre el proyecto *H*-ATLAS). Estas regiones tienen una forma aproximadamente rectangular de $12 \times 4 \text{ deg}$ y están situadas sobre el ecuador celeste, sumando un total de $\sim 144 \text{ deg}^2$. El límite de magnitud para estas fuentes es de 19.4 en las regiones G09 y G15 y 19.8 en la región G12.

La segunda publicación cubre el mismo área del cielo, pero representa un conjunto de objetos más reducido de que la primera, con más información adicional. El límite de magnitud de los objetos pertenecientes a esta publicación es de 19.0 en la región G09 y G12 y de 19.4 para la región G15. Los datos de la región G15 son prácticamente los mismos en ambas publicaciones.

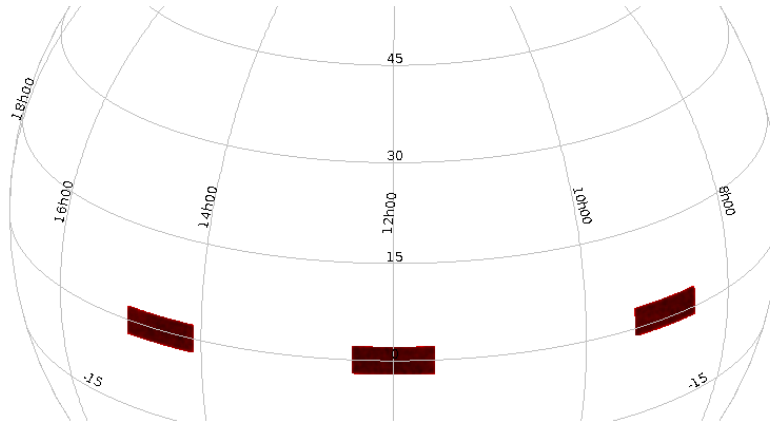


Figura 6: Representación de los objetos publicados en el catálogo GAMA DR1 en coordenadas ecuatoriales utilizando el programa TOPCAT. De derecha a izquierda, las regiones reciben los nombres G09, G12, G15.

En este trabajo usaremos los datos de la publicación GAMA DR1²⁵. La descripción completa de las columnas del catálogo se encuentra en el artículo [Driver et al. \(2011\)](#); nosotros mostramos aquí una descripción solo de las columnas que vamos a utilizar:

²⁵El fichero con los datos de esta publicación se encuentra alojado en la dirección: http://www.gama-survey.org/dr1/data/GamaCoreDR1_v1.fits

- GAMA_IAU_ID: Identificador de la fuente astronómica del catalogo GAMA asignado por la IAU.
- RA: Ascensión recta, en grados, de la fuente astronómica obtenida del proyecto SDSS DR6.
- DEC: Declinación, en grados, de la fuente astronómica obtenida del proyecto SDSS DR6.
- Z_HELIO: Valores del *redshift* heliocéntrico proporcionados por el proyecto GAMA. Aquellas medidas que no se encuentran disponibles se indican con un -2 o 9999.
- Z_QUALITY: Etiqueta que indica la confianza que se tiene sobre los valores presentes en la columna Z_HELIO. Nos referiremos a esta etiqueta como Q_g .
- Z_SOURCE: Etiqueta que indica la procedencia de la medida del *redshift*. El criterio es el siguiente: 1 = SDSS DR6, 2 = 2dFGRS, 3 = MGC, 4 = 2SLAQ-LRG, 5 = GAMA, 6 = 6dFGS, 7 = UZC, 8 = 2QZ, 9 = 2SLAQ-QSO, 10 = NED.

El proyecto GAMA solo proporciona medidas espectroscópicas del *redshift*. De las 114441 fuentes presentes en GAMA DR1, solo 59479 tienen *redshift* con factor de calidad $Q_g \geq 3$ (los factores de calidad de los proyectos GAMA y *H*-ATLAS son independientes). Los detalles relativos al factor de calidad asignado a las medidas espectroscópicas del proyecto GAMA se encuentran en [Driver et al. \(2011\)](#).

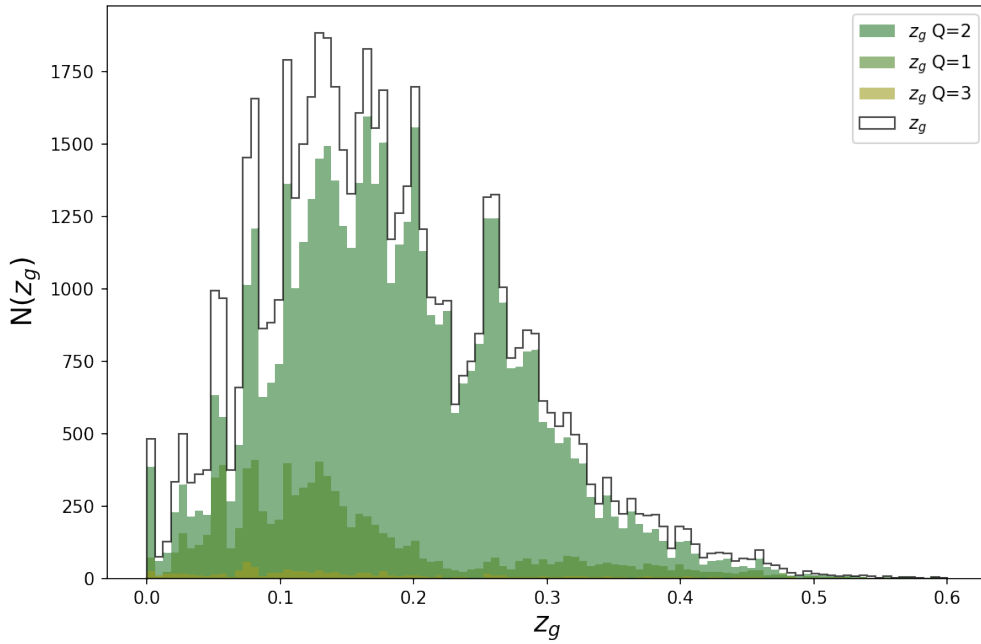


Figura 7: Histograma que representa el número de objetos del proyecto GAMA de los cuales su *redshift* está disponible, $N(z_g) = 59479$. Todas las medidas del *redshift* son espectroscópicas, no obstante, el proyecto les ha asignado distintos factores de calidad. Cuanto menor es el factor Q_g mayor es la confianza que se tiene sobre esa medida. Hay 9197 con factor de calidad $Q_g = 1$, 49416 con $Q_g = 2$, y 866 con $Q_g = 3$. Hay objetos en este catálogo con $z_g > 0.6$, pero no se encuentran aquí, porque suponen una parte muy pequeña de la población representada.

Al igual que ocurre con *H*-ATLAS, este proyecto tampoco proporciona la desviación estándar asociada a las medidas del *redshift*. En la Figura 8 se muestra el ajuste que se ha realizado para estimar la dependencia σ^z con z a partir de una muestra de 20 objetos del catálogo GAMA, con factor de calidad $Q_g=5$ cuyas medidas se encuentran disponibles en NED (ver Tabla B).

En el caso de las posiciones proporcionadas por GAMA el valor de la FWHM = 0.7 arcsec, por lo que el valor de la desviación típica en este caso es $\sigma_g^p \sim 0.30$ arcsec ([Driver et al., 2009](#); [Driver and GAMA team, 2008](#)).

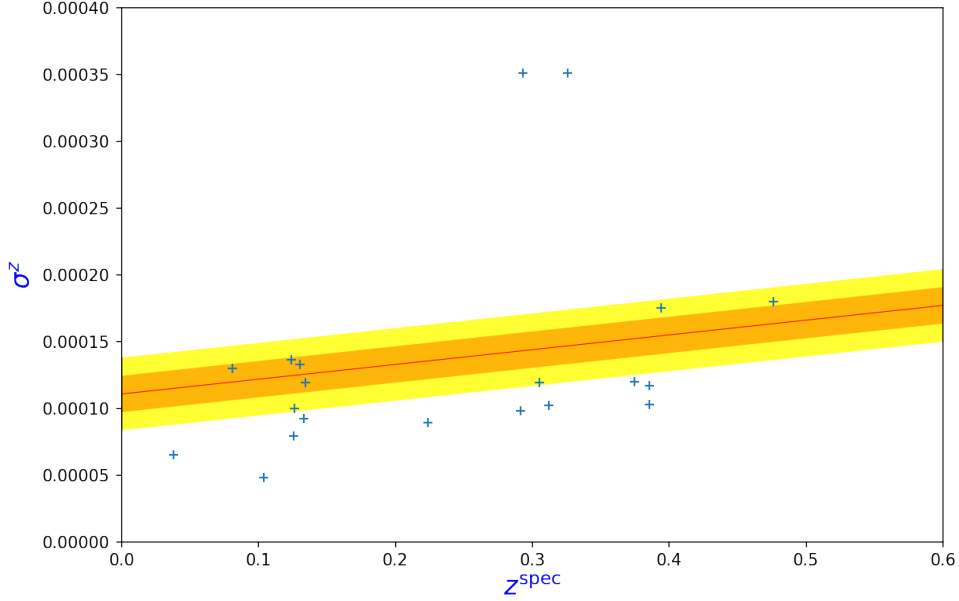


Figura 8: Representación de la desviación estándar del *redshift*, σ^z en función del *redshift* espectroscópico, z^{spec} de una muestra de 20 objetos del catálogo GAMA (los 20 primeros objetos que aparecen en la Tabla 3). La recta se ha obtenido mediante un ajuste lineal de la forma $\sigma^z = a \times (1 + z)$ con a como parámetro de libre. El parámetro resultante del ajuste ha sido $a = (11 \pm 1) \times 10^{-5}$.

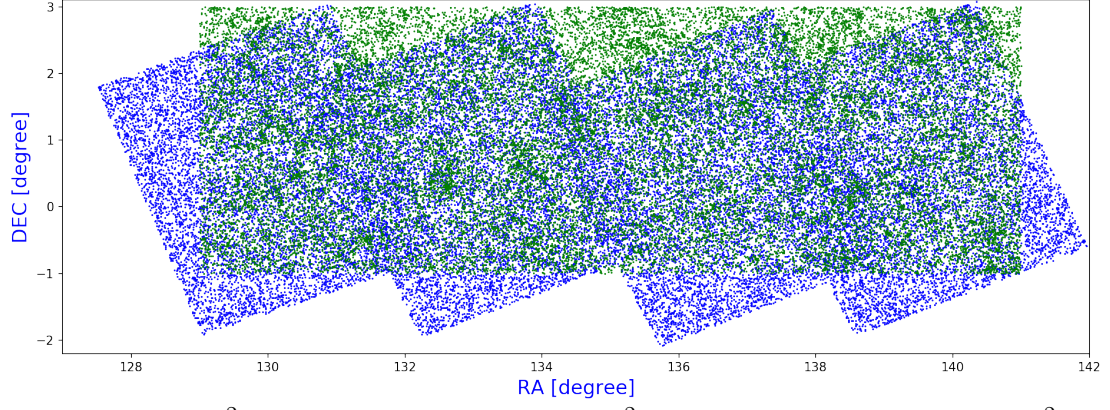
2.3 Región de solapamiento de ambos catálogos

Los proyectos GAMA y *H*-ATLAS cubren áreas pequeñas en comparación con la superficie total de la esfera sobre el ecuador celeste (ver Figura 6 y Figura 4). Esto permite proyectar estas regiones sobre el plano sin cambiar demasiado el valor de la superficie²⁶.

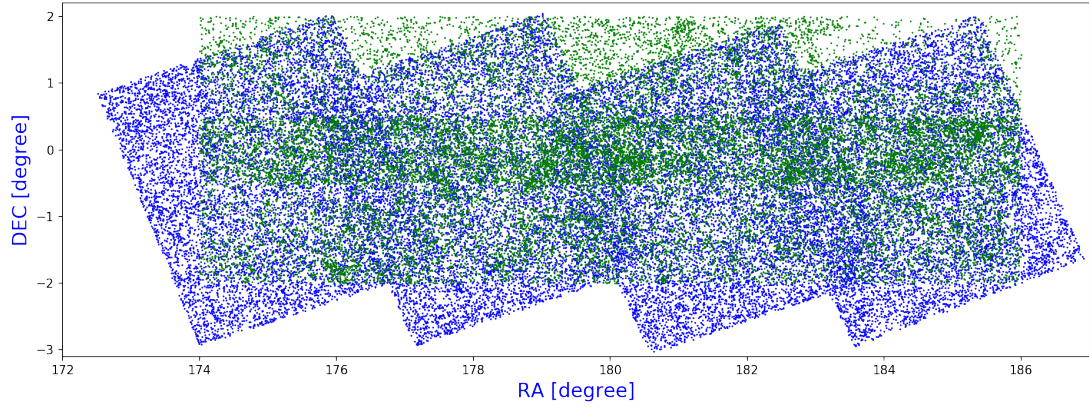
Las áreas cubiertas por GAMA tienen una forma que se aproxima muy bien a un rectángulo, mientras las regiones de *H*-ATLAS forman un polígono de 16 vértices con el que resulta más difícil trabajar. Por ese motivo, para calcular las áreas de las regiones de intersección de ambos catálogos se ha recurrido a la función `area_region` que se encuentra definida en el Apéndice C.9. De forma resumida, lo que se hace es proyectar los puntos sobre un plano y crear una red con celdas cuadradas. Después se hace un conteo de las celdas que tienen objetos de GAMA y de *H*-ATLAS a una distancia de su centro igual o inferior a la mitad de la diagonal de la celda. De esta manera se obtiene el área de las regiones de solapamiento como el producto del área de la celda por el número surgido del conteo.

Al aplicar el algoritmo para determinar el área de las regiones cubiertas por cada uno de los catálogos vemos que existen diferencias de entorno a 1 deg^2 con las áreas que consideramos correctas. A partir de esta observación hemos estimado un área de intersección entre ambos catálogos de $130 \pm 1 \text{ deg}^2$.

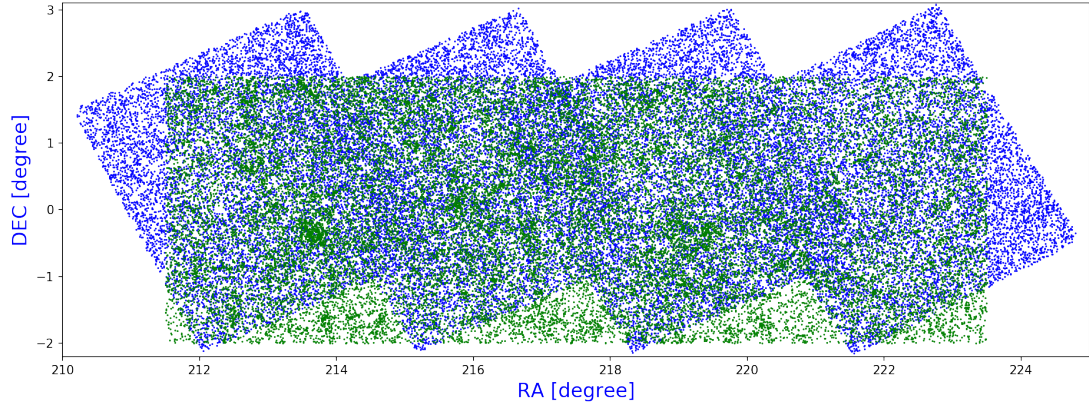
²⁶La superficie de una sección esférica definida entre los paralelos 0 y DEC y los meridianos 0 y AR viene dada por la expresión $A_{sr} = AR \times \sin(DEC)$. Si tomamos por ejemplo la región G12 que se encuentra entre los paralelos DEC=-2 y DEC=2 y meridianos RA=174 y RA=186 el área que obtenemos con la ecuación anterior es de $A_{sr} \simeq 0.014618666993941102 \text{ sr} \simeq 47.99025283641695 \text{ deg}^2$ mientras que al proyectar sobre una superficie plana es 48 deg^2 . La diferencia entre el valor real y la aproximación es por tanto inferior a 0.01 deg^2 .



G09 (48.56 deg^2 , verde) y Bloque 2 (54.15 deg^2 , azul). Solapamiento: 42.96 deg^2 .



G12 (49.03 deg^2 , verde) y Bloque 3 (54.40 deg^2 , azul). Solapamiento: 43.46 deg^2 .



G15 (48.68 deg^2 , verde) y Bloque 4 (54.80 deg^2 , azul). Solapamiento: 43.88 deg^2 .

Figura 9: Superposición de los catálogos GAMA (verde) y *H*-ATLAS (azul). Las áreas que se indican en el pie de cada imagen, se han obtenido a partir de la función `area_region` que se encuentra definida en el Apéndice C.9. El área total cubierto por el catálogo *H*-ATLAS es de $\sim 163.35 \text{ deg}^2$ (según el proyecto *H*-ATLAS el área de los bloques 2, 3 y 4 es de 161 deg^2), el de GAMA de $\sim 146.27 \text{ deg}^2$ (según GAMA el área total de las tres regiones G09, G12 Y G15 es de 144 deg^2) y el área total de solapamiento de $\sim 130.3 \text{ deg}^2$.

3 *Redshift* fotométrico de ETGs a partir de SPIRE

En esta sección se describe un método válido para estimar el *redshift* fotométrico de las ETGs a partir de las medidas del instrumento SPIRE del satélite espacial *Herschel*. El método se basa en el hecho de que las ETGs poseen el máximo absoluto de emisión en el IR-lejano, sobre los $100\ \mu\text{m}$ para el sistema en reposo, debido a la emisión IR por parte del gas y polvo interestelar; dado que las medidas del SPIRE se encuentran a $250\ \mu\text{m}$, $350\ \mu\text{m}$ y $500\ \mu\text{m}$, estas coinciden en torno al máximo cuando $1 \lesssim z \lesssim 3.5$.

Se considera que a la emisión IR de las galaxias *starburst* contribuyen tres componentes diferentes, dependiendo del entorno astrofísico en que se originó: nubes moleculares, nubes difusas de baja densidad (cirros) y regiones circunucleares calentadas por el Núcleo Galáctico Activo (AGN) (Lapi et al., 2011). De las tres, la componente “caliente”, procedente de las nubes de gas moleculares, es la que resulta relevante para el ajuste que se va llevar a cabo, debido a que es la componente más intensa en la zona del espectro en la que SPIRE realiza las medidas. En el rango de $\lambda \sim 50 - 500\ \mu\text{m}$ (considerando el sistema en reposo), la SED de las galaxias *starburst* típicas puede modelarse (mostrando diferencias de entorno al 10 % – 20 %) como suma de dos cuerpos grises²⁷ siendo el flujo S_ν , para cada uno de ellos,

$$S_\nu \propto \frac{\nu^{3+\beta}}{\exp \frac{h\nu}{KT_d} - 1}, \quad (19)$$

con temperaturas $T_d \approx 30\ \text{K}$ y $T_d \approx 60\ \text{K}$ y unos índices de emisividad para el polvo de $\beta = 1.7$ y $\beta = 2$ respectivamente. A diferencia de la mayoría de los algoritmos para determinar el *redshift* fotométrico, el método propuesto tomará como única referencia la SED de la galaxia SMM J2135-0102. Como veremos, esta decisión se fundamenta en los estudios realizados por González-Nuevo et al. (2012) y Lapi et al. (2011).

3.1 Selección de la SED de referencia

La idea de obtener el *redshift* fotométrico a partir de la SED de la galaxia SMM J2135-0102 no es nuestra y las razones para elegir la SED de esta galaxia en concreto se explican con detalle en Lapi et al. (2011). Estos autores seleccionaron un conjunto de cuatro galaxias que consideraron representativas de las galaxias en formación típicas y cuya SED se encuentra bien determinada e hicieron varios estudios para saber cual era la SED más adecuada para realizar los ajustes. Para ello, en primer lugar identificaron las posibles fuentes de error a la hora de realizar el ajuste y consideraron que había dos fuentes de error principales.

La primera fuente de error está relacionada con el hecho de que resolución angular de los instrumentos de medida es menor cuanto mayor es la longitud de onda. Por este motivo la radiación electromagnética procedente de una fuente, tiende a mezclarse con la procedente de las fuentes próximas, lo cuál produce un incremento del flujo medido respecto del valor real. Para cuantificar el efecto que producen las fuentes débiles sobre las medidas del flujo llevaron a cabo simulaciones Rigby et al. (2011) que muestran 56.5 % de las fuentes detectadas a $\geq 5\ \sigma$ a $500\ \mu\text{m}$ muestran un incremento en un factor > 1.5 , y el 27.3 % por un factor > 2 , mientras que si la medida del flujo se encuentra por encima de los $10\ \sigma$, el incremento de flujo debido a este fenómeno ya puede considerarse despreciable. Para saber cómo afecta el aumento de flujo sobre el cálculo del *redshift* fotométrico calcularon los *redshifts* fotométricos de 39 galaxias, utilizado como referencia la SED de cuatro galaxias diferentes, y lo compararon con el obtenido a partir de medidas espectroscópicas (Figura 10). Obtuvieron que el valor medio de la magnitud $\Delta z/(1+z) \equiv (z_{\text{phot}} - z_{\text{spec}})/(1+z_{\text{spec}})$ era menor cuando se utiliza la SED de SMM J2135-0102 como modelo.

²⁷Puede encontrarse mucha más información sobre la ecuaciones del cuerpo gris en el artículo Casey (2012).

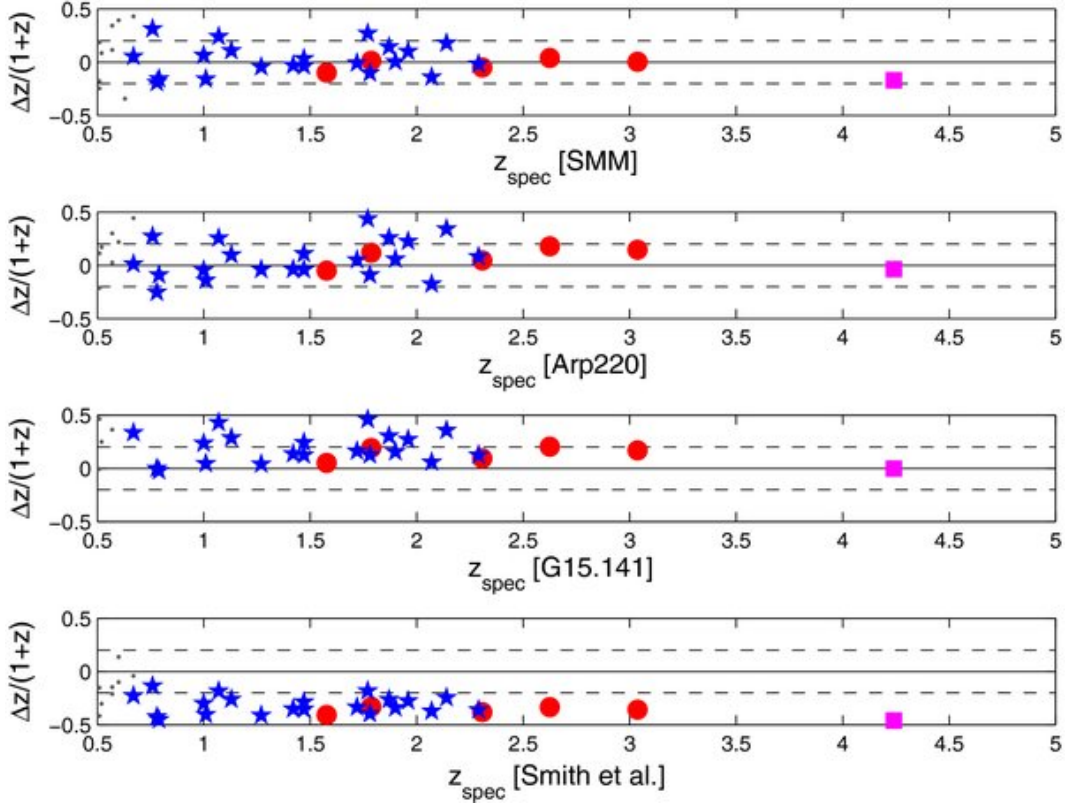


Figura 10: Comparación del *redshift* espectroscópico con el *redshift* fotométrico obtenido mediante el ajuste por mínimos tomando como referencia cuatro SEDs diferentes. Figura extraída del artículo [Lapi et al. \(2011\)](#).

Otra posible causa de error en la estimación del *redshift* proviene de la gran variedad de las galaxias con una tasas de formación estelar elevadas. Para estudiar cómo afecta la diversidad de galaxias en la estimación del *redshift* generaron una muestra de 9×10^3 galaxias a las que asignaron aleatoriamente un *redshift* comprendido entre $1 \geq z \geq 3.5$ y una SED procedente de un conjunto de 19 galaxias con una SED bien conocida, todas ellas con una $\text{SFRs} \geq 20 M_{\odot} \text{ yr}^{-1}$ y una contribución del núcleo galáctico activo al flujo en el IR-Lejano inferior al 10% (a los “flujos simulados” de $250 \mu\text{m}$, $350 \mu\text{m}$ y $500 \mu\text{m}$ se les asignaron errores de forma aleatoria a partir de errores procedentes de observaciones reales). El siguiente paso fue calcular el *redshift* fotométrico de la muestra simulada utilizando como referencia la SED de las galaxias Arp220, G15.141 y SMM J2135-0102. Después, corrigieron los valores del *redshift* fotométrico a partir de las desviaciones medias obtenidas en el estudio anterior y realizaron el histograma de la Figura 11, reconociendo que la distribución de los *redshifts* obtenidos solo se ve moderadamente afectada por la elección de la SED.

A partir de estos estudios concluyeron que, si bien los *redshifts* obtenidos son muy parecidos independientemente del modelo que tomemos como referencia (cualquiera de los considerados), la SED de la galaxia SMM J2135-0102 resulta la más adecuada para realizar los ajustes.

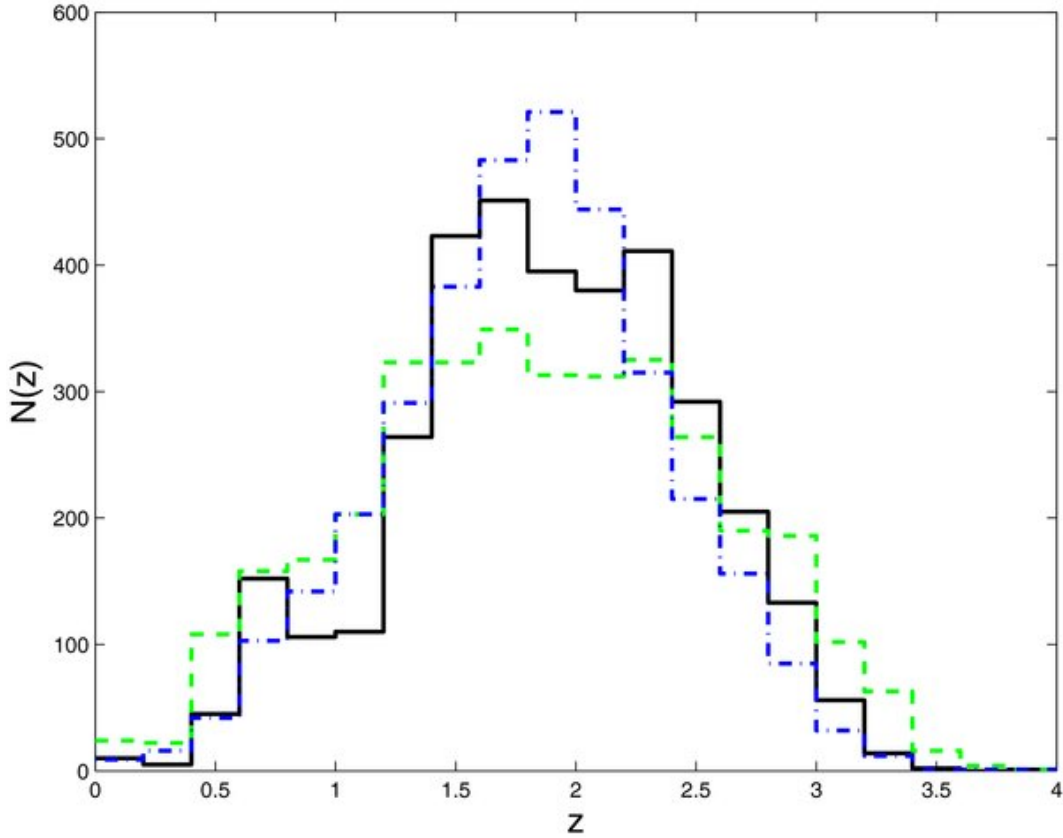


Figura 11: Distribución del *redshift* fotométrico, para las fuentes de la muestra con 9×10^3 galaxias simuladas, tomando como referencia la SED SMM J2135-0102 (curva negra continua), tomando como referencia la SED de la galaxia Arp 220 (curva discontinua verde) y la SED de la galaxia G15.141 (curva discontinua azul). La figura pertenece al artículo [Lapi et al. \(2011\)](#).

3.1.1 La galaxia SMM J2135-0102.

La galaxia SMM J2135-0102 ha sido cuidadosamente estudiada y se dispone de información detallada sobre ella. Como nos explican en [Swinbank et al. \(2010\)](#) se trata de una galaxia que muestra un *redshift* $z = 2.3259 \pm 0.0001$ y que ha sido gravitatoriamente magnificada un factor $\mu = 32.5 \pm 4.5$ por un grupo de galaxias con $z=0.325$. Es una galaxia especialmente brillante en el infrarrojo, con un flujo $S_{870 \mu\text{m}} = 106.0 \pm 7.0 \text{ mJy}$. Su tasa de formación estelar es $\text{SFR} = 210 \pm 50 M_{\odot} \text{ yr}^{-1}$ y se estima que la cantidad de materia bariónica es de $M_{\text{bar}} = (4 \pm 2) \times 10^{10} M_{\odot}$ siendo $\sim 75\%$ masa estelar y el resto gas y polvo.

En la Figura 12 se muestra la representación de la radiancia espectral (normalizada a 5570 \AA) S_{λ} de la galaxia a partir de la interpolación lineal de los puntos del fichero que utilizamos como referencia. El máximo de emisión en el IR-lejano puede observarse sobre los $10^6 \text{ \AA} \equiv 100 \mu\text{m}$ (el máximo se aprecia mejor en la representación S_{ν} , Figura 13). Esta no es una curva obtenida de forma completamente experimental; se parte de un conjunto de puntos reducido (del orden de una decena) y después a partir de modelos teóricos²⁸, se obtienen el resto de puntos de la curva que vemos. La zona $\lambda \sim 50 - 500 \mu\text{m}$ se modela a partir de las ecuaciones del cuerpo gris. En realidad, el hecho de que los valores de los *redshifts* obtenidos solo se vean moderadamente afectados por la elección de la SED, se debe en gran medida a que la información de la que se dispone de todas las galaxias consideradas como posible referencia en la Sección 3.1 es limitada en esa zona del espectro y cuando se modela la SED de cada una de ellas el resultado es parecido en todos los casos.

²⁸En [González-Nuevo et al. \(2012\)](#) y [Lapi et al. \(2011\)](#) se indica que las SEDs que ellos utilizan han sido modeladas utilizando el código GRASIL.

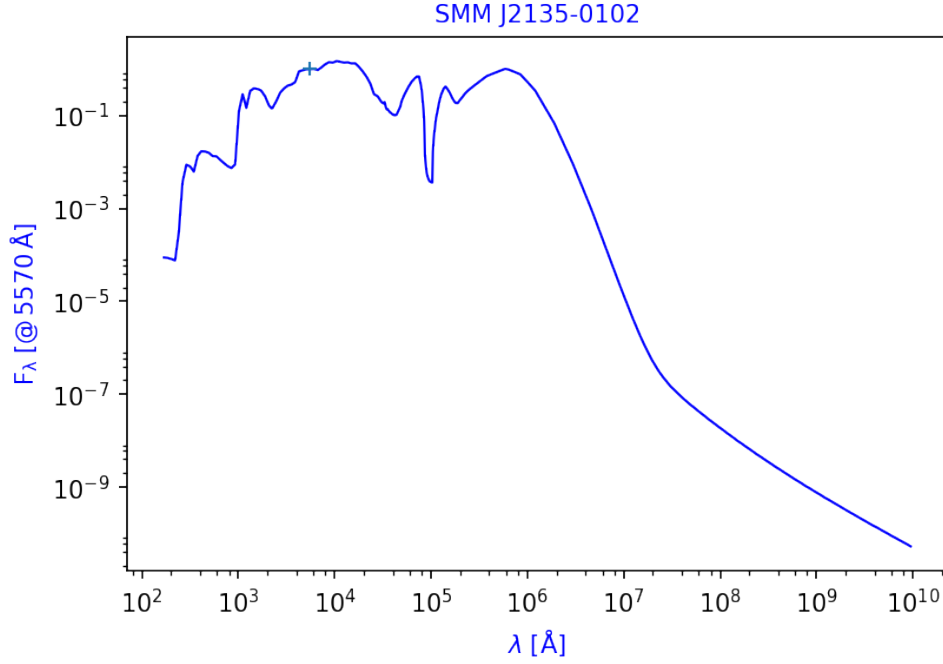


Figura 12: Representación en escala logarítmica del flujo F_λ de la galaxia SMM J2135-0102 normalizado a 5570 Å. El punto en el que se encuentra el símbolo + indica el punto de normalización.

3.2 Algoritmo

El *redshift* se obtiene mediante un ajuste de mínimos χ^2 de la SED normalizada de la galaxia SMM J2135-0102 a los tres puntos experimentales proporcionados por el instrumento SPIRE. La SED de la galaxia modelo se obtiene a partir de un fichero con dos columnas; la primera contiene los valores de la longitud de onda en Å, la segunda los valores la densidad espectral de flujo²⁹ F_λ , normalizado a 5570 Å. El flujo expresado en unidades de F_λ tiene dimensiones de $[ML^{-1}T^{-3}]$, sin embargo las medidas del SPIRE se encuentran en Jy, que es una unidad de F_ν con dimensiones de $[MT^{-2}]$. Para pasar de una unidad a la otra es suficiente tener en cuenta la relación $|F_\lambda d\lambda| = |F_\nu d\nu|$. Los valores del flujo espectral que no se encuentran en el fichero se obtienen mediante interpolación lineal.

Para realizar el ajuste se tomarán dos parámetros, que denominaremos 'C' y 'K'. La curva de ajuste se obtiene multiplicando los valores del flujo por 'C' y los valores de la longitud de onda por 'K'. En el sentido matemático, el ajuste consiste en una dilatación (o contracción) de la SED de referencia en la dirección de cada uno de los ejes de coordenadas.

En el sentido físico la dilatación en el eje y podría interpretarse como un factor que indica cómo de luminosa es la galaxia considerada respecto de la galaxia de referencia. Sin embargo, la curva de referencia que se ha utilizado, está normalizada para un valor concreto de λ por lo que no conocemos cuáles son las unidades físicas. Nos serviría para comparar magnitudes relativas entre los objetos del catálogo. La dilatación en λ se corresponde con un desplazamiento al rojo del espectro electromagnético con respecto a la SED de referencia. Para entender cómo se relaciona el parámetro de ajuste 'K' con el *redshift*, z , podemos verlo del siguiente modo: Dado que la SED de referencia no tiene *redshift* puesto que se supone que es la SED de la galaxia SMM J2135-0102 en un sistema en reposo (también habiendo corrigiendo el *redshift* debido al corrimiento al rojo

²⁹La *SWIRE Template Library* contiene la SED de 25 galaxias que se pueden utilizar como modelo: http://www.iasf-milano.inaf.it/~polletta/templates/swire_templates.html. La SED de la galaxia SMM J2135-0102 ("The Cosmic Eyelash") que se utiliza en este trabajo, no se encuentra en esa librería, pero sigue la misma normalización.

cosmológico), esta curva nos daría la longitud de onda de emisión, λ_e . La longitud de onda que observada³⁰ se corresponderá con la obtenida a partir del ajuste, es decir, $\lambda_o = \lambda_e \times K$, por tanto, utilizando la definición general de corrimiento al rojo,

$$z = \frac{\lambda_o - \lambda_e}{\lambda_e} = \frac{\lambda_o - \frac{\lambda_o}{K}}{\frac{\lambda_o}{K}} = \frac{(\lambda_o \times K) - \lambda_o}{\lambda_o} = K - 1.$$

La función del programa escrito en **Python** hace uso de esta relación para calcular z .

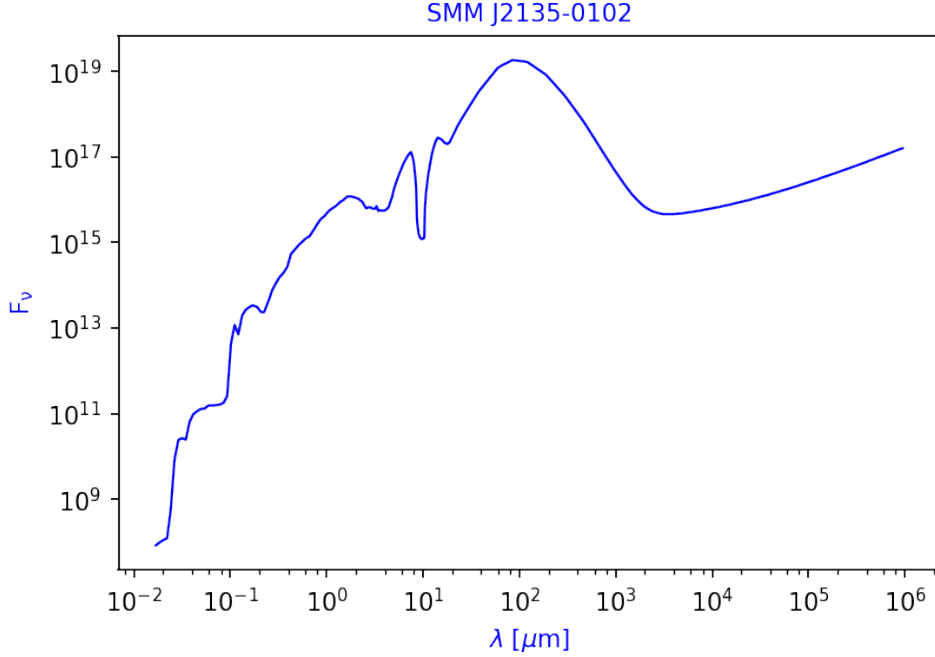


Figura 13: SED de la galaxia SMM J2135-0102. A diferencia de la Figura 12, en este caso se está representando F_ν en vez de F_λ . Esta curva es la que se utilizará como referencia para los ajustes. La zona $\lambda \sim 50 - 500 \mu\text{m}$ ha sido modelada como suma de dos cuerpos grises con temperaturas $T_d \approx 30 \text{ K}$ y $T_d \approx 60 \text{ K}$ y unos índices de emisividad para el polvo de $\beta = 1.7$ y $\beta = 2$ respectivamente.

³⁰El nombre desplazamiento al rojo sugiere inmediatamente la posibilidad de realizar un ajuste mediante un “desplazamiento” de la SED de referencia sobre el eje x , sin embargo esto carece de sentido físico. Si en vez de multiplicar por ‘K’ consideramos una traslación del tipo $\lambda_o = \lambda_e + K$ y sustituimos en la Ecuación 2 obtendremos

$$z = \frac{\lambda_o - \lambda_e}{\lambda_e} = \frac{\lambda_o - (\lambda_o + K)}{\lambda_o + K}.$$

Esto no puede ser, porque obtendríamos un *redshift* diferente dependiendo de qué valor de λ estuviésemos midiendo.

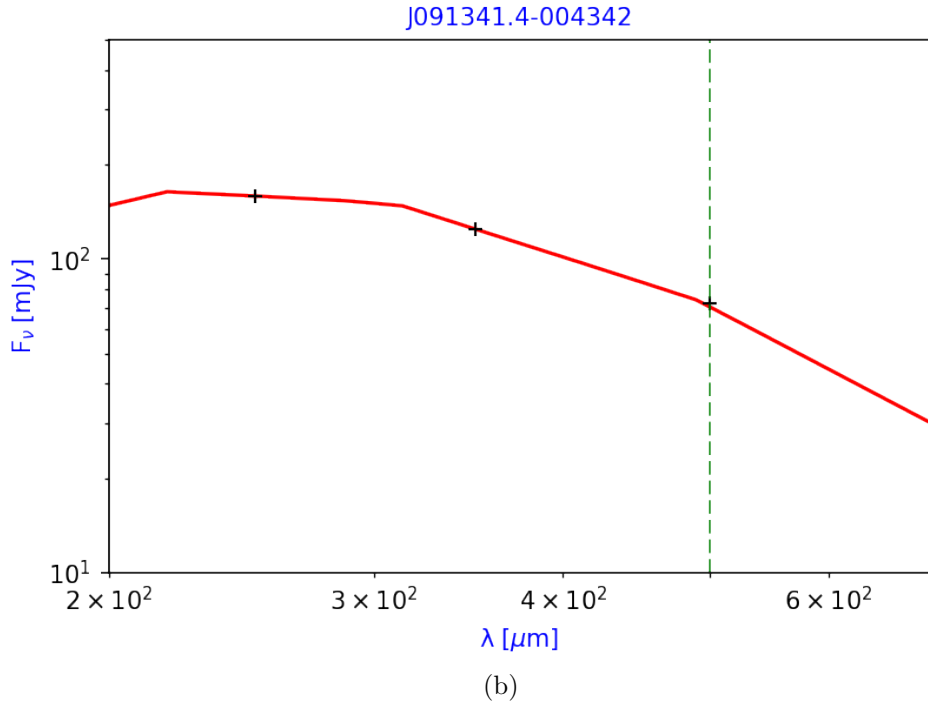
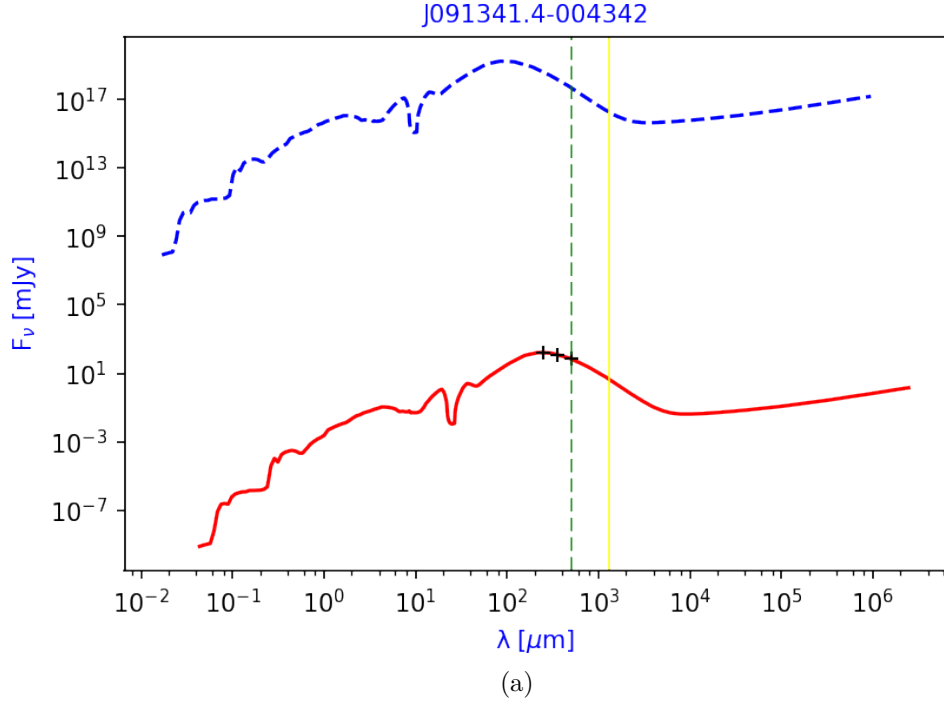


Figura 14: Ajuste por mínimos cuadrados de la SED de la galaxia SMM J2135-0102 para el objeto J091341.4-004342 del catálogo *H*-ATLAS. En la figura 14a, la curva discontinua es la curva mostrada en la Figura 13. La curva continua es la curva del ajuste, que se encuentra sobre los tres puntos experimentales de un objeto, que vienen representados por el símbolo '+'. Al utilizar la representación logarítmica, en cada uno de los ejes, la curva roja aparenta ser un desplazamiento de la curva azul. Las líneas verticales se utilizan para visualizar el desplazamiento horizontal relacionado con el desplazamiento al rojo z . En el caso del eje x el desplazamiento será $\log_{10}(z + 1)$. En la figura 14b se muestra con más detalle la zona de encuentro de la curva de ajuste con los puntos experimentales. Se aprecia perfectamente que la curva roja se compone de segmentos unidos, debido a que solo conocemos un conjunto de ~ 1000 valores de la SED el resto se obtiene mediante interpolación lineal a partir de estos.

3.3 Confrontación del método

Cabe señalar que en este trabajo no se ha probado que el método descrito en esta sección proporcione ajustes válidos para obtener el *redshift* fotométrico de las ETGs. Para hacer eso, deberíamos disponer de una muestra suficientemente grande de fuentes con *redshift* espectroscópico para poder comparar los valores obtenidos con una referencia fiable. Aunque no disponemos de esa información, en el artículo [González-Nuevo et al. \(2012\)](#) se ha publicado una tabla con el valor del *redshift* de 64 galaxias que ellos calcularon utilizando su propio programa. En la Figura 10 se ha hecho un ajuste para comparar sus resultados con los nuestros.

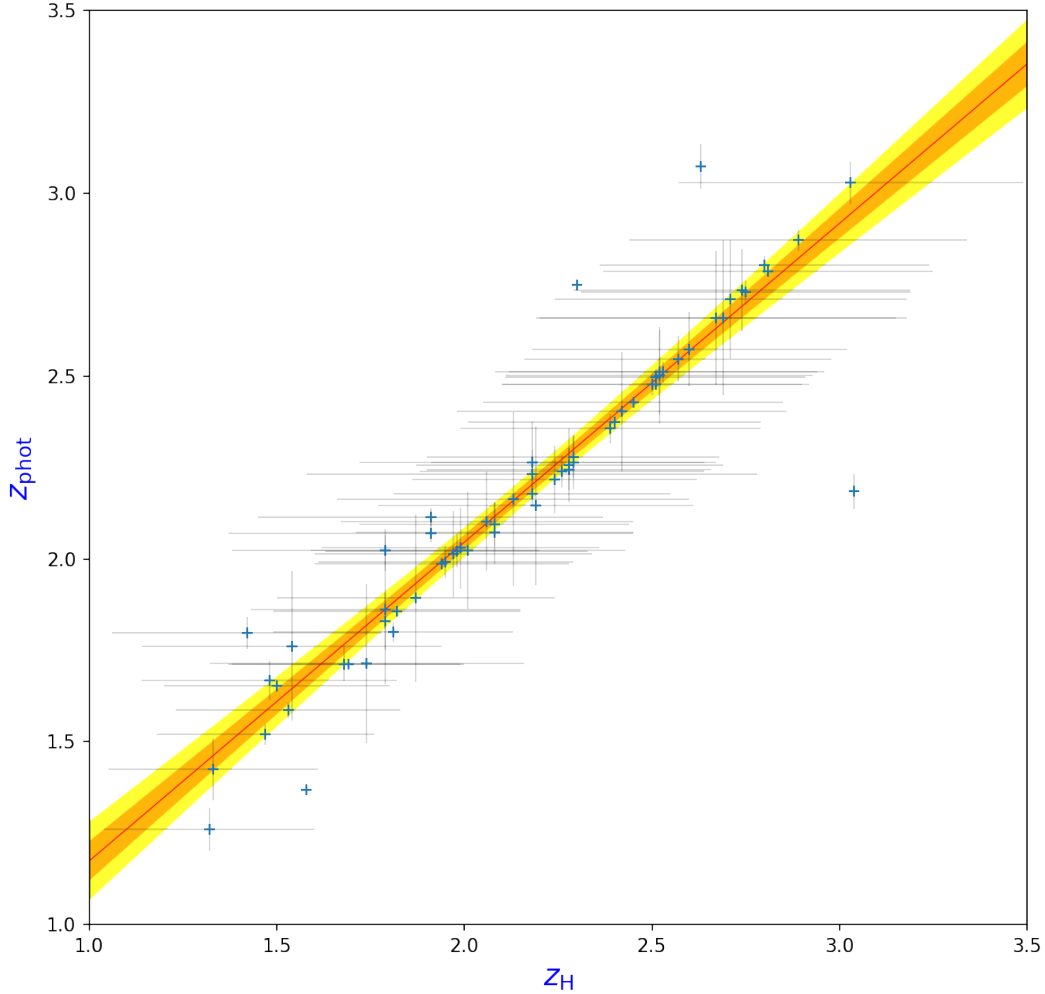


Figura 15: Comparación del *redshift* fotométrico obtenido utilizando el algoritmo descrito en la Sección 3, z_{phot} , con los valores del *redshift* fotométrico estimado por [González-Nuevo et al. \(2012\)](#), z_H . La línea roja representa el ajuste lineal, $z_{\text{phot}} = (0.87 \pm 0.04) \times z_H + (0.3 \pm 0.1)$. La zona anaranjada representa la zona con una confianza $\sigma < 1$ y la amarillenta $\sigma < 2$. Las barras de error no se han tenido en cuenta para realizar el ajuste. Las barras de error horizontales se corresponden los los valores asignados a las medidas que aparecen en el artículo, las barras de error verticales se obtienen a partir de la varianza del ajuste proporcionado por el programa que aparece en la Sección C.8 (Los valores utilizados para realizar la gráfica se encuentran en la Tabla 2).

La tabla del artículo no solo proporciona los valores del *redshift* fotométrico que ellos obtuvieron, también los valores de la desviación estándar que ellos asignaron a estas medidas. Al no disponer de una muestra de referencia con la que realizar estudios estadísticos, tampoco disponemos de los recursos suficientes para asignar valores a σ^z . Para asignar los valores de σ^z a los

valores z_{phot} obtenidos mediante nuestro método, se ha realizado un ajuste similar al mostrado en la Figura 8, partiendo de las desviaciones estándar asignadas a los desplazamientos al rojo del ajuste realizado en el artículo (Figura 16).

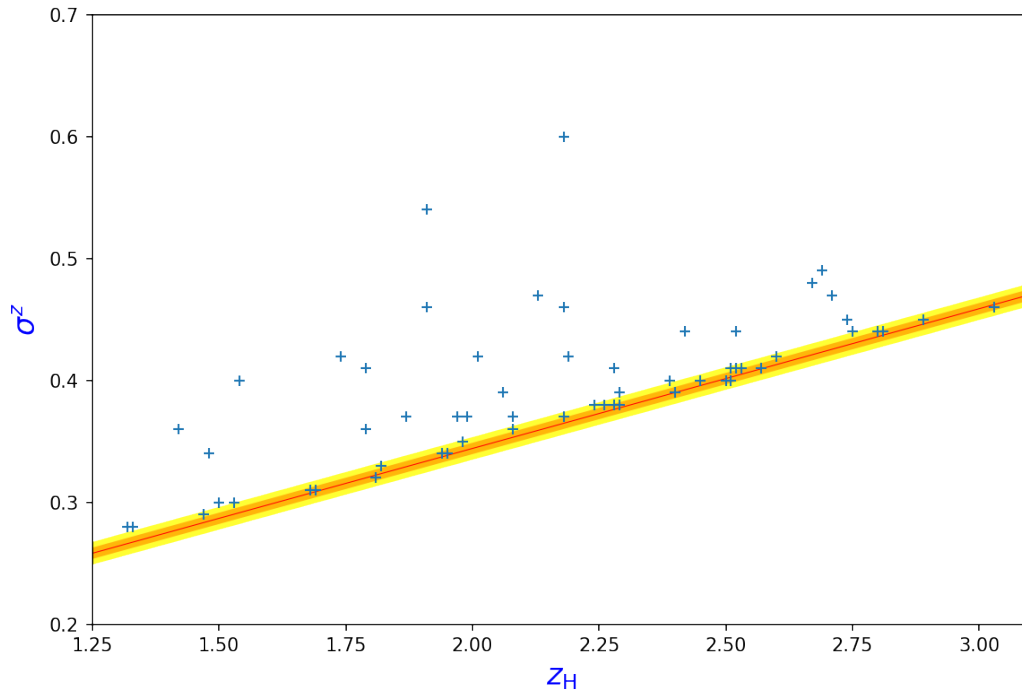


Figura 16: Representación de la desviación estándar del *redshift*, σ^z en función de z_H . Los valores de los puntos representados se encuentran en la Tabla 2. La recta se ha obtenido mediante un ajuste lineal de la forma $\sigma^z = a \times (1 + z)$ con a como parámetro de libre. El parámetro resultante del ajuste ha sido $a = 0.115 \pm 0.005$.

4 Método probabilístico de Cross-Identificación

Los métodos de cross-identificación son herramientas que proporcionan criterios estadísticos para determinar cuando observaciones pertenecientes a catálogos diferentes pertenecen a un mismo objeto astronómico o a varios. Son especialmente útiles porque facilitan el estudio de los objetos astronómicos en múltiples longitudes de onda; dado que no es posible estudiar un objeto en múltiples longitudes de onda con un único instrumento, se hace necesario acudir a algoritmos que automaticen la tarea de identificar las observaciones de un conjunto de objetos en distintos cartografiados. Cuando los cartografiados cuentan con cientos de miles de observaciones esta tarea resulta imposible de realizar de otro modo.

El criterio de cross-identificación que nosotros proponemos se sirve dos factores de Bayes; un factor de Bayes posicional y otro fotométrico propuestos por Tamás Budavári y Alexander S. Szalay. Partiremos de dos hipótesis mutuamente excluyentes; en adelante H_1 es la hipótesis de que los emparejados considerados están formados por observaciones de un mismo objeto astronómico y H_2 la hipótesis de que se trata de observaciones pertenecientes a dos objetos diferentes. El factor de Bayes que nos permitirá discernir entre ambas hipótesis será,

$$B_{12} = B_{12}^p \times B_{12}^z \quad (20)$$

llamado factor de Bayes conjunto, obtenido como producto del factor de Bayes posicional B_{12}^p y el factor de Bayes fotométrico B_{12}^z . Los objetos cuyo valor $B_{12} > 1$ son las contrapartidas formadas por observaciones de un mismo objeto en los dos catálogos. Para estar seguros de que esto es así, es conveniente que el factor de Bayes posicional sea mucho mayor que 1 (ver criterio de Harold Jeffreys: Tabla 1)

El criterio de cross-identificación será tanto mejor cuanto mayor sea la resolución espacial de los catálogos utilizados y menores sean los errores de la medida del *redshift*. Todas las observaciones pertenecientes a un mismo catálogo tienen el mismo error posicional (para los catálogos considerados). Esto va dar lugar a una distancia angular ~ 50 arcsec por debajo de la cual dos observaciones son siempre consideradas como pertenecientes a un mismo objeto teniendo en cuenta únicamente el factor de Bayes posicional (este límite lo marca el valor $B_{12}^p = 1$, ver Figura 17). El caso del criterio fotométrico es diferente y no existirá algo como una diferencia umbral entre *redshifts*; las medidas del *redshift* de cada observación en cada uno de los catálogos tiene un error diferente, por lo cual la casuística es mucho más variada (Figura 18).

La representación de los factores de Bayes de los emparejamientos encontrados que se muestra a continuación es importante para entender posteriormente cuales son los límites de nuestro método para la identificación de SLGs teniendo en cuenta las desviaciones estándar de las medidas de las que disponemos. Las dos secciones que siguientes se dedican a describir cómo obtuvieron los autores citados anteriormente el factor de Bayes posicional y fotométrico.

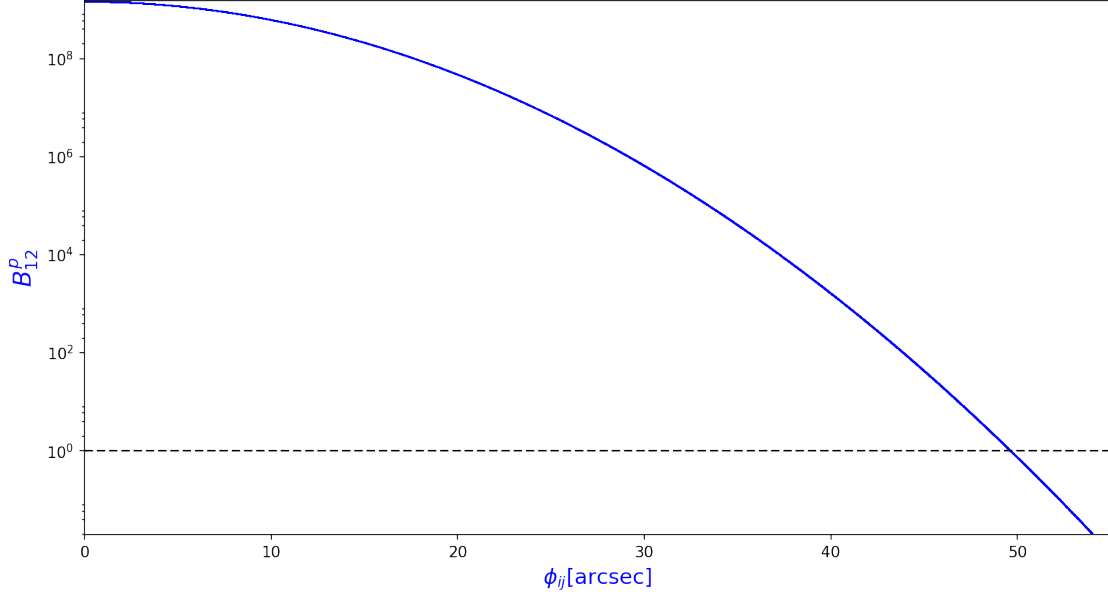


Figura 17: Representación del factor de Bayes posicional en función de la distancia angular para las contrapartidas formadas por observaciones pertenecientes a *H*-ATLAS que se encuentran a una distancia angular inferior a 54 arcsec de otra perteneciente a GAMA. La línea horizontal discontinua separa las contrapartidas con valores $B_{12}^p < 1$ y $B_{12}^p > 1$. Considerando los valores de los errores instrumentales del catálogo GAMA, $\sigma_g^p \simeq 0.297$ arcsec y del catálogo *H*-ATLAS, $\sigma_h^p \simeq 7.63$ arcsec, el valor máximo que puede tomar $B_{12}^p = 9 \times 10^9$ y la separación angular para la cual $B_{12}^p = 1$ es $\phi_{12} \sim 49.64$ arcsec.

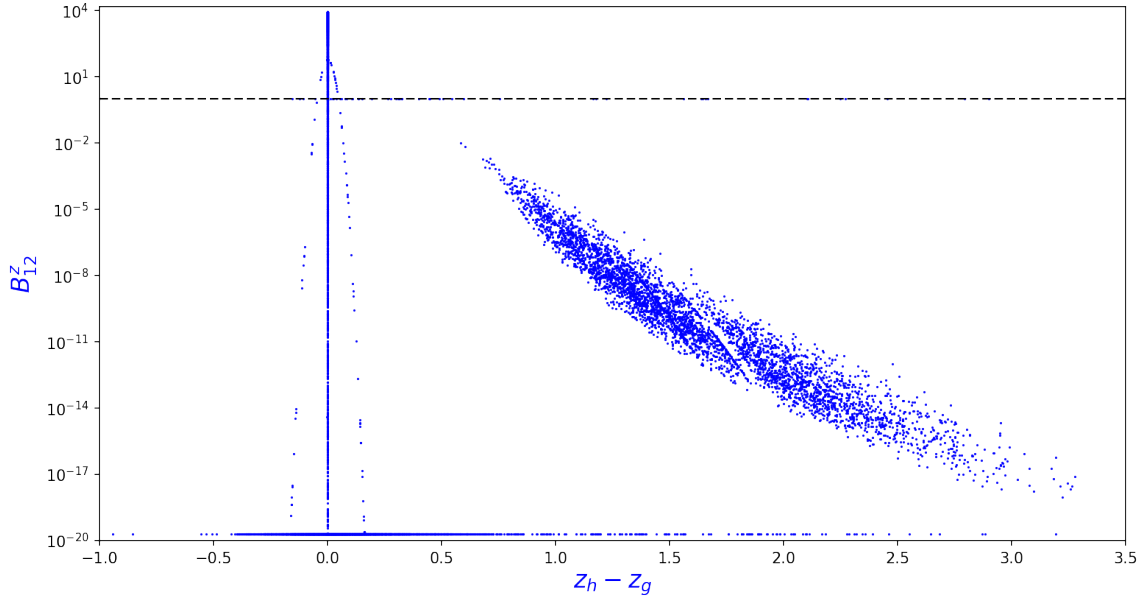


Figura 18: Representación del factor de Bayes fotométrico en función de la diferencia entre los *redshift* de los catálogos *H*-ATLAS y GAMA, $z_h - z_g$, para todas las contrapartidas formadas por observaciones de *H*-ATLAS que se encuentran a una distancia angular inferior a 54 arcsec de otra de GAMA tomando $z_{max} = 3.5$. La línea discontinua horizontal marca el límite entre los puntos con valores $B_{12}^z < 1$ y $B_{12}^z > 1$. En este caso, a diferencia del factor de Bayes posicional, cada medida del *redshift* tiene su propio valor del error, por lo que al realizar la representación los puntos aparecen como una nube de puntos. El valor máximo que puede tomar el factor de Bayes espectroscópico es $B_{12}^z = 8462$ considerando $z_h - z_g = 0$ y $\sigma^z = 1.1 \times 10^{-4}$ para las dos observaciones que forman el pareo.

4.1 Factor de Bayes posicional propuesto por Budavári & Szalay

Estos autores modelizan la posición verdadera de un objeto sobre la esfera celeste mediante un vector tridimensional unitario \mathbf{m} y las posiciones observadas del mismo a partir del conjunto de vectores unitarios $D = \{\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_n\}$. Debido a que hay un error instrumental, no podemos afirmar que la medida sobre la posición de un objeto, \mathbf{x}_i , se corresponda exactamente con su posición verdadera \mathbf{m} ; por ese motivo se describe de forma estadística su posición a través de una función densidad de probabilidad (PDF), $p(\mathbf{x}|\mathbf{m}, H_k)$, que nos proporciona la probabilidad de que la verdadera posición del objeto sea \mathbf{m} partiendo de la medida \mathbf{x} . Estos autores hacen la propuesta de modelizar la PDF a través de una función gaussiana tridimensional normalizada³¹,

$$p(\mathbf{x}|\mathbf{m}, H_k) = N(\mathbf{x}|\mathbf{m}) = \frac{w\delta(|\mathbf{x}| - 1)}{4\pi \sinh w} \exp(w\mathbf{x} \cdot \mathbf{m}),$$

con $w = 1/\sigma^2$ y la *probabilidad a priori* $P(\mathbf{m}|H_k)$, como una delta de Dirac³²,

$$p(\mathbf{m}|H_k) = \frac{1}{4\pi} \delta(|\mathbf{m}| - 1). \quad (21)$$

El factor de Bayes se obtiene a partir del cociente de las funciones verosimilitud (Definición 1.2),

$$B_{12}^p = \frac{p(D|H_1)}{p(D|H_2)}. \quad (22)$$

En el caso de que se cumpla la hipótesis H_1 , el conjunto de medidas D , hará referencia a una única posición \mathbf{m} . Debido a que cada medida \mathbf{x}_i tiene su propia PDF $p_i(\mathbf{x}_i|\mathbf{m}, H_1)$, la PDF conjunta, $p(D|\mathbf{m}, H_1)$, se expresa como el producto de las PDFs independientes,

$$p(D|\mathbf{m}, H_1) = \prod_{i=1}^n p_i(\mathbf{x}_i|\mathbf{m}) = \prod_{i=1}^n N(\mathbf{x}_i|\mathbf{m}) = \prod_{i=1}^n \frac{w_i \delta(|\mathbf{x}_i| - 1)}{4\pi \sinh w_i} \exp(w_i \mathbf{x}_i \cdot \mathbf{m})$$

y la función verosimilitud $p(D|H_1)$ resulta ser

$$\begin{aligned} p(D|H_1) &= \int p(\mathbf{m}|H_1) p(D|\mathbf{m}, H_1) d^3m = \\ &= \int \frac{\delta(|\mathbf{m}| - 1)}{4\pi} \prod_{i=1}^n \frac{w_i \delta(|\mathbf{x}_i| - 1)}{4\pi \sinh w_i} \exp(w_i \mathbf{x}_i \cdot \mathbf{m}) d^3m = \\ &= \left[\prod_{i=1}^n \frac{w_i \delta(|\mathbf{x}_i| - 1)}{4\pi \sinh w_i} \right] \int \frac{\delta(|\mathbf{m}| - 1)}{4\pi} \exp\left(\sum_{i=1}^n w_i \mathbf{x}_i \cdot \mathbf{m}\right) d^3m. \end{aligned}$$

Introduciendo

$$w\mathbf{x} = \sum_{i=1}^n w_i \mathbf{x}_i$$

y multiplicando y dividiendo por $\frac{\sinh w}{w}$, tenemos que

³¹En la Sección 4.1 y la Sección 4.2 se ha simplificado la notación con respecto al resto del documento para evitar posible confusiones con las potencias. En esta sección, σ es equivalente a lo que en el resto de la memoria denominamos σ^z y en la siguiente, σ equivale a σ^p .

³²Se trata de una distribución no informativa, es decir, la probabilidad se reparte por igual en todo el espacio paramétrico.

$$p(D|H_1) = \left[\frac{\sinh w}{w} \prod_{i=1}^n \frac{w_i}{4\pi \sinh w_i} \delta(|\mathbf{x}_i| - 1) \right] \int \frac{w \delta(|\mathbf{m}| - 1)}{4\pi \sinh w} \exp(w \mathbf{x} \cdot \mathbf{m}) d^3 m = \frac{\sinh w}{w} \prod_{i=1}^n \frac{w_i}{\sinh w_i} \frac{\delta(|\mathbf{x}_i| - 1)}{4\pi}.$$

Por otra parte, la hipótesis alternativa H_2 , representa la hipótesis de que las medidas realizadas son pertenecientes a los objetos diferentes, por tanto el conjunto de medidas $D = \{\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_n\}$ hace referencia al conjunto de posiciones verdaderas $\{\mathbf{m}_1, \mathbf{m}_2 \dots \mathbf{m}_n\}$. y la función verosimilitud de la hipótesis H_2 ,

$$p(D|H_2) = \prod_{i=1}^n \left[\int p(\mathbf{m}_i|H_2) p_i(\mathbf{x}_i|\mathbf{m}_i, H_2) d^3 m_i \right] = \prod_{i=1}^n \int \frac{\delta(|\mathbf{m}_i| - 1)}{4\pi} \frac{w_i \delta(|\mathbf{x}_i| - 1)}{4\pi \sinh w_i} \exp(w_i \mathbf{x}_i \mathbf{m}_i) d^3 m_i = \prod_{i=1}^n \frac{\delta(|\mathbf{x}_i| - 1)}{4\pi}.$$

Al sustituir en la Ecuación 22 obtenemos,

$$B_{12}^p = \frac{p(D|H_1)}{p(D|H_2)} = \frac{\sinh w}{w} \prod_{i=1}^n \frac{w_i}{\sinh w_i}$$

que se aproxima a

$$B_{12}^p = 2^{(n-1)} \frac{\prod_{i=1}^n w_i}{\sum_{i=1}^n w_i} \exp\left(-\frac{\sum_{i < j} w_i w_j \phi_{ij}^2}{2 \sum_{i=1}^n w_i}\right)$$

y para dos catálogos astronómicos, se reduce a

$$B_{12}^p = \frac{2}{\sigma_1^2 + \sigma_2^2} \exp\left(-\frac{\phi_{12}^2}{2(\sigma_1^2 + \sigma_2^2)}\right) \quad (23)$$

como expresión final para el factor de Bayes posicional.

4.2 Factor de Bayes fotométrico propuesto por Budavári

A diferencia del factor de Bayes posicional en que se resolvía el problema para un caso general, aquí se va resolver el caso en el que se dispone de dos catálogos. Ahora, el valor verdadero del *redshift* de un objeto viene representado por τ y el conjunto de medidas se reduce a $D = \{z_1, z_2\}$. De nuevo, debido a que existe una incertidumbre asociada a estas medidas, el valor medido no se corresponde exactamente con el valor real por lo que hay que proponer una PDF. En este caso se elige la función gaussiana unidimensional normalizada de la forma

$$p(z|\tau, H_k) = N(z|\tau) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(z - \tau)^2}{2\sigma^2}\right)$$

y una *probabilidad a priori* constante entre 0 y un valor máximo z_{max} ,

$$p(\tau|H_k) = p_0,$$

cuyo valor viene dado por la condición de normalización

$$\int_0^{z_{max}} p(\tau|H_k) d\tau = 1 \implies p_0 = \frac{1}{z_{max}}.$$

En caso de que ambas medidas se correspondan a único valor, la densidad de probabilidad conjunta se expresa como

$$p(D|\tau, H_1) = \prod_{i=1}^2 p_i(z_i|\tau, H_1) = \prod_{i=1}^2 N(z_i|\tau) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{(\tau - z_1)^2}{2\sigma_1^2} - \frac{(\tau - z_2)^2}{2\sigma_2^2}\right)$$

y la función verosimilitud para la hipótesis H_1 será

$$p(D|H_1) = \int_0^{z_{max}} p(\tau|H_1)p(D|\tau, H_1)d\tau = \frac{p_0}{2\sqrt{2\pi}\sqrt{\sigma_1^2+\sigma_2^2}} \exp\left(-\frac{(z_1-z_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right) \left[\operatorname{erf}\left(\frac{\sigma_1^2 z_2 + \sigma_2^2 z_1}{\sqrt{2}\sigma_1\sigma_2\sqrt{\sigma_1^2+\sigma_2^2}}\right) - \operatorname{erf}\left(\frac{\sigma_1^2(z_2-z_{max}) + \sigma_2^2(z_1-z_{max})}{\sqrt{2}\sigma_1\sigma_2\sqrt{\sigma_1^2+\sigma_2^2}}\right) \right].$$

Por otro lado, si las dos medidas pertenecen a dos objetos diferentes, la función verosimilitud se obtiene a partir de

$$p(D|H_2) = \prod_{i=1}^2 \left[\int_0^{z_{max}} p(\tau_i|H_2)p_i(z_i|\tau_i, H_2) d\tau_i \right] = \frac{p_0^2}{4} \left[\operatorname{erf}\left(\frac{z_1}{\sqrt{2}\sigma_1}\right) - \operatorname{erf}\left(\frac{z_1 - z_{max}}{\sqrt{2}\sigma_1}\right) \right] \left[\operatorname{erf}\left(\frac{z_2}{\sqrt{2}\sigma_2}\right) - \operatorname{erf}\left(\frac{z_2 - z_{max}}{\sqrt{2}\sigma_2}\right) \right].$$

Sustituyendo las expresiones obtenidas para las funciones verosimilitud en la definición del factor de Bayes 1.2, tenemos la expresión que se va a utilizar en este trabajo,

$$B_{12}^z = \frac{\sqrt{2} z_{max} \exp\left(-\frac{(z_1-z_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right) \left[\operatorname{erf}\left(\frac{\sigma_1^2 z_2 + \sigma_2^2 z_1}{\sqrt{2}\sigma_1\sigma_2\sqrt{\sigma_1^2+\sigma_2^2}}\right) - \operatorname{erf}\left(\frac{\sigma_1^2(z_2-z_{max}) + \sigma_2^2(z_1-z_{max})}{\sqrt{2}\sigma_1\sigma_2\sqrt{\sigma_1^2+\sigma_2^2}}\right) \right]}{\sqrt{\pi(\sigma_1^2 + \sigma_2^2)} \left[\operatorname{erf}\left(\frac{z_1}{\sqrt{2}\sigma_1}\right) - \operatorname{erf}\left(\frac{z_1 - z_{max}}{\sqrt{2}\sigma_1}\right) \right] \left[\operatorname{erf}\left(\frac{z_2}{\sqrt{2}\sigma_2}\right) - \operatorname{erf}\left(\frac{z_2 - z_{max}}{\sqrt{2}\sigma_2}\right) \right]}. \quad (24)$$

para el factor de Bayes fotométrico³³. Si consideramos que las medidas del *redshift* de uno de los catálogos no tienen error (por ejemplo si la medida es espectroscópica), $\sigma_2 = 0$ y la Ecuación 24 se reduce a

$$B_{12}^z = \frac{\sqrt{\frac{2}{\pi}} z_{max} \exp\left(-\frac{(z_1-z_2)^2}{2\sigma_1^2}\right)}{\sigma_1 \left(\operatorname{erf}\left(\frac{z_1}{\sqrt{2}\sigma_1}\right) - \operatorname{erf}\left(\frac{z_1 - z_{max}}{\sqrt{2}\sigma_1}\right) \right)}.$$

En el límite en el que $z_{max} \gg z_1 \gg \sigma$ el paréntesis del denominador que contiene las funciones error tiende a $1 - (-1) = 2$ y se obtiene la ecuación:

$$B_{12}^z = \frac{z_{max}}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(z_1 - z_2)^2}{2\sigma_1^2}\right)$$

que encontramos en [Garcia \(2015\)](#).

³³Cuando realicemos los cálculos vamos a asumir que $z_1 \geq 0$, $z_2 \geq 0$, $\sigma_1 > 0$, $\sigma_2 > 0$, $z_{max} > 0$ y $z_{max} > z_1 \wedge z_2$

5 Identificación sistemas candidatos a lente gravitatoria

En esta sección daremos la explicación de nuestra propuesta para la búsqueda de candidatos a SLGs a partir de las medidas del instrumento SPIRE. Además se dará una breve descripción de dos propuestas anteriores basadas en la selección de uno o varios límites de densidades de flujo espectral a longitudes de onda concretas por encima de los cuales se espera que las galaxias submilimétricas hayan sido lensadas. Estos dos métodos tienen bastante incertidumbre pero se consideran útiles para estudios estadísticos (González-Nuevo et al., 2012).

5.1 Criterio propuesto por Negrello et al.

Uno de los métodos propuestos para la búsqueda de HALOS (*Herschel*-ATLAS *Lensed Objects Selection*) es la selección de aquellos objetos con un flujo a $500\ \mu\text{m}$ superior a 100 mJy. Estos autores se basan en modelos teóricos para calcular la densidad de flujo límite, a partir de la cual esperan maximizar la proporción de SLGs. Sostienen que la densidad superficial de galaxias submilimétricas que no han sido magnificadas es próxima a cero cuando $S_{500} > 100\ \mu\text{m}$. El brillo aparente, en el infrarrojo lejano de una ETG con $S_{500} > 100\ \text{mJy}$ es $L_{\text{FIR}} > 3 \times 10^{13}\ L_{\odot}$, lo cual, teniendo en cuenta un factor de magnificación $\mu \sim 10$ debido a la lente gravitatoria indica una $\text{SFR} > 500\ M_{\odot}\ \text{yr}^{-1}$. Esto contrasta con otras observaciones realizadas con telescopios de 8 – 10 m de abertura montados en tierra, que indican una tasa de formación estelar mucho menor para casos excepcionalmente altos ($\sim 100 - 200\ M_{\odot}\ \text{yr}^{-1}$).

Las predicciones indican una densidad de SLGs que cumplen esta condición de $\sim 0.3\ \text{deg}^{-2}$, por lo que el área de $550\ \text{deg}^2$ cubierto por PACS y SPIRE debería proporcionar en torno a 150-200 SLGs (González-Nuevo et al., 2012). La pureza obtenida mediante este método se estima próxima al 100 %.

5.2 Criterio propuesto por González-Nuevo et al.

Este método propone que todos aquellos objetos que cumplen las condiciones de $S_{350} \geq 85\ \text{mJy}$, $S_{250} \geq 35\ \text{mJy}$, $S_{350}/S_{250} > 0.6$, $S_{500}/S_{250} > 0.4$ han sido magnificados gravitatoriamente por un factor $\gtrsim 2$. Los argumentos en los que se basan estos autores para proponer este método son similares a los utilizados por Negrello et al. (2010) pero tienen en cuenta más cosas. Concretamente se basan en un estudio teórico de las funciones luminosidad de las galaxias submilimétricas a 100 mJy y 250 mJy llevado a cabo en Lapi et al. (2011).

En este caso encontraron 31 candidatos sobre la región de la *H*-ATLAS *Science Demonstration Phase* que cubre una región de $\sim 14.4\ \text{deg}^{-2}$ lo que supone una densidad de objetos de $\sim 1.5 - 2\ \text{deg}^{-2}$ ($\sim 4 - 6$ mayor que con el criterio anterior). Estos autores también estimaron una pureza de $\sim 72\ %$ para la muestra estudiada.

5.3 Propuesta para encontrar SLGs

El método que nosotros proponemos para la búsqueda de SLGs consiste en identificar sistemas lente gravitatoria completos. Se basa en el hecho de que las lentes gravitatorias están siempre formadas por al menos dos cuerpos; un objeto lente y un objeto fuente. Supondremos que los objetos fuente se encuentran en el catálogo *H*-ATLAS y los objetos lente en el GAMA. Esta decisión se debe a que el observatorio espacial *Herschel* ha sido diseñado con el propósito de identificar objetos con alto *redshift* y además ya existen propuestas para identificar SLGs y estudios previos para estimar el *redshift* de galaxias en formación a partir de las medidas que ha realizado. Por su parte el proyecto GAMA tiene objetivos muy diferentes a los perseguidos por *H*-ATLAS ya que pretende estudiar objetos del universo próximo, con *redshift* bajos e intermedios ($z < 1$), por lo

que consideramos que los objetos estudiados por GAMA muestran una distribución de distancia adecuada para encontrar en él nuestros candidatos a lente. Nuestro método, a diferencia de los vistos anteriormente que se aplicaban sobre toda la superficie estudiada por *H-ATLAS*, se limita a la zona de estudio común entre los proyectos GAMA y *H-ATLAS*.

Los objetos lente en su mayoría son grandes galaxias elípticas o cúmulos de galaxias (Negrello et al., 2010). Para hacer una estimación razonable de la distancia óptima a la que se produce el efecto de lente gravitatoria, consideramos en caso más simple posible. Supongamos que se ha formado un anillo de Einstein para el caso en el que la lente posee una masa puntual y una masa típica de una galaxia elíptica, entre $\sim 10^9 M_\odot$ y $\sim 10^{12} M_\odot$. El radio de Einstein depende también tanto de la distancia entre la lente y la fuente como la distancia entre la lente y el observador (Ecuación 10), sin embargo para los casos que nos ocupan (objetos fuente con alto *redshift*) la distancia entre el observador y la lente es mucho mas pequeña que la distancia entre la lente y la fuente por lo que se puede utilizar la Ecuación 11, a partir de la cual se ha realizado la representación de la Figura 19.

El radio efectivo de las galaxias espirales varía entre $\sim 20 - 30$ kpc para las galaxias gigantes y varios cientos de pc para las galaxias más pequeñas (Murdin, 2001). Como puede verse en la Figura 19 un objeto con una masa $10^{12} M_\odot$ que se encuentra a una distancia de ~ 460 Mpc muestra un $\theta_E \simeq 4 \text{ arcsec} \simeq 1.9 \times 10^{-5} \text{ rad}$, equivalente a un radio aparente de $1.9 \times 10^{-5} \times 460 \text{ Mpc} \simeq 9 \text{ kpc}$. Eso quiere decir que el anillo de Einstein quedaría dentro del radio efectivo de la galaxia elíptica gigante si esta concentrase toda su masa en su centro, lo cual no es cierto evidentemente (la masa está distribuida en el volumen que ocupa la galaxia). Este cálculo no es muy exacto pero nos está indicando que los anillos de Einstein aparecen prácticamente sobre los objetos lente al menos cuando las lentes son galaxias elípticas típicas.

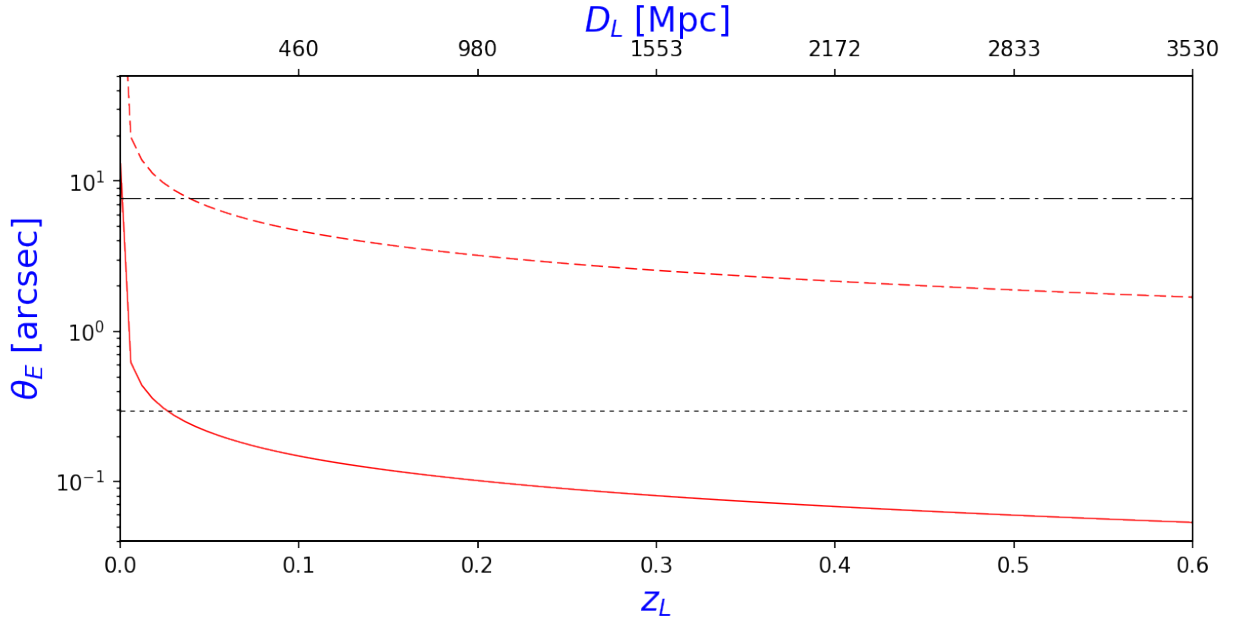


Figura 19: Variación del radio de Einstein con la distancia observador-lente (Ecuación 11) suponiendo un objeto lente puntual de masa $10^9 M_\odot$ (curva roja continua) y masa $10^{12} M_\odot$ (curva roja discontinua). Las líneas horizontales representan los límites de resolución de los catálogos *H-ATLAS* ($\sigma_h^p \sim 7.63 \text{ arcsec}$) y GAMA ($\sigma_g^p \sim 0.297 \text{ arcsec}$). El intervalo z_L cubre el rango de valores que toma el *redshift* para la gran mayoría de los objetos del catálogo GAMA (ver Figura 7). La escala que aparece sobre la parte inferior de la figura es el *redshift* equivalente a la distancia que aparece en la escala superior de la figura según el modelo Lambda-CDM³⁴. Obsérvese que la Ecuación 11 relaciona θ_E con D_L , no con z_L que es lo que se mide directamente a partir de la observación del objeto.

Nosotros no esperamos encontrar anillos de Einstein, pero los emparejados que estamos buscando forman lentes gravitatorias fuertes y sabemos que separación angular entre nuestros candidatos a lente gravitatoria debe ser de ese orden de magnitud. Teniendo en cuenta el radio efectivo de una galaxia elíptica de gran tamaño, su masa y un desplazamiento al rojo $z \sim 0.1$, cabría esperar una separación angular de ~ 17 arcsec. Dado que se trata de un caso extremo para una galaxia especialmente grande y especialmente próxima el resto de candidatos deben mostrar separaciones angulares inferiores.

La idea del método que proponemos es encontrar lentes gravitatorias en un subconjunto de emparejados que surgen al aplicar el método de cross-identificación de catálogos de galaxias descrito en la sección anterior al considerar los catálogos *H-ATLAS* y *GAMA*. Se trata de observaciones que dada su proximidad angular podrían pertenecer al mismo objeto, pero que sin embargo tienen *redshifts* diferentes (esto es equivalente a decir que se cumplen las condiciones $B_{12}^p > 1$ y $B_{12} < 1$). Como hemos visto en la Sección 4, dada la resolución angular de los catálogos utilizados, el factor de Bayes posicional considerado toma el valor 1 cuando la separación angular de dos observaciones se encuentra en ~ 49.64 arcsec. Esta distancia angular es varias veces mayor que la distancia de angular máxima que estimamos para la lente gravitatoria. Esto resulta beneficioso por una parte porque de esta manera probablemente la mayor parte de las lentes gravitatorias cuando intervienen dos objetos pertenezcan a este conjunto, pero también es un problema porque si la distancia de emparejamiento es muy grande, se incrementará también el número de contrapartidas formadas por observaciones entre objetos que no guardan ningún tipo de relación entre sí. Lo ideal sería que la distancia para la cuál $B_{12}^p = 1$ se encontrase en ~ 17 arcsec.

A modo de resumen, el procedimiento que se va a seguir en este trabajo para identificar lentes gravitatorias será el siguiente:

- Seleccionaremos todas aquellas observaciones de los catálogos *H-ATLAS* y *GAMA* de los cuales disponemos una medida razonable de su *redshift*³⁵.
- Partiendo de la selección anterior, se realizará una búsqueda exhaustiva de todas las las observaciones del catálogo *H-ATLAS* DR1 que se encuentran a una distancia inferior a los ~ 49.64 arcsec de otro de *GAMA*. En realidad, en este trabajo se ha escogido una distancia de emparejamiento de 54 arcsec y después se ha utilizado la condición de $B_{12}^p < 1$, pero esto es equivalente a elegir una distancia de emparejamiento de máxima de $\sim 49,64$ arcsec desde el principio.
- A cada emparejado, le asignaremos tres factores de Bayes: B_{12}^p , B_{12}^z y B_{12} . Seleccionaremos aquellos que cumplen las condiciones de $B_{12}^p > 1$ y $B_{12} < 1$.
- Por último seleccionaremos aquellos emparejados en los que $z_h > 1$ (para poder considerar la observación perteneciente *H-ATLAS* un objeto con alto *redshift*) y $z_h > z_g$ (ya que los objetos lente los buscamos en *GAMA* y las fuentes en *H-ATLAS*). La muestra resultante será nuestra selección de candidatos a sistemas lente gravitatoria en la que participa una fuente con un alto *redshift*.

³⁴Para realizar este cálculo se ha utilizando el paquete `astropy.cosmology` y el modelo Lambda-CDM (Lambda Cold Dark Matter) con valor de la constante de Hubble $70 \text{ km s}^{-1} \text{ Mpc}^{-1}$ y densidad de la materia no relativista en unidades de la densidad crítica $\Omega_0=0.3$.

³⁵Por medida razonable del *redshift* entendemos, una medida del *redshift* proporcionada por los catálogos o una estimación válida obtenida mediante el método descrito en la Sección 3. Obsérvese este método solo sirve para galaxias lejanas en formación. Estos objetos son abundantes a *redshifts* altos, pero no son los únicos. El método propuesto para la identificación de lentes gravitatorias no depende de si las fuentes son ETGs o no.

6 Resultados y análisis

A continuación expondremos los resultados que hemos obtenido al aplicar los criterios de selección de candidatos a SLGs sobre las regiones seleccionadas.

6.1 Candidatos según Negrello et al. y González-Nuevo et al.

Nosotros hemos utilizado los criterios de [Negrello et al. \(2010\)](#) y de [González-Nuevo et al. \(2012\)](#) para identificar SLGs en el catálogo *H-ATLAS* DR1 y hemos encontrado que hay 20 objetos que cumplen el criterio de [Negrello et al. \(2010\)](#), 147 que cumplen el criterio de [González-Nuevo et al. \(2012\)](#) y 10 que cumplen ambos criterios. Teniendo en cuenta que el catálogo cubre un área de $\sim 161 \text{deg}^2$ estimamos que hay una densidad de objetos de $\sim 0.124 \text{deg}^{-2}$ para aquellos que cumplen el primer criterio, de $\sim 0.913 \text{deg}^{-2}$ que cumplen el segundo y de $\sim 0.975 \text{deg}^{-2}$ si consideramos todos aquellos objetos que cumplen al menos uno de los criterios anteriores.

Por tanto las densidades de objetos que cumplen estos criterios en el catálogo *H-ATLAS* DR1 son inferiores a las estimaciones previas que se habían realizado. Compárese la densidad de objetos que hemos encontrado, $\sim 0.1 \text{deg}^{-2}$ y $\sim 1 \text{deg}^{-2}$ cuando las estimaciones eran $\sim 0.3 \text{deg}^{-2}$ y $\sim 1.5 - 2 \text{deg}^{-2}$ respectivamente ([González-Nuevo et al., 2012](#)).

6.2 Candidatos según el criterio bayesiano

En la zona de intersección de los catálogos *H-ATLAS* DR1 y GAMA se han encontrado 3824 contrapartidas que cumplen las condiciones para ser candidatos a sistema lente gravitatoria según el criterio bayesiano que hemos propuesto en la Sección 5. Este conjunto contrapartidas se ha dividido en dos categorías. La primera formada por aquellas observaciones que disponían de *redshift* espectroscópico en ambos casos y la segunda formada por las observaciones pertenecientes al catálogo *H-ATLAS* que tienen un *redshift* obtenido mediante el procedimiento descrito en la Sección 3. De la primera categoría se han encontrado 116 contrapartidas y de la segunda 3708. Entre las contrapartidas propuestos se han encontrado 16 en las cuales la observación de *H-ATLAS* cumple los criterios de [González-Nuevo et al. \(2012\)](#) o [Negrello et al. \(2010\)](#) para ser SLGs.

Por otra parte también se ha realizado una simulación en la que a cada observación se le ha reasignado una posición aleatoria sobre una región con un área equivalente al área del catálogo al que pertenece. Después estas dos regiones se han solapado en un área igual al área de intersección de los catálogos originales y se ha procedido a realizar un conteo de las contrapartidas exactamente de la misma forma que la que se ha llevado a cabo con las muestras originales. Esta simulación se ha repetido mil veces encontrándose 3521 ± 58 contrapartidas que cumplen los criterios para ser candidatas a sistema lente gravitatoria, de las cuales 111 ± 11 pertenecen a la primera categoría y 3410 ± 57 pertenecen a la segunda. Entre estos candidatos hay 26 ± 5 contrapartidas en las que además la observación en *H-ATLAS* cumple los criterios de [González-Nuevo et al. \(2012\)](#) o [Negrello et al. \(2010\)](#).

Estos resultados indican que hay $\sim 200 - 300$ contrapartidas que no pueden ser explicadas por una distribución aleatoria de las observaciones, sin embargo, evidencian también que nuestra muestra tiene un número muy elevado de contrapartidas formadas por observaciones que no tienen ninguna relación entre sí.

La mayor diferencia entre la muestra simulada y la real se encuentra entre los objetos de la segunda categoría con una diferencia de un 8 % de la muestra real. También es significativo que en la muestra simulada se estén encontrando más contrapartidas en las que se cumplen los criterios de [González-Nuevo et al. \(2012\)](#) o [Negrello et al. \(2010\)](#) que para el caso de la muestra real. Esto nos está indicando que los objetos seleccionados por nuestro criterio de identificación son diferentes a los seleccionados por estos dos criterios para la identificación de SLGs.

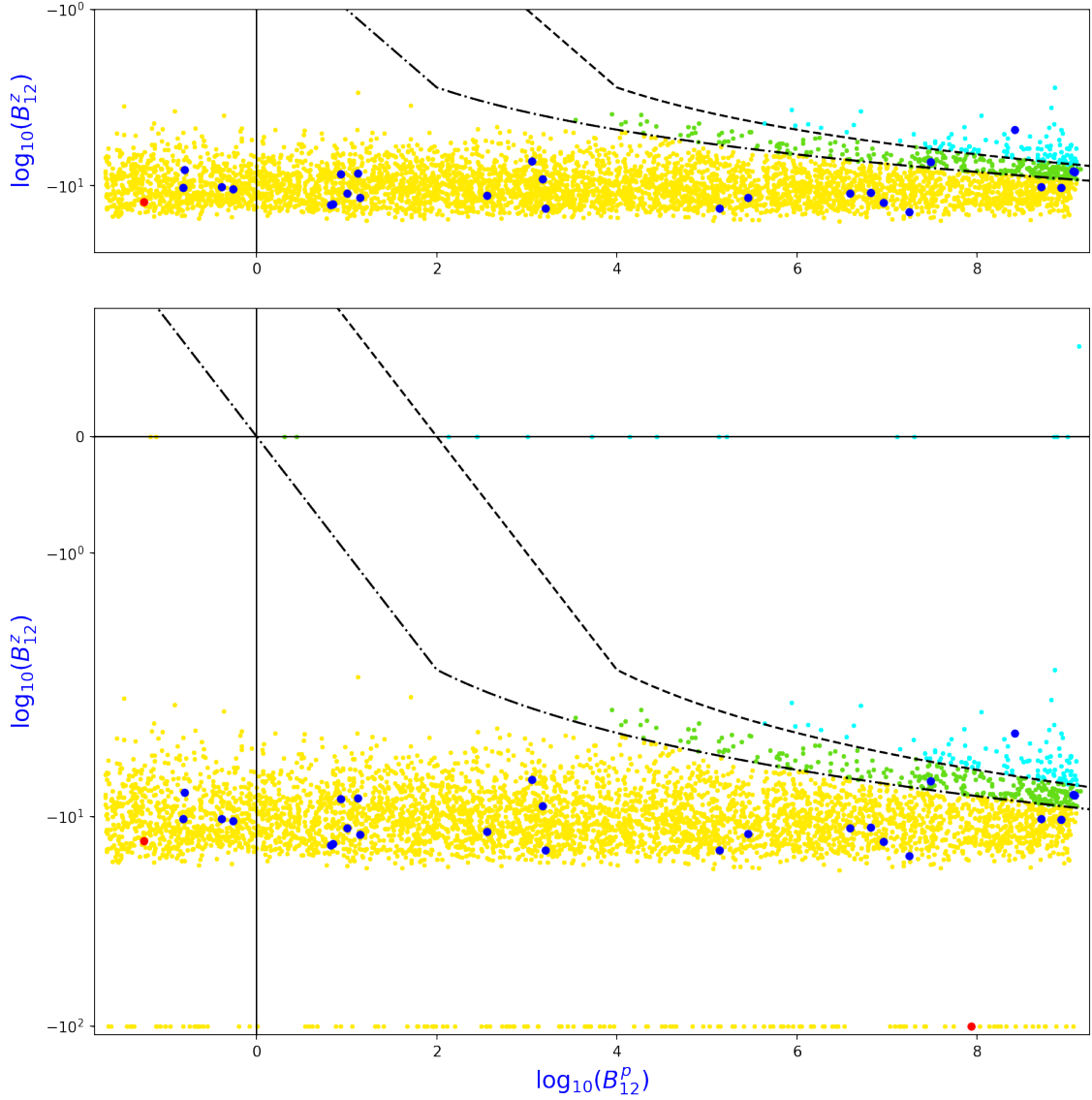


Figura 20: Representación del logaritmo en base diez del factor de Bayes espectroscópico, B_{12}^z (en la figura, el logaritmo está en escala logarítmica) frente al logaritmo en base diez del factor de Bayes posicional, B_{12}^p para un total de 4938 emparejamientos que cumplen que $z_h > 1$, $z_h > z_g$ y $\phi < 54$ arcsec. Los puntos de color rojo son los pareados en los que participa un candidato a HALO que cumple el criterio de [Negrello et al. \(2010\)](#); en los de color azul añil participa un candidato a HALO que no cumple la condición anterior, pero si que cumple el criterio de [González-Nuevo et al. \(2012\)](#); los de color azul cian, son aquellos que no cumplen ninguno de los criterios anteriores pero tienen un valor $B_{12} > 100$ (la confianza sobre la hipótesis de que se trata del mismo objeto es absoluta); los de color verde no cumplen ninguno de los criterios anteriores pero $1 > B_{12} > 100$; el resto de pareados se representan de color anaranjado. Dado que $\log_{10}(0) = -\infty$, cuando se cumple la condición de que el pareado tiene un valor $B_{12}^z < 10^{-100}$ entonces se le reasigna el valor $B_{12}^z = 10^{-100}$. La línea vertical continua divide la figura en dos secciones, con $B_{12}^p < 1$ y $B_{12}^p > 1$. Los puntos con $B_{12}^p < 1$ se corresponden con aquellas contrapartidas con una separación angular $\phi \geq 49.64$ arcsec. En la zona superior de la figura, se han destacado aquellos pareados en los que participan objetos de *H*-ATLAS cuyo redshift ha sido calculado mediante el método descrito en la Sección 3. Los candidatos propuestos por nuestro criterio están representados en la zona con $B_{12}^p > 1$ y $B_{12} < 1$.

7 Discusión y conclusiones

Para considerar que nuestro método supone un método fiable para la identificación de sistemas lente gravitatoria, el número de contrapartidas que no pueden ser explicadas por una distribución aleatoria debería ser mucho mayor que el número de asociaciones aleatorias que cumplen nuestros criterios y lo que encontramos es justo lo contrario. Los resultados de la simulación indican que siempre que tengamos una población de observaciones con una densidad como la que tenemos en la muestra, es inevitable que se formen del orden de ~ 3500 emparejamientos entre observaciones que distan a $\lesssim 50$ arcsec que no tienen ningún tipo de interacción física. Si las medidas que caracterizan a cada observación fuesen más precisas y más exactas se haría evidente que muchos de los candidatos propuestos o bien se encuentran a una distancia demasiado grande para ser considerados candidatos a lente o bien son observaciones de un mismo objeto en los dos catálogos.

La gran mayoría de los desplazamientos al rojo pertenecientes a las observaciones de *H-ATLAS* que participan en el pareo (*matching*) han sido determinados mediante el ajuste de la SED de la galaxia SMM J2135-0102. Como hemos visto en la Sección 3 tenemos cierta confianza de que el ajuste es válido para estimar el *redshift* de las galaxias en formación, pero no sabemos cuantas de las observaciones de las que hemos estimado el *redshift* son en realidad observaciones de galaxias lejanas. Es posible que estemos realizando un ajuste a galaxias que en realidad se encuentran mucho más cerca. De hecho se podría explicar que encontremos más contrapartidas en la muestra real que en la simulada sin que se diera el fenómeno de lente gravitatoria; ese exceso de contrapartidas podría deberse a observaciones del mismo objeto en los dos catálogos que no hemos sido capaces de identificar como tal porque nuestra estimación del *redshift* es incorrecta.

En este trabajo hemos utilizado el método descrito en la Sección 3 para estimar el *redshift* de muchos de los objetos del catálogo *H-ATLAS*. Para considerar válida esta medida del ajuste debían cumplirse varias condiciones: no haber una estimación previa del corrimiento al rojo de esa observación, debe ser una galaxia (recordemos, según la información disponible en columna GSQ_FLAG de este mismo catálogo) y el valor del ajuste debe encontrarse en un rango comprendido entre $z = 1$ y $z = 3.5$. Opino que deberíamos buscar más indicios antes de considerar válido el ajuste realizado con ese método.

Nuestra propuesta para identificar lentes gravitatorias se ha planteado en el supuesto de que únicamente interviene un único objeto como lente, sin embargo, en muchas ocasiones intervienen grupos de galaxias y no galaxias individuales en constitución de la lente. El criterio de que se trata de una lente gravitatoria por el hecho encontrar dos objetos con *redshift* diferentes suficientemente próximos posiblemente también sea poco realista. Partiendo de un estudio cuidadoso del entorno que se encuentran lentes gravitatorias confirmadas podríamos elaborar métodos para la identificación de entornos favorables para la presencia de lentes gravitatorias. En este sentido es posible que tengamos que identificar cúmulos de galaxias más que objetos individuales como lentes gravitatorias.

El criterio Bayesiano descrito en la Sección 4 es un criterio de cross-identificación de catálogos y por tanto no ha sido pensado para la identificación SLGs. Los factores bayesianos están diseñados para contrastar la hipótesis H_1 que supone los emparejados están formados por observaciones de un mismo objeto astronómico frente a la hipótesis H_2 que supone que las observaciones pertenecen a dos objetos diferentes. Deberían plantearse nuevos factores de Bayes, que permitieran contrastar por ejemplo la hipótesis de que una observación considerada pertenece a una ETGs o no, o la hipótesis de que el emparejado considerado es una lente gravitatoria frente a la hipótesis de que no lo es. Plantear estos factores de Bayes exige conocer con mayor profundidad el fenómeno de las lentes gravitatorias y la naturaleza de las ETGs, lo cual se escapa los objetivos académicos de este trabajo.

También hay lugar para mejorar nuestro método de cross-identificación. Los factores de Bayes son quizás demasiado simples y se pueden mejorar. Las funciones densidad de *probabilidad a*

priori son funciones no informativas que se han introducido por simplicidad matemática pero cabe plantearse otras más adecuadas. Por ejemplo podríamos incorporar la información de la que disponemos sobre la distribución del *redshift* en los catálogos para mejorar el factor de Bayes fotométrico. El modelo de precisión astrométrica que se ha considerado es bastante conservador. Se ha considerado que el error astrométrico es igual a la resolución angular del telescopio, lo cuál es razonable cuando las fuentes son puntuales. Las galaxias que aparecen en los catálogos no son puntuales, tienen una extensión y en estos casos es frecuente que σ^p sea inferior a la resolución angular del telescopio. Se podría haber tomado un valor de σ^p menor, pero no se ha hecho por prudencia.

La resolución espacial del telescopio *Herschel* también juega un papel fundamental en este trabajo. Una mejora de este parámetro permitiría tener unas medidas sobre el flujo de las fuentes más precisas con una menor contaminación por parte de las fuentes próximas más débiles. Además permitiría mejorar los modelos de la SED de las galaxias tempranas y podríamos introducir mejoras en nuestro método para la estimación del *redshift* fotométrico. No solo eso. La mejora del modelo de precisión astrométrica, junto con una menor incertidumbre asociada a la posición de la fuente permitiría estudiar las estructuras formadas por las lentes gravitatorias a una escala menor, más adecuada al orden de magnitud en el que se producen estos fenómenos. Esto también reduciría la distancia máxima de emparejamiento que se ha considerado entre las fuentes y se producirían menos emparejamientos aleatorios entre observaciones de objetos que no guardan ningún tipo de relación entre sí.

Por tanto no podemos considerar nuestro método como una alternativa razonable a las ya existentes y tampoco tiene mucho sentido comparar el número de candidatos obtenidos por nuestro método con el número obtenido utilizando los métodos que encontramos en la literatura. Sin embargo se pueden plantear modificaciones en el método que quizás conduzcan a una selección de candidatos más fiable y con mayor pureza. Además podríamos considerar el método propuesto como una fase previa a un estudio más detallado porque permitiría identificar de una forma rápida, sencilla y directa, basada en un criterio estadístico riguroso, en torno al tres por ciento de los candidatos con mayor probabilidad de ser sistemas lente de entre las galaxias del catálogo *H-ATLAS DR1*.

A Redshifts fotométricos de los candidatos

H-ATLAS	RA	DEC	S_{500} (mJy)	S_{350} (mJy)	S_{250} (mJy)	z_H	σ_H	z_{phot}	σ_{phot}
J090740.0-004200	136.917	-0.7	183 (9)	358 (8)	507 (7)	1.58 (0.01)		1.37 (0.02)	
J091043.0-000321	137.679	$-5.58 \cdot 10^{-2}$	249 (10)	403 (8)	462 (7)	1.79 (0.01)		1.83 (0.08)	
J090302.9-014127	135.762	-1.69	230 (9)	342 (8)	343 (7)	3.04 (0.01)		2.18 (0.05)	
J091331.3-003642	138.380	-0.61	89 (10)	151 (8)	187 (7)	1.50 (0.30)		1.65 (0.02)	
J090051.0+015049	135.212	1.85	54 (10)	120 (8)	177 (7)	1.32 (0.28)		1.26 (0.06)	
J090952.9-010811	137.470	-1.14	90 (9)	131 (8)	159 (7)	1.54 (0.40)		1.76 (0.20)	
J091341.4-004342	138.422	-0.73	72 (9)	124 (8)	159 (7)	1.53 (0.30)		1.59 (0.02)	
J090957.6-003619	137.490	-0.61	83 (9)	129 (8)	132 (7)	1.91 (0.46)		2.11 (0.02)	
J090749.7-003807	136.957	-0.64	69 (9)	113 (8)	140 (7)	1.48 (0.34)		1.67 (0.05)	
J090311.6+003906	135.798	0.65	173 (10)	202 (8)	135 (7)	2.63 (0.01)		3.07 (0.06)	
J090356.8+002310	135.987	0.39	64 (9)	115 (8)	131 (7)	1.42 (0.36)		1.80 (0.04)	
J090448.8+021646	136.203	2.28	52 (9)	97 (8)	126 (7)	1.47 (0.29)		1.52 (0.03)	
J090033.8+001957	135.141	0.33	44 (9)	94 (8)	125 (7)	1.33 (0.28)		1.42 (0.08)	
J090459.3+020837	136.247	2.14	66 (9)	102 (8)	111 (7)	1.79 (0.41)		2.02 (0.06)	
J091056.5-002919	137.735	-0.49	54 (9)	98 (8)	116 (7)	1.68 (0.31)		1.71 (0.05)	
J091521.5-002443	138.840	-0.41	82 (9)	106 (8)	116 (7)	2.19 (0.42)		2.15 (0.22)	
J090542.0+020733	136.425	2.13	58 (10)	100 (8)	112 (7)	1.82 (0.33)		1.86 (0.00)	
J091304.9-005343	138.270	-0.9	112 (9)	141 (8)	110 (7)	2.30 (0.01)		2.75 (0.00)	
J090626.6+022612	136.611	2.44	71 (9)	99 (8)	113 (7)	2.01 (0.42)		2.02 (0.16)	
J090459.9+015043	136.250	1.85	62 (9)	88 (8)	110 (7)	1.74 (0.42)		1.71 (0.22)	
J090408.6+012610	136.036	1.44	46 (10)	95 (8)	103 (7)	1.79 (0.36)		1.86 (0.20)	
J090403.9+005619	136.016	0.94	50 (9)	87 (8)	104 (7)	1.69 (0.31)		1.71 (0.01)	
J090653.3+023207	136.722	2.54	64 (9)	95 (8)	102 (7)	2.08 (0.37)		2.07 (0.08)	
J090707.9-003134	136.783	-0.53	56 (10)	95 (8)	102 (7)	1.94 (0.34)		1.99 (0.02)	
J091305.1-001409	138.271	-0.24	62 (9)	97 (8)	95 (7)	2.18 (0.37)		2.18 (0.01)	
J090732.3-005207	136.885	-0.87	52 (9)	103 (8)	91 (7)	2.13 (0.47)		2.16 (0.24)	
J091354.6-004539	138.478	-0.76	50 (9)	87 (8)	92 (7)	1.95 (0.34)		1.99 (0.04)	
J090504.8+000800	136.270	0.13	71 (9)	97 (8)	97 (7)	2.18 (0.46)		2.26 (0.11)	
J090308.3-000420	135.785	$-7.22 \cdot 10^{-2}$	55 (10)	95 (8)	99 (7)	1.98 (0.35)		2.02 (0.05)	
J090705.7+002128	136.774	0.36	64 (10)	92 (8)	90 (7)	2.28 (0.38)		2.25 (0.06)	
J090433.4-010740	136.139	-1.13	70 (9)	104 (8)	84 (7)	2.52 (0.44)		2.51 (0.12)	
J090239.0+002819	135.663	0.47	48 (9)	85 (8)	97 (7)	1.81 (0.32)		1.80 (0.03)	
J090931.8+000133	137.383	$2.58 \cdot 10^{-2}$	56 (10)	88 (8)	92 (7)	1.91 (0.54)		2.07 (0.03)	
J090459.0-012911	136.246	-1.49	44 (9)	88 (8)	92 (7)	1.87 (0.37)		1.89 (0.19)	
J091148.2+003355	137.951	0.57	72 (9)	102 (8)	90 (7)	2.45 (0.40)		2.43 (0.01)	
J090319.6+015635	135.832	1.94	61 (9)	89 (8)	87 (7)	2.26 (0.38)		2.24 (0.04)	
J090405.3-003331	136.022	-0.56	76 (10)	98 (8)	85 (7)	2.57 (0.41)		2.55 (0.06)	
J085751.3+013334	134.464	1.56	64 (10)	90 (8)	88 (7)	2.29 (0.38)		2.26 (0.07)	
J085900.3+001405	134.751	0.23	50 (10)	92 (8)	89 (7)	2.06 (0.39)		2.10 (0.14)	
J091351.7-002340	138.465	-0.39	48 (10)	89 (8)	90 (7)	1.99 (0.37)		1.95 (0.32)	
J090446.4+022218	136.193	2.37	61 (9)	86 (8)	87 (7)	2.24 (0.38)		2.22 (0.09)	
J091003.5+021028	137.515	2.17	46 (9)	86 (8)	87 (7)	1.97 (0.37)		2.01 (0.12)	
J090429.6+002935	136.123	0.49	60 (10)	86 (8)	85 (7)	2.18 (0.60)		2.23 (0.07)	
J090032.7+004316	135.136	0.72	50 (10)	85 (8)	85 (7)	2.08 (0.36)		2.09 (0.06)	
J090613.7-010044	136.557	-1.01	69 (9)	86 (8)	79 (7)	2.52 (0.41)		2.50 (0.13)	
J090453.2+022018	136.222	2.34	88 (9)	107 (8)	83 (7)	2.80 (0.44)		2.80 (0.02)	
J085855.3+013728	134.730	1.62	68 (9)	92 (8)	80 (7)	2.51 (0.40)		2.50 (0.02)	
J090346.1+013428	135.942	1.57	68 (9)	91 (8)	79 (7)	2.53 (0.41)		2.51 (0.03)	
J090954.6+001754	137.478	0.3	76 (10)	111 (8)	83 (7)	2.67 (0.48)		2.66 (0.18)	
J090954.6+001754	137.478	0.3	76 (10)	111 (8)	83 (7)	2.67 (0.48)		2.66 (0.18)	
J090440.0-013439	136.167	-1.58	76 (9)	98 (8)	76 (7)	2.75 (0.44)		2.73 (0.03)	
J090950.8+000427	137.462	$7.42 \cdot 10^{-2}$	68 (9)	92 (8)	81 (7)	2.50 (0.40)		2.48 (0.03)	

H-ATLAS	RA	DEC	S_{500} (mJy)	S_{350} (mJy)	S_{250} (mJy)	z_H	σ_H	z_{phot}	σ_{phot}
J090204.1-003829	135.517	-0.64	61 (9)	87 (8)	79 (7)	2.40 (0.39)		2.37 (0.01)	
J090930.4+002224	137.377	0.37	57 (9)	85 (8)	76 (7)	2.39 (0.40)		2.36 (0.04)	
J090409.4+010734	136.039	1.13	55 (10)	86 (8)	79 (7)	2.29 (0.39)		2.28 (0.06)	
J090310.6+015149	135.794	1.86	71 (9)	105 (8)	77 (7)	2.69 (0.49)		2.66 (0.21)	
J085859.2+002818	134.747	0.47	52 (9)	85 (8)	77 (7)	2.28 (0.41)		2.26 (0.10)	
J090530.4+012800	136.377	1.47	65 (9)	90 (8)	78 (7)	2.51 (0.41)		2.48 (0.00)	
J090818.9+023330	137.079	2.56	57 (10)	92 (8)	77 (7)	2.42 (0.44)		2.40 (0.16)	
J090441.5+015154	136.173	1.86	70 (10)	87 (8)	76 (7)	2.60 (0.42)		2.57 (0.10)	
J090739.1-003948	136.913	-0.66	81 (9)	99 (8)	73 (7)	2.89 (0.45)		2.87 (0.03)	
J091257.2+000300	138.238	$5 \cdot 10^{-2}$	87 (10)	96 (8)	69 (7)	3.03 (0.46)		3.03 (0.06)	
J090819.1-002026	137.080	-0.34	66 (10)	93 (8)	69 (7)	2.71 (0.47)		2.71 (0.16)	
J090813.0-003657	137.054	-0.62	65 (10)	88 (8)	66 (7)	2.74 (0.45)		2.73 (0.11)	
J085908.5+011320	134.785	1.22	70 (10)	87 (8)	67 (7)	2.81 (0.44)		2.79 (0.00)	

Cuadro 2: Muestra de 64 candidatos a SLGs procedente del artículo [González-Nuevo et al. \(2012\)](#). Los errores se encuentran entre paréntesis a la derecha de cada medida. La columna z_H contiene los valores del *redshift* que estos autores proponen para estos objetos mientras que la columna z contiene los valores obtenidos a partir de nuestro código.

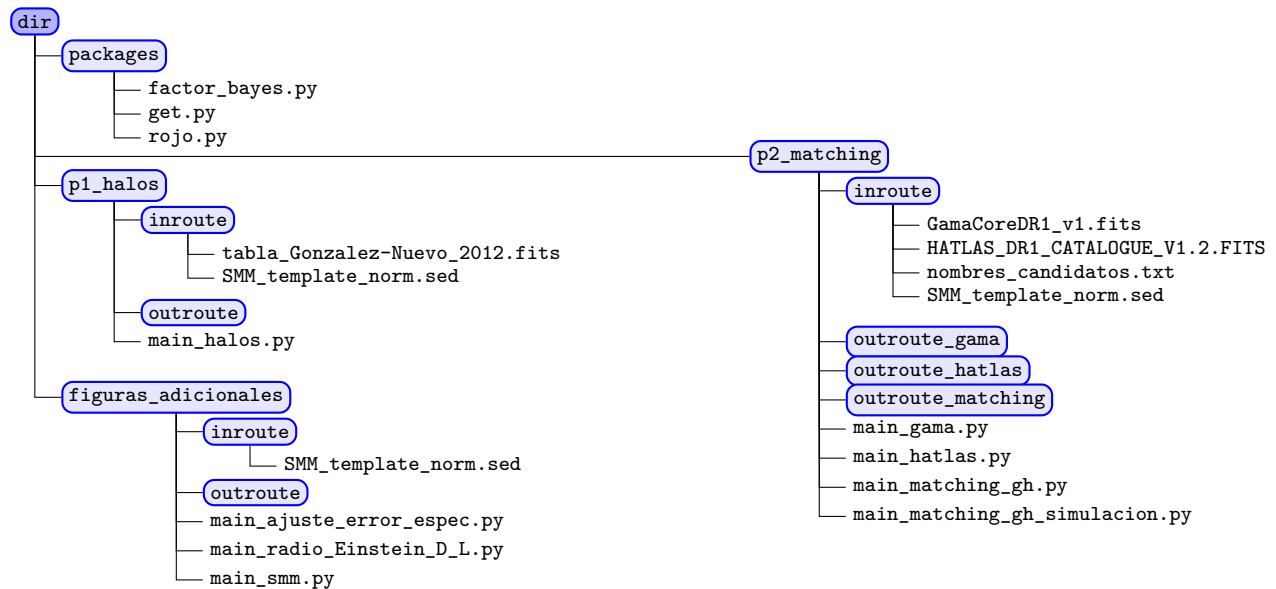
B Selección de objetos del catálogo GAMA

N	GAMA_IAU_ID	Z_HELIO	z (NED)	σ^z (NED)	Z_SOURCE	Z_QUALITY
82	GAMAJ113849.92+004538.5	0.311930	0.312005	0.000102	1	5
100	GAMAJ113908.49+004730.4	0.385340	0.385261	0.000103	1	5
317	GAMAJ114324.58+004208.9	0.133410	0.133342	0.000092	1	5
382	GAMAJ114451.38+004528.8	0.125480	0.125497	0.000079	1	5
526	GAMAJ114836.47+004652.6	0.126110	0.126204	0.000100	1	5
31,586	GAMAJ115703.70-015022.1	0.130600	0.130568	0.000133	1	5
37,709	GAMAJ114410.74-012829.0	0.123880	0.123867	0.000136	1	5
41,400	GAMAJ091612.24-004559.3	0.103640	0.103692	0.000048	1	5
41,522	GAMAJ092045.05-003814.1	0.223980	0.223991	0.000089	1	5
42,400	GAMAJ085838.75-001346.6	0.385480	0.385721	0.000117	1	5
43,827	GAMAJ085317.71+000652.3	0.374820	0.374708	0.000120	1	5
58,913	GAMAJ121304.13+015112.8	0.081010	0.081022	0.000130	1	5
59,034	GAMAJ121436.56+014019.0	0.394020	0.394029	0.000175	1	5
64,534	GAMAJ145131.71+014151.6	0.134480	0.134499	0.000119	1	5
88,264	GAMAJ091232.95-005020.9	0.476090	0.476045	0.000180	1	5
92,501	GAMAJ143741.98-005132.2	0.291600	0.291690	0.000098	1	5
99,245	GAMAJ144348.43-002559.8	0.305250	0.305359	0.000119	1	5
111,851	GAMAJ091613.75+004202.3	0.038080	0.038159	0.000065	1	5
114,310	GAMAJ142545.90+002242.7	0.326270	0.326157	0.000351	1	5
114,432	GAMAJ121142.58+010337.1	0.293560	0.293476	0.000351	1	5
90,574	GAMAJ121334.58-005813.3	0.099400	0.099792	0.000084	2	4
90,673	GAMAJ121514.87-005252.3	0.075317	0.075319	0.000088	1	4
220	GAMAJ114058.31+004424.2	0.430774	0.430740	0.000214	1	3
315	GAMAJ114316.16+003848.7	0.118100	0.118100	0.000410	2	3
964	GAMAJ115827.70+004304.4	0.047500	0.047439	0.000117	10	3
1,206	GAMAJ120213.41+004401.7	0.116100	0.116000	0.005000	8	3
41,568	GAMAJ092147.41-004048.7	0.154451	0.154481	0.000076	1	3

Cuadro 3: Muestra que contiene 27 observaciones procedentes del catálogo GAMA DR1 cuyo *redshift* se encuentra también disponible en la base de datos NED.

C Código en Python

Los programas que se presentan en esta sección contienen todo lo necesario para realizar todos los cálculos y gráficas a los que se ha hecho referencia en las secciones anteriores; están escritos en lenguaje de programación Python 3.6.0 y se requiere tener instaladas las librerías `scipy` 0.18, `matplotlib` 2.0.0, `numpy` 1.11.3 y `astropy` 1.3. El código se encuentra repartido en varios ficheros llamados módulos. Para que los ejecutables funcionen correctamente, sin cambiar nada, debemos mantener la siguiente estructura de directorios:



Los ejecutables son todos aquellos ficheros cuyo nombre contiene la palabra `main`. Todos ellos leen los datos que necesitan de los ficheros situados en los directorios `inroute`, a excepción de los ejecutables `main_matching.py` y `main_matching_simulacion.py` cuyos ficheros de entrada se encuentran en `outroute_gama` y `outroute_hatlas`. Las gráficas y tablas de datos se guardan en los directorios que contienen la palabra `outroute`.

Todos los ejecutables hacen llamadas a una serie de funciones que son comunes, por ese motivo, para evitar repetir el código de esas funciones en cada ejecutable, éstas funciones se han definido en tres módulos que se encuentran en el directorio `packages`. Cada módulo y cada función se encuentra debidamente comentada. Recuerdo al lector las cadenas de documentación son accesibles mediante:

```

>>> help("nombre_modulo.nombre_funcion")
o
>>> print(nombre_funcion.__doc__)

```

También se ha realizado un esquema sobre la estructura del ejecutable `main_halos.py` (C.1).

C.1 Esquema del módulo `main_halos.py`

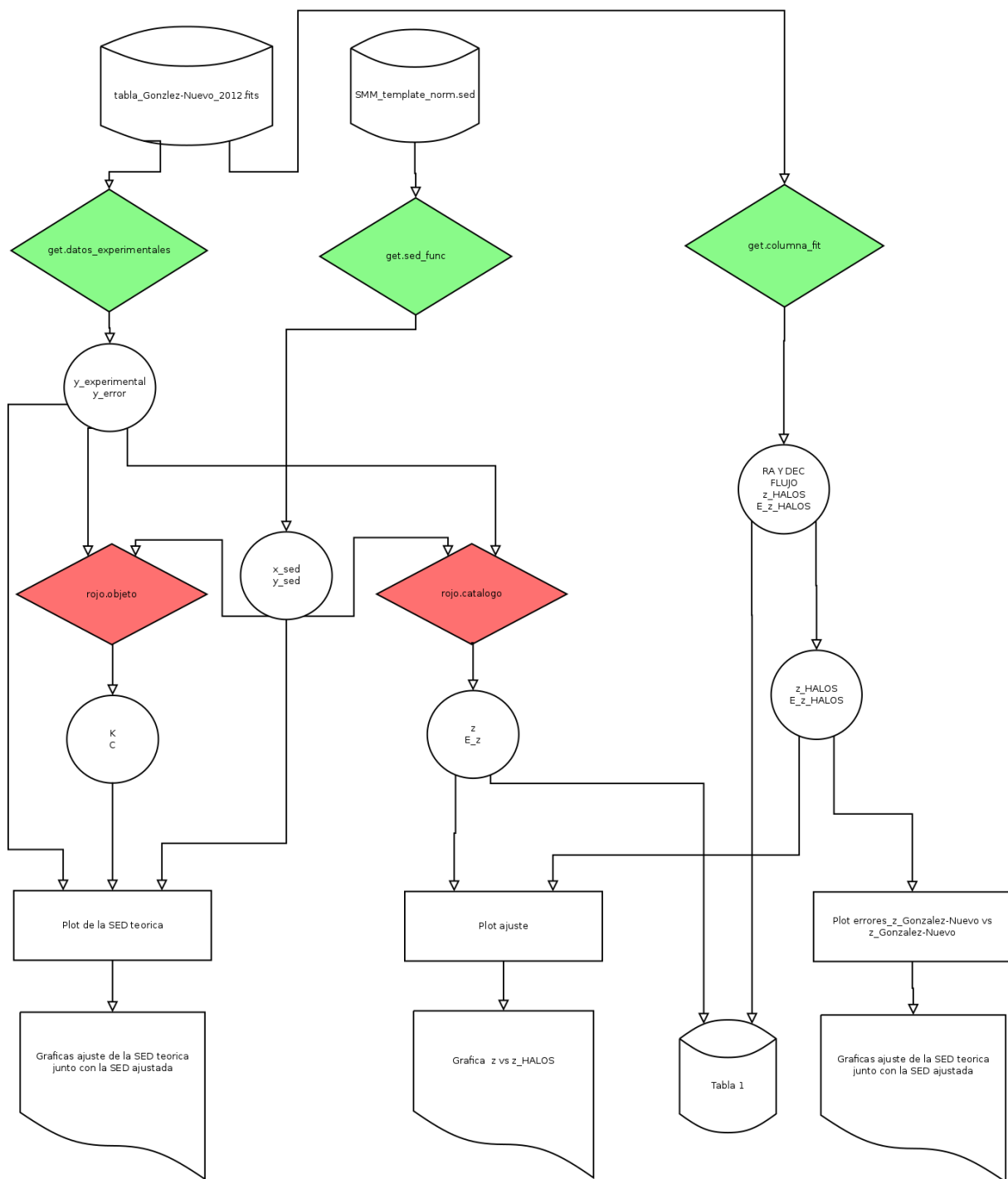


Figura 21: Este esquema representa la estructura completa del ejecutable `main_halos.py`. Los ficheros con formato `.fits` están representados mediante figuras cilíndricas; los que se encuentran en la parte superior de la figura, son los ficheros de entrada y el situado en la parte inferior es el fichero de salida a partir del cual se obtiene la tabla 2. Este ejecutable también devuelve varias figuras representadas en la zona inferior del esquema. Las funciones se representan mediante rombos y los colores ayudan a identificar el módulo en el que están definidas. La salida de que se obtiene a partir de cada función se encuentra en el interior de un círculo.

C.2 Módulo main_halos.py

```
# -*- coding: utf-8 -*-

"""
Calcula los redshifts de los objetos que aparecen en la tabla del artículo
Gonzalez-Nuevo et al 2012 y compara estos valores con los que obtuvieron estos
autores. Por último se obtiene un ajuste con el que se calcularán los
errores asociados a las medidas obtenidas con las funciones definidas en el
modulo "rojo".

Python 3.6.0    scipy 0.18    matplotlib 2.0.0    numpy 1.11.3    astropy 1.3

@author: Javier Gutiérrez Solórzano
"""

#=====
#   Nombres de los ficheros de entrada/salida y su localización.
#=====
#-----
#   Directorio en el que se encuentran los modulos.
#-----
import sys
sys.path.append("../packages/")
#-----
#   Directorios de los ficheros.
#-----
inroute = './inroute/'
outroute = './outroute/'
#-----
#   Nombre de los ficheros.
#-----
infile_template = 'SMM_template_norm.sed'
infile_catalogo = "catalogo_paper_joaquin_2012.fits"

outfile = "comparacion_Z.fits"
outpicture = "comparacion_Z.png"

#=====
#   Desplazamiento al rojo de los objetos H-Atlas.
#=====
#   Soporte de unidades físicas.
from astropy import units as u
#   Soporte de magnitudes físicas. Solo se utiliza la velocidad de la luz.
from astropy.constants import c

#   Importamos el modulo get donde se definen las funciones para las funciones
#   utilizadas para la lectura de los ficheros y tablas.
import get

#-----
```



```

# # Entrada:
# Leemos el fichero de la SED modelo. El fichero contiene dos columnas; la
# primera representa las longitudes de onda en Ångström y la segunda valores
# valores de la densidad espectral de flujo ( $F_{\lambda}$ ), en unidades de:
#  $(\text{erg}) * (\text{cm})^{(-2)} * (\text{s})^{(-1)} * (\text{\AA})^{(-1)}$ 
#-----
[x_sed, y_sed] = get.sed_func(inroute + infile_template)
# Unidades a las longitudes de onda
x_sed=x_sed*u.AA # En Ångström
# Unidades de densidad espectral de flujo ( $F_{\lambda}$ )
y_sed=y_sed*((u.erg)* (u.cm)**(-2)* (u.s)**(-1)* (u.AA)**(-1))
# Cambio de  $F_{\lambda}$  a  $F_{\epsilon}$  multiplicando por  $(\lambda^2)/c$ 
y_sed=y_sed*(x_sed**2)*(c**-1)
# Dejo los arrays con las unidades que tienen los valores experimentales que
# voy a utilizar
x_sed=x_sed.to(u.micron)
y_sed=y_sed.to(u.mJy)

#-----
# Obtenemos los valores experimentales (mediciones realizadas por HATLAS) de
# todos los objetos de un catálogo a partir de la lectura de un fichero con
# formato .fits
#-----
x_experimental = [250,350,500]*u.micron
[y_experimental, y_error]= get.datos_experimentales(inroute + infile_catologo)

y_experimental = y_experimental*u.mJy
y_error = y_error*u.mJy

#-----
# Salida
#-----
import numpy as np

import rojo
[z,E_z] = rojo.z_phot_hatlas(x_sed,y_sed,x_experimental,y_experimental,y_error)

#=====
# Creación del fichero fits de 'salida'
#=====
# Creamos varios arrays que contendrán los elementos de las otras filas
# del fichero de salida.
outfile_array_HATLAS_DR1_CATALOGUE = get.columna_fits(inroute +
                                                    infile_catologo,0)

outfile_array_RA_h = get.columna_fits(inroute + infile_catologo,2)
outfile_array_DEC_h = get.columna_fits(inroute + infile_catologo,3)

outfile_array_S250 = get.columna_fits(inroute + infile_catologo,4)
outfile_array_S350 = get.columna_fits(inroute + infile_catologo,5)

```

```

outfile_array_S500 = get.columna_fits(inroute + infile_catalogo,6)

outfile_array_e_S250 = get.columna_fits(inroute + infile_catalogo,7)
outfile_array_e_S350 = get.columna_fits(inroute + infile_catalogo,8)
outfile_array_e_S500 = get.columna_fits(inroute + infile_catalogo,9)
#   outfile_array_z es un array con los valores de z que se encuentran en el
#   articulo.
outfile_array_z = get.columna_fits(inroute + infile_catalogo,10)
outfile_array_e_z = get.columna_fits(inroute + infile_catalogo,11)

from astropy.io import fits
#-----
#   Creamos las columnas del fichero de salida.
#-----
col0 = fits.Column(name='HATLAS_DR1_CATALOGUE', format='23A',
                    array=outfile_array_HATLAS_DR1_CATALOGUE)
col1 = fits.Column(name='RA_H', format='D', array=outfile_array_RA_h,
                    unit='Degrees')
col2 = fits.Column(name='DEC_H', format='D', array=outfile_array_DEC_h,
                    unit='Degrees')

col3 = fits.Column(name='S500', format='D', array=outfile_array_S500,
                    unit='Jy')
col4 = fits.Column(name='e_S500', format='D', array=outfile_array_e_S500,
                    unit='Jy')

col5 = fits.Column(name='S350', format='D', array=outfile_array_S350,
                    unit='Jy')
col6 = fits.Column(name='e_S350', format='D', array=outfile_array_e_S350,
                    unit='Jy')

col7 = fits.Column(name='S250', format='D', array=outfile_array_S250,
                    unit='Jy')
col8 = fits.Column(name='e_S250', format='D', array=outfile_array_e_S250,
                    unit='Jy')

#   Desplazamiento al rojo
col9 = fits.Column(name='Z_ajuste', format='D', array=outfile_array_z)
col10 = fits.Column(name='Error_Z_ajuste', format='D', array=outfile_array_e_z)

col11 = fits.Column(name='Z_HALOS', format='D', array=z)
col12 = fits.Column(name='Error_Z_HALOS', format='D', array=E_z)
#   Ahora fabrico un objeto con esas columnas:
cols = fits.ColDefs([col0, col1, col2, col3, col4, col5, col6, col7, col8,
                     col9, col10, col11, col12])
#   Ese objeto se utiliza para crear la tabla HDU.
tbhdu = fits.BinTableHDU.from_columns(cols)
#   Y al final lo escribimos en un fichero 'outfile' en la ruta 'outroute'.
tbhdu.writeto(outroute + outfile,overwrite=True)
#-----

```

```

#  Revisión de datos; z, E_z, outfile_array_z, outfile_array_e_z .
#-----
print('\n Longitud del array z=',len(z),'\n',
      '\n z = ',z,'\n',
      '\n E_z = ',E_z,'\n',
      '\n Longitud del array outfile_array_z=',len(outfile_array_z),'\n',
      '\n outfile_array_z = ',outfile_array_z,'\n',
      '\n outfile_array_e_z = ',outfile_array_e_z)

#=====
#  Graficas
#=====
import matplotlib.pyplot as plt
#-----
#  Graficas de los ajustes.
#-----
for i in range(6,10,1): # 3,10
    K=rojo.z_phot_objeto(x_sed, y_sed, x_experimental, y_experimental[i],
                        y_error[i])[0][0]
    C=rojo.z_phot_objeto(x_sed, y_sed, x_experimental, y_experimental[i],
                        y_error[i])[0][1]

    figura=plt.figure()

    plt.loglog(x_sed,y_sed, color="blue", linewidth=1.5, linestyle="--",
               label="SMM J2")
    plt.loglog(x_sed*K,y_sed*C, color="red")
    plt.loglog(x_experimental,y_experimental[i], '+', color='black')

    plt.axvline(500*(1+z[i]), linewidth=0.75, color = 'yellow')
    plt.axvline(500, color = 'green',linewidth=0.75, linestyle="--")

    plt.grid(False)

    plt.title(outfile_array_HATLAS_DR1_CATALOGUE[i], fontsize = 10,
              color='blue')
    plt.xlabel('$ \mathrm{\lambda}; [\mu \mathrm{m}]$', fontsize
              = 10, color='blue')
    plt.ylabel('$ {\mathrm{F}}_{\mathrm{J}}; [\mathrm{mJy}]$',
              fontsize = 10, color='blue')
    # Transforma 'i' en un string para utilizarlo en el nombre de la figura.
    indice = str(i)
    #-----
    #  a)
    figura.savefig(outroute + 'ajuste_' + indice, dpi=150, transparent=True)
    #-----
    #  b)
    #  Cambiamos los limites de los ejes para hacer ampliar la zona en la
    #  que los puntos experimentales se encuentran mas proximos a la curva de
    #  ajuste.

```

```

#      #      [xmin,xmax,ymin,ymax]
#      plt.axis([200, 700, 10, 500])
#      figura.savefig(outroute + 'ajuste_' + indice + '_zoom', dpi=150,
#                      transparent=True)
#-----
# Muestra la figura.
plt.show()

#-----
# Grafica de Z_articulo vs Z_programa
#-----
# Transforma un tipo 'list' a 'array'
# Ajuste, escribe en la variable f los parámetros del ajuste y la matriz de
# covarianza de estos parámetros.
f = np.polyfit(outfile_array_z, z, 1, cov=True)

C = f[1] # matriz de covarianza.
p = f[0] # valores del ajuste.

afit = p[0] # pendiente ajustada.
bfit = p[1] # ord. origen ajustada.
#-----
def fsup(x,Cmat,nsigmas): # función que calcula el extremo superior de la
                        # banda, a NSIGMAS sigmas.
    ymean = afit*x+bfit
    sa     = Cmat[0,0]
    sb     = Cmat[1,1]
    sab    = Cmat[0,1]
    y      = ymean+nsigmas*np.sqrt(x*x*sa+sb+2*x*sab)
    return y

def finf(x,Cmat,nsigmas): # función que calcula el extremo inferior de la
                        # banda, a NSIGMAS sigmas.
    ymean = afit*x+bfit
    sa     = Cmat[0,0]
    sb     = Cmat[1,1]
    sab    = Cmat[0,1]
    y      = ymean-nsigmas*np.sqrt(x*x*sa+sb+2*x*sab)
    return y
#-----
# Figura z_articulo vs z_programa.
#-----
figura = plt.figure(num = None, figsize = (9, 9), dpi = 80, facecolor = 'w',
                    edgecolor = 'k')
# Cambiamos los limites de los ejes.
plt.axis([1, 3.5, 1, 3.5])
# Genera 100 valores igualmente espaciados entre 1 y 3.5.
x = np.linspace(1, 3.5, 100, endpoint=True)
# Recta del ajuste.
plt.plot(x,afit*x+bfit,'r',linewidth=0.5)

```

```

# Sombreado amarillo para la banda 1 sigma.
plt.fill_between(x,finf(x,C,2),fsup(x,C,2),facecolor='yellow', alpha=0.8)
# Sombreado amarillo para la banda 2 sigma.
plt.fill_between(x,finf(x,C,1),fsup(x,C,1),facecolor='orange', alpha=0.8)
# Dibuja los puntos y sus barras de error.
plt.errorbar(outfile_array_z, z, fmt='+', label="data", linewidth=0.15,
             xerr=outfile_array_e_z, yerr=E_z, ecolor='black')
# Sin rejilla de fondo.
plt.grid(False)
# Etiqueta eje x.
plt.xlabel('$\{z\}_{\mathrm{H}}$', fontsize = 16, color='blue')
# Etiqueta eje y.
plt.ylabel('$\{z\}_{\mathrm{phot}}$', fontsize = 16, color='blue')
# Muestra la figura.
plt.show()
# Guarda una imagen de la gráfica.
figura.savefig(outroute + outpicture, dpi=150, transparent=True)
#-----
# Muestra datos del ajuste por pantalla.
#-----

print('z_HALOS vs z_ajuste:', '\n',
      '--- Best fit      y = ({0}'.format(p[0]), chr(177) ,
      '{0})'.format(np.sqrt(C[0,0])), chr(215),
      'x + ({0}'.format(p[1]), chr(177) , '{0})'.format(np.sqrt(C[1,1])))

#-----
# Grafica errores_z_articulo vs z_articulo.
#-----
from scipy.optimize import curve_fit

def func(x, A):
    """
        Modelo para nuestros datos.

    """
    return A*(1+x)

# pcov - Matriz covarianza.
# popt - Vector con valores de ajuste.
popt_errores, pcov_errores = curve_fit(func, outfile_array_z
                                     , outfile_array_e_z)

def recta_sup(x,opt_par,cov_matrix,nsigmas): # función que calcula el
                                             # extremo superior de la banda, a NSIGMAS sigmas.
    ymean = opt_par[0]*(x+1)
    sa     = cov_matrix[0,0]
    y      = ymean+nsigmas*np.sqrt(sa)
    return y

```

```

def recta_inf(x,opt_par,cov_matrix,nsigmas): # función que calcula el
# extremo inferior de la banda, a NSIGMAS sigmas.
    ymean = opt_par[0]*(x+1)
    sa      = cov_matrix[0,0]
    y       = ymean-nsigmas*np.sqrt(sa)
    return y

figura = plt.figure(num = None, figsize = (9, 6), dpi = 80, facecolor = 'w',
                    edgecolor = 'k')
# Cambiamos los limites de los ejes.
plt.axis([1.25, 3.1, 0.2, 0.7])
# Genera 100 valores igualmente espaciados entre 1 y 3.5
x = np.linspace(1, 3.2, 10, endpoint=True)
# Recta del ajuste.
plt.plot(x,poptErrores[0]*(1+x), 'r',linewidth=0.5)
# Sombreado amarillo para la banda 1 sigma.
plt.fill_between(x,recta_inf(x,poptErrores,pcovErrores,2),
                ,recta_sup(x,poptErrores,pcovErrores,2),facecolor='yellow', alpha=0.8)
# Sombreado amarillo para la banda 2 sigma.
plt.fill_between(x,recta_inf(x,poptErrores,pcovErrores,1),
                ,recta_sup(x,poptErrores,pcovErrores,1),facecolor='orange', alpha=0.8)
# Dibuja los puntos.
plt.errorbar(outfile_array_z, outfile_array_e_z, fmt='+', label="data"
            , linewidth=0.15, ecolor='black')
# Sin rejilla de fondo.
plt.grid(False)
# Etiqueta eje x.
plt.xlabel('$\{z\}_{\mathrm{H}}$', fontsize = 16, color='blue')
# Etiqueta eje y.
plt.ylabel('$\{\sigma\}^{\{z\}}$', fontsize = 16, color='blue')
# Muestra la figura.
plt.show()
# Guarda una imagen de la gráfica.
figura.savefig(outroute + "ajuste_errores.png", dpi=150, transparent=True)
#-----
# Muestra datos del ajuste por pantalla.
#-----

print('Errores z_HALOS vs z_HALOS:\n',
      '--- Best fit      y = ({0}'.format(poptErrores[0]), chr(177) ,
      '{0})'.format(np.sqrt(pcovErrores[0,0])), chr(215), '(1+x)')

```

C.3 Módulo main_gama.py

```
# -*- coding: utf-8 -*-
```

```
"""
```

A partir de este código se obtiene la tabla .fits con toda la información relativa al proyecto GAMA que se va a utilizar en el matching; el fichero de salida será uno de los ficheros de entrada de main_matching_gh.py

Python 3.6.0 scipy 0.18 numpy 1.11.3 astropy 1.3

@author: Javier Gutiérrez Solórzano

"""

```
#=====
#    Nombres de los ficheros de entrada/salida y su localización.
#=====
#-----
#    Directorio en el que se encuentran los modulos.
#-----
import sys
sys.path.append("../packages/")
#-----
#    Directorios de los ficheros.
#-----
inroute = './inroute/'
outroute = './outroute_gama/'
#-----
#    Nombre de los ficheros.
#-----
infile_catalogo = "GamaCoreDR1_v1.fits"
outfile = "Gama.fits"

import get
import rojo

#=====
#    Desplazamiento al rojo disponible de los objetos de GAMA.
#=====
z_HELIO = get.columna_fits(inroute + infile_catalogo,6)
z_QUALITY = get.columna_fits(inroute + infile_catalogo,7)

[z_gama, E_z_gama, flag_gama] = rojo.z_gama(z_HELIO,z_QUALITY)

#=====
#    Creación del fichero fits de 'salida'.
#=====

outfile_array_GAMA_IAU_ID = get.columna_fits(inroute + infile_catalogo,0)
outfile_array_RA_g = get.columna_fits(inroute + infile_catalogo,3)
outfile_array_DEC_g = get.columna_fits(inroute + infile_catalogo,4)

from astropy.io import fits

#-----
#    Creamos las columnas del fichero de salida.
#-----

col0 = fits.Column(name='GAMA_IAU_ID', format='23A')
```

```

        , array=outfile_array_GAMA_IAU_ID)
col1 = fits.Column(name='RA_G', format='D', array=outfile_array_RA_g
        , unit='Degrees')
col2 = fits.Column(name='DEC_G', format='D', array=outfile_array_DEC_g
        , unit='Degrees')

# Información proporcionada por HATLAS.
col3 = fits.Column(name='z_HELIO', format='D', array=z_HELIO)
col4 = fits.Column(name='z_QUALITY', format='D', array=z_QUALITY)

# Desplazamientos al rojo considerados válidos.
col5 = fits.Column(name='Z_G', format='D', array=z_gama)
col6 = fits.Column(name='Error_Z_G', format='D', array=E_z_gama)
col7 = fits.Column(name='Flag_gama', format='D', array=flag_gama)

cols = fits.ColDefs([col0, col1, col2, col3, col4, col5, col6, col7])

tbhdu = fits.BinTableHDU.from_columns(cols)

tbhdu.writeto(outroute + outfile, overwrite=True)

```

C.4 Módulo main_hatlas.py

```
# -*- coding: utf-8 -*-
```

```
"""
```

A partir de éste código se obtiene una tabla .fits que será uno de los ficheros de entrada al ejecutable main_matching_gh.py ; El fichero de salida contendrá todos los datos relativos al catálogo HATLAS que se van a utilizar en el matching. El cálculo de los desplazamientos al rojo a partir del ajuste a la SED empírica es una tarea que requiere mucho tiempo de cálculo (mas de 1 hora para los 120000 objetos de HATLAS_DR1 con un procesador Intel Core2Duo). Como puede ser desesperante, a medida que se ejecuta el código aparecen distintos mensajes en la consola sobre el tiempo de ejecución que permiten conocer las instrucciones que se están ejecutando.

Python 3.6.0 scipy 0.18 numpy 1.11.3 astropy 1.3 datetime

@author: Javier Gutiérrez Solórzano

```
"""
```

```
# Para conocer el tiempo de ejecución.
```

```
import datetime
```

```
tiempo_inicio = datetime.datetime.now()
```

```
#=====
```

```
#    Nombres de los ficheros de entrada/salida y su localización.
```

```
#=====
```

```
#-----
```

```
#    Directorio en el que se encuentran los modulos.
```

```
#-----
```



```

import sys
sys.path.append("../packages/")
#-----
#   Directorios de los ficheros.
#-----
inroute = './inroute/'
outroute = './outroute_hatlas/'
#-----
#   # Nombre de los ficheros.
#-----
infile_template = 'SMM_template_norm.sed'
infile_catalogo1 = "HATLAS_DR1_CATALOGUE_V1.2.FITS"
outfile = "HATlas.fits"

import datetime
tiempo_inicio = datetime.datetime.now()

#=====
#   Desplazamiento al rojo de los objetos HATLAS a partir del ajuste.
#=====
#   Soporte de unidades físicas.
from astropy import units as u
#   Soporte de magnitudes físicas. Solo se utiliza la velocidad de la luz.
from astropy.constants import c

import get

#-----
#   Entrada:
#   Leemos el fichero de la SED modelo. El fichero contiene dos columnas; la
#   primera representa las longitudes de onda en Angstrom y la segunda valores
#   de la densidad espectral de flujo ( $F_{\lambda}$ ), en "unidades" (está
#   normalizado en 5500Å ) de :  $(\text{erg}) * (\text{cm})^{(-2)} * (\text{s})^{(-1)} * (\text{Å})^{(-1)}$ 
#-----
[x_sed, y_sed] = get.sed_func(inroute + infile_template)
#   Unidades a las longitudes de onda.
x_sed=x_sed*u.Å
#   Unidades de densidad espectral de flujo ( $F_{\lambda}$ )
y_sed=y_sed*((u.erg)*(u.cm)**(-2)*(u.s)**(-1)*(u.Å)**(-1))
#   Cambio de  $F_{\lambda}$  a  $F_{\nu}$  multiplicando por  $(\lambda^2/c)$ 
y_sed=y_sed*(x_sed**2)*(c**-1)
#   Dejo los arrays con las unidades que tienen los valores experimentales que
#   voy a utilizar.
x_sed=x_sed.to(u.micron)
y_sed=y_sed.to(u.Jy)

#-----
#   Obtenemos los valores experimentales (mediciones realizadas por HATLAS) de
#   todos los objetos de un catálogo a partir de la lectura de un fichero con
#   formato .fits

```

```

#-----
x_experimental = [250,350,500]*u.micron
[y_experimental, y_error]= get.datos_experimentales(inroute + infile_catalogo1)
#   Le asignamos unidades físicas.
# EN ESTE CASO LAS MEDIDAS DE DENSIDAD DE FLUJO SE ENCUENTRAN EN mJy!
y_experimental = y_experimental*u.mJy
y_error = y_error*u.mJy
#   Hacemos un cambio a las unidades.
y_experimental=y_experimental.to(u.Jy)
y_error = y_error.to(u.Jy)

#-----
#   Salida.
#-----
b = datetime.datetime.now()
print('Tiempo de ejecucion:',b-tiempo_inicio,'\n',
      'Comienza el calculo de los desplazamientos al rojo ')

import rojo
[z_ajuste, E_z_ajuste] = rojo.z_phot_hatlas(x_sed,y_sed,x_experimental
                                           ,y_experimental,y_error)
b = datetime.datetime.now()
print('El calculo ha durado ',b-tiempo_inicio)

#=====
#   Desplazamiento al rojo HATLAS.
#=====
gsq_flag=get.columna_fits(inroute + infile_catalogo1,32)
z_spec=get.columna_fits(inroute + infile_catalogo1,44)
z_qual=get.columna_fits(inroute + infile_catalogo1,45)

# Rango de z en el que "z_phot_hatlas" se considera valido (z= 1 - 3.5).
z_min=1
z_max=3.5

# Crea las columnas con los redshifts considerados válidos
[z_hatlas, E_z_hatlas, flag_hatlas]=rojo.z_hatlas(gsq_flag,z_spec,z_qual
                                                  ,z_ajuste,z_min,z_max)

#=====
# Creación del fichero .fits
#=====
outfile_array_HATLAS_DR1_CATALOGUE = get.columna_fits(inroute
                                                    + infile_catalogo1,0)
outfile_array_RA_h = get.columna_fits(inroute + infile_catalogo1,2)
outfile_array_DEC_h = get.columna_fits(inroute + infile_catalogo1,3)

from astropy.io import fits
#-----
#   Creamos las columnas del fichero de salida.

```

```

#-----
col0 = fits.Column(name='HATLAS_DR1_CATALOGUE', format='23A'
                    , array=outfile_array_HATLAS_DR1_CATALOGUE)
col1 = fits.Column(name='RA_H', format='D', array=outfile_array_RA_h
                    , unit='Degrees')
col2 = fits.Column(name='DEC_H', format='D', array=outfile_array_DEC_h
                    , unit='Degrees')

col3 = fits.Column(name='S250', format='D', array=get.columna_fits(inroute
                        + infile_catalogo1,4), unit='mJy')
col4 = fits.Column(name='S350', format='D', array=get.columna_fits(inroute
                        + infile_catalogo1,5), unit='mJy')
col5 = fits.Column(name='S500', format='D', array=get.columna_fits(inroute
                        + infile_catalogo1,6), unit='mJy')

col6 = fits.Column(name='E250', format='D', array=get.columna_fits(inroute
                        + infile_catalogo1,7), unit='mJy')
col7 = fits.Column(name='E350', format='D', array=get.columna_fits(inroute
                        + infile_catalogo1,8), unit='mJy')
col8 = fits.Column(name='E500', format='D', array=get.columna_fits(inroute
                        + infile_catalogo1,9), unit='mJy')

# Desplazamiento al rojo calculado.
col9 = fits.Column(name='Z_AJUSTE', format='D', array=z_ajuste)
col10 = fits.Column(name='Error_Z_AJUSTE', format='D', array=E_z_ajuste)

# Información proporcionada por HATLAS.
col11 = fits.Column(name='z_SPEC', format='D', array=z_spec)
col12 = fits.Column(name='z_QUAL', format='D', array=z_qual)
col13 = fits.Column(name='GSQ_FLAG', format='D', array=gsq_flag)

# Desplazamientos al rojo considerados válidos.
col14 = fits.Column(name='Z_H', format='D', array=z_hatlas)
col15 = fits.Column(name='Error_Z_H', format='D', array=E_z_hatlas)
col16 = fits.Column(name='Flag_hatlas', format='D', array=flag_hatlas)

# Ahora fabrico un objeto con esas columnas:
cols = fits.ColDefs([col0, col1, col2, col3, col4, col5, col6, col7, col8
                    , col9, col10, col11, col12, col13, col14, col15, col16])

# Ese objeto se mete en una tabla HDU.
tbhdu = fits.BinTableHDU.from_columns(cols)

tbhdu.writeto(outroute + outfile,overwrite=True)

#=====
b = datetime.datetime.now()
print('El tiempo de ejecución total ha sido de ',b-tiempo_inicio)

```

C.5 Módulo main_matching_gh.py

```
# -*- coding: utf-8 -*-
"""
Este ejecutable realiza el emparejamiento entre las observaciones de los
catálogos GAMA y HATLAS e identifica los candidatos que forman un sistema
lente gravitatoria según nuestro criterio. También proporciona varias gráficas
que ayudan al análisis de los resultados.

Python 3.6.0    scipy 0.18    matplotlib 2.0.0    numpy 1.11.3    astropy 1.3

@author: Javier Gutiérrez Solórzano
"""

#=====
#   Nombres de los ficheros de entrada/salida y su localización.
#=====
#-----
#   Directorio en el que se encuentran los módulos.
#-----
import sys
sys.path.append("../packages/")
#-----
#   Directorios de los ficheros.
#-----
inroute_g = './outroute_gama/'
inroute_h = './outroute_hatlas/'
outroute = './outroute_matching/'
#-----
#   Nombre de los ficheros
#-----
infile_catálogo1 = "Gama.fits"
infile_catálogo2 = "HAtlas.fits"
infile_objetos_interesantes="nombres_candidatos.txt"
outfile = "matching_gh"

#   Soporte de unidades físicas.
from astropy import units as u

#   Para el cálculo del factor de bayes espectroscópico.
z_max=3.501

import numpy as np
#   Resolución angular de los catálogos a partir de la FWHM.
fwhm_sigma=2*np.sqrt(2*np.log(2))
sg=(0.7/fwhm_sigma)*u.arcsec    # GAMA
sh=(17.98/fwhm_sigma)*u.arcsec  # H-ATLAS

#   Separación angular máxima entre los objetos que forman el "matching".
sep=54*u.arcsec
```

```

#=====
#  Lectura de los datos necesarios para el matching.
#=====

import get
import rojo

#-----
#  Preparación del catálogo GAMA.
#-----
#  Todos los valores del redshift de los objetos de GAMA.
z_g=get.columna_fits(inroute_g + infile_catalogo1,5)

#  Nos devuelve el identificador (posición en el fichero) de aquellos objetos
#  de GAMA de los que disponemos de una medida del redshift válida.
idg_catalogo1 = rojo.id_z_validos(z_g)

# Posiciones de todos los objetos del catálogo GAMA
ra_c_GAMA=get.columna_fits(inroute_g + infile_catalogo1,1)
dec_c_GAMA=get.columna_fits(inroute_g + infile_catalogo1,2)

#  Posiciones de aquellos objetos de GAMA con z válidos.
ra_catalogo_GAMA = get.ordena_Array(idg_catalogo1,ra_c_GAMA) # RA (0 y 360)
dec_catalogo_GAMA = get.ordena_Array(idg_catalogo1,dec_c_GAMA) # DEC (-90 y 90)

#-----
#  Preparación del catálogo HATLAS.
#-----
#  Todos los valores del redshift de los objetos de HATLAS.
z_h=get.columna_fits(inroute_h + infile_catalogo2,14)

#  Identificador de aquellos observaciones de HATLAS con z válido.
idh_catalogo2 = rojo.id_z_validos(z_h)

# Posiciones de todos los objetos del catálogo HATLAS.
ra_c_HATLAS=get.columna_fits(inroute_h + infile_catalogo2,1)
dec_c_HATLAS=get.columna_fits(inroute_h + infile_catalogo2,2)

#  Posiciones de los objetos HATLAS.
ra_catalogo_HATLAS = get.ordena_Array(idh_catalogo2,ra_c_HATLAS)
dec_catalogo_HATLAS = get.ordena_Array(idh_catalogo2,dec_c_HATLAS)

#=====
#  Matching para identificar aquellas observaciones de H-Atlas que se encuen-
#  tran a una distancia angular inferior a "sep" de otra observación en GAMA.
#=====

from astropy.coordinates import SkyCoord

h = SkyCoord(ra=ra_catalogo_HATLAS*u.degree, dec=dec_catalogo_HATLAS*u.degree)
g = SkyCoord(ra=ra_catalogo_GAMA*u.degree, dec=dec_catalogo_GAMA*u.degree)

```

```

# d2d   Separación angular entre los objetos que forman el "matching".
# _ indica que no queremos realizar ese cálculo.
[idg_arrays, idh_arrays, d2d, _] = h.search_around_sky(g, sep)

# Posiciones de las observaciones que forman el "matching" en los
# catálogos de entrada (idxg,idxh):
idxg=get.ordena_Array(idg_arrays, idg_catalogo1)
idxh=get.ordena_Array(idh_arrays, idh_catalogo2)

# El primer elemento del array en Python es un 0 mientras que en la tabla
# fits comienza por un 1. Las dos líneas siguientes resuelven esto.
idg=get.posicion_fits(idxg)
idh=get.posicion_fits(idxh)

#=====
# Factores de Bayes: posicional, fotométrico y conjunto
#=====

# GAMA
sg=sg.to(u.radian)
sg=sg.value
zg = get.ordena_Array(idxg,z_g)
szg = get.ordena_Array(idxg,get.columna_fits(inroute_g + infile_catalogo1,6))
flag_g = get.ordena_Array(idxg,get.columna_fits(inroute_g + infile_catalogo1,
                                                7))

# HALTAS
sh=sh.to(u.radian)
sh=sh.value
zh = get.ordena_Array(idxh,z_h)
szh = get.ordena_Array(idxh,get.columna_fits(inroute_h + infile_catalogo2,15))
flag_h = get.ordena_Array(idxh,get.columna_fits(inroute_h + infile_catalogo2,
                                                16))

# Distancias angulares en radianes
d2d = d2d.to(u.radian)
d2d = d2d.value

import factor_bayes
# sh      número   error posicional del catalogo1
# sg      número   error posicional en el catalogo2
# d2d     array    distancias entre los objetos
# zh      array    valores de z en el catalogo1
# szh     array    error de z de los objetos del catalogo1
# zg      array    valores de z sobre los objetos del catalogo2
# szg     array    error de z sobre los objetos del catalogo2
# z_max   número   valor maximo considerado

[bayes_posicional, bayes_z, bayes_conjunto] = factor_bayes.bayes_conjunto(sh,
                                                                    sg,d2d,zh,szh,zg,szg,z_max)

```

```

#=====
#  Creación del fichero fits con los datos del matching completo
#=====

#  outfile_array_IDNAME_GAMA = idxg
outfile_array_GAMA_IAU_ID = get.ordena_Array(idxg,get.columna_fits(inroute_g
                                                                    + infile_catalogo1,0))
outfile_array_RA_g = get.ordena_Array(idxg,get.columna_fits(inroute_g
                                                                    + infile_catalogo1,1))
outfile_array_DEC_g = get.ordena_Array(idxg,get.columna_fits(inroute_g
                                                                    + infile_catalogo1,2))

#  outfile_array_IDNAME_HATLAS = idxh
outfile_array_HATLAS_DR1_CATALOGUE = get.ordena_Array(idxh
                                                                    ,get.columna_fits(inroute_h + infile_catalogo2,0))
outfile_array_RA_h = get.ordena_Array(idxh
                                                                    ,get.columna_fits(inroute_h + infile_catalogo2,1))
outfile_array_DEC_h = get.ordena_Array(idxh
                                                                    ,get.columna_fits(inroute_h + infile_catalogo2,2))

outfile_array_S250=get.ordena_Array(idxh
                                                                    ,get.columna_fits(inroute_h + infile_catalogo2,3))
outfile_array_S350=get.ordena_Array(idxh
                                                                    ,get.columna_fits(inroute_h + infile_catalogo2,4))
outfile_array_S500=get.ordena_Array(idxh
                                                                    ,get.columna_fits(inroute_h + infile_catalogo2,5))

from astropy.io import fits
#-----
#  Creamos las columnas del fichero de salida.
#-----
col0 = fits.Column(name='idxg', format='14A', array=idxg)
col1 = fits.Column(name='GAMA_IAU_ID', format='23A'
                    , array=outfile_array_GAMA_IAU_ID)
col2 = fits.Column(name='RA_G', format='D', array=outfile_array_RA_g
                    , unit='Degrees')
col3 = fits.Column(name='DEC_G', format='D'
                    , array=outfile_array_DEC_g, unit='Degrees')

col4 = fits.Column(name='idxh', format='14A', array=idxh)
col5 = fits.Column(name='HATLAS_DR1_CATALOGUE', format='23A'
                    , array=outfile_array_HATLAS_DR1_CATALOGUE)
col6 = fits.Column(name='RA_H', format='D'
                    , array=outfile_array_RA_h, unit='Degrees')
col7 = fits.Column(name='DEC_H', format='D'
                    , array=outfile_array_DEC_h, unit='Degrees')

col8 = fits.Column(name='S250', format='D'
                    , array=outfile_array_S250, unit='mJy')

```

```

col9 = fits.Column(name='S350', format='D'
                    , array=outfile_array_S350, unit='mJy')
col10 = fits.Column(name='S500', format='D'
                     , array=outfile_array_S500, unit='mJy')

col11 = fits.Column(name='E250', format='D'
                    , array=get_ordena_Array(idXH,get.columna_fits(inroute_h + infile_catalogo2,6))
                    , unit='mJy')
col12 = fits.Column(name='E350', format='D'
                    , array=get_ordena_Array(idXH,get.columna_fits(inroute_h + infile_catalogo2,7))
                    , unit='mJy')
col13 = fits.Column(name='E500', format='D'
                    , array=get_ordena_Array(idXH,get.columna_fits(inroute_h + infile_catalogo2,8))
                    , unit='mJy')

col14 = fits.Column(name='Dist. angular', format='D', array=d2d,unit='Degrees')
# Desplazamiento al rojo.
col15 = fits.Column(name='Z_H', format='D', array=zh)
col16 = fits.Column(name='Error_Z_H', format='D', array=szh)
col17 = fits.Column(name='flag_H', format='D', array=flag_h)

col18 = fits.Column(name='Z_G', format='D', array=zg)
col19 = fits.Column(name='Error_Z_G', format='D', array=szg)
col20 = fits.Column(name='flag_G', format='D', array=flag_g)

# Factores de bayes.
col21 = fits.Column(name='Bayes_posicional', format='D'
                    , array=bayes_posicional)
col22 = fits.Column(name='Bayes_z', format='D', array=bayes_z)
col23 = fits.Column(name='Bayes_conjunto', format='D', array=bayes_conjunto)

# Ahora fabrico un objeto con esas columnas:
cols = fits.ColDefs([col0, col1, col2, col3, col4, col5, col6, col7, col8,
                     col9, col10, col11, col12, col13, col14, col15, col16,
                     col17, col18, col19, col20, col21, col22, col23])

# Ese objeto se mete en una tabla HDU.
tbhdu = fits.BinTableHDU.from_columns(cols)

# Y al final lo escribimos en un fichero "outfile" en la ruta "outroute".
tbhdu.writeto(outroute + outfile + '.fits',overwrite=True)

#=====
# Representación de los factores de Bayes.
#=====

# Devolvemos las distancias angulares del matching a arcsec.
d2d = d2d*u.radian
d2d = d2d.to(u.arcsec)
d2d = d2d.value

```



```

import matplotlib.pyplot as plt

#-----
#   Factor de Bayes posicional vs separación angular.
#-----
figura=plt.figure(num = None, figsize = (12, 6.4), dpi = 80, facecolor = 'w',
                  edgecolor = 'k')

plt.scatter(d2d, bayes_posicional, color="blue",linewidth=0.05, s=1)

plt.xlabel('$\{\phi\}_{ij}$ [ $\mathrm{arcsec}$ ]', fontsize = 14, color='blue')
plt.ylabel('$B^{\{p\}}_{12}$', fontsize = 16, color='blue')

plt.axhline(y=1, linewidth=1, color='k', linestyle='--')

plt.yscale('log', lenthreshx=0.1)

#   Cambiamos los límites de los ejes.
plt.axis([0, 55, 0, 1.5e9])

plt.show()
figura.savefig(outroute + outfile + 'bayes_pos' + outfile, dpi=150
              , transparent=True)

print("\nDistancia angular para para la cual el factor de Bayes posicional "
      +"toma el valor 1, B_p=1, fi=",
      (np.sqrt(-2*(sg**2+sh**2)*(np.log((sg**2+sh**2)/2)))*u.radian).to(u.arcsec))

#-----
#   Factor de Bayes espectroscópico vs zh-zg.
#-----
# Se hace una copia del array original para no modificar sus valores.
import copy
bayes_z_representacion = copy.copy(bayes_z)

# Para poder representar aquellos valores con factor de Bayes fotométrico
# inferior a 1e-10.
for i in range(0,len(bayes_z),1):
    if np.any(bayes_z[i]<2e-20):
        bayes_z_representacion[i]=2e-20

zh_g=[]
for i in range(0,len(zh),1):
    zh_g.append(zh[i]-zg[i])

figura=plt.figure(num = None, figsize = (12, 6.4), dpi = 80, facecolor = 'w',
                  edgecolor = 'k')

plt.scatter(zh_g, bayes_z_representacion, color="blue", linewidth=0.1, s=2)

```

```

plt.grid(False)
plt.xlabel('$z_{\{h\}}-z_{\{g\}}$', fontsize = 16, color='blue')
plt.ylabel('$B^{\{z\}}_{\{12\}}$', fontsize = 16, color='blue')

plt.axhline(y=1, linewidth=1, color='k', linestyle='--')

plt.yscale('log', linthreshx=0.1)

# Cambiamos los limites de los ejes.
plt.axis([-1, 3.5, 1e-20, 15000])

plt.show()
figura.savefig(outroute + outfile + 'bayes_z' + outfile, dpi=150
               , transparent=True)

#=====
# Representación posición de las observaciones, con los que se va a realizar
# el matching.
#=====

#-----
# G09 Y BLOQUE 2
#-----
figura=plt.figure(num = None, figsize = (15,5.3), dpi = 800, facecolor = 'w'
                  , edgecolor = 'k')

plt.scatter(ra_c_HATLAS, dec_c_HATLAS, color="blue", linewidth=0.1, s=2,
            label="HATLAS")
plt.scatter(ra_c_GAMA, dec_c_GAMA, color="green", linewidth=0.1, s=2,
            label="GAMA")

plt.grid(False)
plt.xlabel('RA [degree]', fontsize = 16, color='blue')
plt.ylabel('DEC [degree]', fontsize = 16, color='blue')

# Límites de los ejes.
plt.axis([127, 142, -2.2, 3.1])

plt.show()
figura.savefig(outroute + outfile + 'region_1', dpi=150, transparent=True)

[area_hatlas,area_gama,area_interseccion]=get.area_region(127,142,-2.1,3.3
,ra_c_GAMA,dec_c_GAMA,ra_c_HATLAS,dec_c_HATLAS,0.08)

print(area_hatlas,area_gama,area_interseccion)

#-----
# G12 Y BLOQUE 3
#-----

```

```

figura=plt.figure(num = None, figsize = (15,5.3), dpi = 800, facecolor = 'w'
                  , edgecolor = 'k')

plt.scatter(ra_c_HATLAS, dec_c_HATLAS, color="blue", linewidth=0.1, s=2,
            label="HATLAS")
plt.scatter(ra_catalogo_GAMA, dec_catalogo_GAMA, color="green", linewidth=0.1,
            s=2, label="GAMA")

plt.grid(False)
plt.xlabel('RA [degree]', fontsize = 16, color='blue')
plt.ylabel('DEC [degree]', fontsize = 16, color='blue')

# Limites de los ejes.
plt.axis([172, 187, -3.1, 2.2])

plt.show()
figura.savefig(outroute + outfile + 'region_2', dpi=150, transparent=True)

[area_hatlas,area_gama,area_interseccion]=get.area_region(172, 187, -3.1, 2.2
,ra_c_GAMA,dec_c_GAMA,ra_c_HATLAS,dec_c_HATLAS,0.08)

print(area_hatlas,area_gama,area_interseccion)

#-----
# G15 Y BLOQUE 4
#-----
figura=plt.figure(num = None, figsize = (15,5.3), dpi = 800, facecolor = 'w'
                  , edgecolor = 'k')

plt.scatter(ra_c_HATLAS, dec_c_HATLAS, color="blue", linewidth=0.1, s=2,
            label="HATLAS")
plt.scatter(ra_c_GAMA, dec_c_GAMA, color="green", linewidth=0.1, s=2,
            label="GAMA")

plt.grid(False)
plt.xlabel('RA [degree]', fontsize = 16, color='blue')
plt.ylabel('DEC [degree]', fontsize = 16, color='blue')

# Limites de los ejes.
plt.axis([210, 225, -2.2, 3.1])

plt.show()
figura.savefig(outroute + outfile + 'region_3', dpi=150, transparent=True)

[area_hatlas,area_gama,area_interseccion]=get.area_region(210, 225, -2.2, 3.1
,ra_c_GAMA,dec_c_GAMA,ra_c_HATLAS,dec_c_HATLAS,0.08)

print(area_hatlas,area_gama,area_interseccion)

#=====

```

```

# Número de HALOS en HATLAS_DR1_CATALOGUE_V1
#=====

s250=get.columna_fits(inroute_h + infile_catalogo2,3)
s350=get.columna_fits(inroute_h + infile_catalogo2,4)
s500=get.columna_fits(inroute_h + infile_catalogo2,5)

print("\nNúmero de HALOs en HATLAS_DR1 es de: (Negrello,Gonzalez,Ambos)="
      , get.numero_halos(idXH,s250,s350,s500))

#=====
# Histogramas de los redshifts
#=====

from astropy.visualization import hist as hist_astropy
#-----
# Histograma redshifts catálogo GAMA con los que se va a realizar el matching.
#-----
idg_pos_o=get.id_pos(z_g)
flag_g_o=get.columna_fits(inroute_g + infile_catalogo1,7)

# _amg (antes matching gama).
[id_spec11_amg,id_spec12_amg,id_spec13_amg,id_disabled_amg
 ,idx_amg]=get.obj_by_flag_gama(idg_pos_o,flag_g_o,"array")

z_spec11_amg=get.ordena_Array(id_spec11_amg, z_g)
z_spec12_amg=get.ordena_Array(id_spec12_amg, z_g)
z_spec13_amg=get.ordena_Array(id_spec13_amg, z_g)
z_amg=get.ordena_Array(idx_amg, z_g)

figura=plt.figure(num = None, figsize = (9, 6), dpi = 80, facecolor = 'w'
                  , edgecolor = 'k')

# "knuth"      "scott"      "freedman"      "blocks"
hist_astropy(z_spec12_amg, bins=100, range=(0,0.6), normed=False, alpha=0.6
             , color='#2e7d32', histtype='stepfilled', label='$z_{g}$ Q=2')
hist_astropy(z_spec11_amg, bins=100, range=(0,0.6), normed=False, alpha=0.6
             , color='#558b2f', histtype='stepfilled', label='$z_{g}$ Q=1')
hist_astropy(z_spec13_amg, bins=100, range=(0,0.6), normed=False, alpha=0.6
             , color='#9e9d24', histtype='stepfilled', label='$z_{g}$ Q=3')
hist_astropy(z_amg, bins=100, range=(0,0.6), normed=False, alpha=0.7
             , color='black', histtype='step', label='$z_{g}$')

plt.xlabel('$z_{g}$', fontsize = 16)
plt.ylabel('$\mathrm{N}(z_{g})$', fontsize = 16)

plt.legend(loc='best')

figura.savefig(outroute + outfile + 'histograma_z_gama_disponible.png'
              , dpi=150, transparent=True)

```

```

plt.show()

print("(Q=1,Q=2,Q=3,disabled,total) =",get.obj_by_flag_gama(idg_pos_o,
                                                             flag_g_o,"numero"))

#-----
#   Histograma redshifts catálogo GAMA que forman parte del matching.
#-----
# _dmg (despues matching gama).
[id_spec11_dmg,id_spec12_dmg,id_spec13_dmg,id_disabled_dmg
 ,idx_dmg]=get.obj_by_flag_gama(idxg,flag_g,"array")

z_spec11_dmg=get.ordena_Array(id_spec11_dmg, z_g)
z_spec12_dmg=get.ordena_Array(id_spec12_dmg, z_g)
z_spec13_dmg=get.ordena_Array(id_spec13_dmg, z_g)
z_dmg=get.ordena_Array(idspec_dmg, z_g)

figura=plt.figure(num = None, figsize = (9, 6), dpi = 80, facecolor = 'w'
                  , edgecolor = 'k')

#   "knuth"      "scott"      "freedman"      "blocks"
hist_astropy(z_spec12_dmg, bins=100, range=(0,0.6), normed=False, alpha=0.6
             , color='#2e7d32', histtype='stepfilled', label='$z_{g}$ Q=2')
hist_astropy(z_spec11_dmg, bins=100, range=(0,0.6), normed=False, alpha=0.6
             , color='#558b2f', histtype='stepfilled', label='$z_{g}$ Q=1')
hist_astropy(z_spec13_dmg, bins=100, range=(0,0.6), normed=False, alpha=0.6
             , color='#9e9d24', histtype='stepfilled', label='$z_{g}$ Q=3')
hist_astropy(z_dmg, bins=100, range=(0,0.6), normed=False, alpha=0.7
             , color='black', histtype='step', label='$z_{g}$')

plt.xlabel('$z_{g}$', fontsize = 16)
plt.ylabel('$\mathrm{N}(z_{g})$', fontsize = 16)

plt.legend(loc='best')

figura.savefig(outroute + outfile + 'histograma_z_gama_matching.png', dpi=150
              , transparent=True)
plt.show()

print("(Q=1,Q=2,Q=3,disabled,total) =",get.obj_by_flag_gama(idxg,flag_g,
                                                             "numero"))

#-----
#   Histograma redshifts catálogo HATLAS con los que se va a realizar el
#   matching.
#-----
idh_pos=get.id_pos(z_h)
flag_h_o=get.columna_fits(inroute_h + infile_catalogo2,16)

# _amh (antes matching hatlas)
[id_spec_amh,id_annz_amh,id_phot_amh,id_disabled_amh
 ,idx_amh]=get.obj_by_flag_hatlas(idh_pos,flag_h_o,"array")

```

```

z_spec_amh=get.ordena_Array(id_spec_amh, z_h)
z_annz_amh=get.ordena_Array(id_annz_amh, z_h)
z_phot_amh=get.ordena_Array(id_phot_amh, z_h)
z_amh=get.ordena_Array(idz_amh, z_h)

figura=plt.figure(num = None, figsize = (9, 6), dpi = 80, facecolor = 'w'
                  , edgecolor = 'k')

#    "knuth"      "scott"      "freedman"      "blocks"
hist_astropy(z_spec_amh, bins=100, range=(0,3.5), normed=False, alpha=0.6
             , color='green', histtype='stepfilled', label='$z_{h}^{\mathrm{\{;spec\}}}$')
hist_astropy(z_phot_amh, bins=100, range=(0,3.5), normed=False, alpha=0.3
             , color='red', histtype='stepfilled', label='$z_{h}^{\mathrm{\{;phot\}}}$')
hist_astropy(z_annz_amh, bins=100, range=(0,3.5), normed=False, alpha=0.6
             , color='blue', histtype='stepfilled', label='$z_{h}^{\mathrm{\{;ANNZ\}}}$')
hist_astropy(z_amh, bins=100, range=(0,3.5), normed=False, alpha=0.7
             , color='black', histtype='step', label='$\{z\}_{h}$')

plt.xlabel('$z_{h}$', fontsize = 16)
plt.ylabel('$\mathrm{N}(z_{h})$', fontsize = 16)

plt.legend(loc='best')

figura.savefig(outroute + outfile + 'histograma_z_hatlas_disponible.png'
              , dpi=150, transparent=True)
plt.show()

print("(z_spec,z_ANNZ,z_phot,disabled,total) =",get.obj_by_flag_hatlas(idh_pos,
                                flag_h_o,"numero"))

#-----
#    Histograma redshifts catálogo HALTAS que forman parte del matching.
#-----
# _dmh (después matching haltlas)
[id_spec_dmh,id_annz_dmh,id_phot_dmh,id_disabled_dmh
 ,idx_dmh]=get.obj_by_flag_hatlas(idhx,flag_h,"array")

z_spec_dmh=get.ordena_Array(id_spec_dmh, z_h)
z_annz_dmh=get.ordena_Array(id_annz_dmh, z_h)
z_phot_dmh=get.ordena_Array(id_phot_dmh, z_h)
z_dmh=get.ordena_Array(idz_dmh, z_h)

figura=plt.figure(num = None, figsize = (9, 6), dpi = 80, facecolor = 'w'
                  , edgecolor = 'k')

#    "knuth"      "scott"      "freedman"      "blocks"
hist_astropy(z_spec_dmh, bins=100, range=(0,3.5), normed=False, alpha=0.6
             , color='green', histtype='stepfilled', label='$z_{h}^{\mathrm{\{;spec\}}}$')
hist_astropy(z_phot_dmh, bins=100, range=(0,3.5), normed=False, alpha=0.3

```

```

    , color='red', histtype='stepfilled', label='$z_{\mathrm{phot}}$')
hist_astropy(z_annz_dmh, bins=100, range=(0,3.5), normed=False, alpha=0.6
    , color='blue', histtype='stepfilled', label='$z_{\mathrm{ANNZ}}$')
hist_astropy(z_dmh, bins=100, range=(0,3.5), normed=False, alpha=0.7
    , color='black', histtype='step', label='$z_{\mathrm{h}}$')

plt.xlabel('$z_{\mathrm{h}}$', fontsize = 16)
plt.ylabel('$\mathrm{N}(z_{\mathrm{h}})$', fontsize = 16)

plt.legend(loc='best')

figura.savefig(outroute + outfile + 'histograma_z_hatlas_matching.png'
    , dpi=150, transparent=True)
plt.show()

print("(z_spec,z_ANNZ,z_phot,disabled,total) =",get_obj_by_flag_hatlas(idhx,
    flag_h,"numero"))

#=====
#   Factor de Bayes fotométrico vs posicional .
#=====

#   Se puede incluir en un fichero el nombre de aquellos objetos de Hatlas de
#   los que queremos destacar el pareado en el que participan por algún motivo.
#   Hay que poner el nombre completo; Ej: "HATLAS J083945.1+023440"
#
#   En principio ninguno => fichero vacío.

fichero_con_nombres=[]
with open(inroute_h + infile_objetos_interesantes,'r') as fichero_nombres:
    fichero_con_nombres=fichero_nombres.readlines()

nombres_resaltar=[]
for o in range (0,len(fichero_con_nombres),1):
    #   Parte cada string del array en dos, por donde esta el elemento.
    #   que le pasamos en split.
    nombres_confirmados_consaltolinea=fichero_con_nombres[o].split('\n')
    nombres_resaltar.append(nombres_confirmados_consaltolinea[0])

#   Necesitmos estos arrays para la representación de dos rectas en las dos
#   figuras siguientes.
x = np.arange(-10, 11, 0.1)
y1 = -1*x
y100 = -1*x+2

#-----
#   Representación de aquellos emparejamientos en las que participa una
#   observación de HATLAS cuyo z ha sido obtenido mediante "z_phot_hatlas".
#-----

```

```

[x_0, y_0, x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, x_5, y_5] = \
    get.representacion_candidatos(bayes_posicional,bayes_z, bayes_conjunto,
    outfile_array_HATLAS_DR1_CATALOGUE,nombres_resaltar,outfile_array_S250,
    outfile_array_S350,outfile_array_S500,flag_h,zh,zg,"ajuste")

figura=plt.figure(num = None, figsize = (12,3), dpi = 800, facecolor = 'w'
    , edgecolor = 'k')

plt.scatter(x_3, y_3, color="#ffea00", linewidth=0.1, s=10,
    label="Resto de objetos")
plt.scatter(x_4, y_4, color="#64dd17", linewidth=0.1, s=10,
    label="Bayes mayor que 100")
plt.scatter(x_5, y_5, color="cyan", linewidth=0.1, s=10,
    label="Bayes mayor que 1")
plt.scatter(x_0, y_0, color="black", linewidth=0.1, s=40,
    label="Objetos del fichero")
plt.scatter(x_1, y_1, color="red", linewidth=0.1, s=30,
    label="Negrello et al.")
plt.scatter(x_2, y_2, color="blue", linewidth=0.1, s=30,
    label="Gonzalez-Nuevo et al.")
plt.axvline(x=0, linewidth=1, color='k')
plt.axhline(y=0, linewidth=1, color='k')

plt.plot(x, y1, color='k', linestyle='-.')
plt.plot(x, y100, 'k--')

plt.grid(False)
plt.ylabel('$\mathrm{\log_{10}}\{({B}^z_{12})\}$', fontsize = 16, color='blue')

# Limites de los ejes. [xmin,xmax,ymin,ymax]
plt.axis([-1.8, 9.25, -30, -1])

plt.yscale('symlog', lenthreshx=0.1)

plt.show()
figura.savefig(outroute + outfile + 'factor_bayes_log_semilog_ajuste', dpi=150,
    transparent=True)

#-----
# Todas las observaciones
#-----
[x_0, y_0, x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, x_5
    , y_5] = get.representacion_candidatos(bayes_posicional,bayes_z,bayes_conjunto
    ,outfile_array_HATLAS_DR1_CATALOGUE,nombres_resaltar,outfile_array_S250
    ,outfile_array_S350,outfile_array_S500,flag_h,zh,zg,"todos")

figura=plt.figure(num = None, figsize = (12,9), dpi = 800, facecolor = 'w',
    edgecolor = 'k')

plt.scatter(x_3, y_3, color="#ffea00", linewidth=0.1, s=10,

```



```

        label="Resto de objetos")
plt.scatter(x_4, y_4, color="#64dd17", linewidth=0.1, s=10,
            label="Bayes mayor que 100")
plt.scatter(x_5, y_5, color="cyan", linewidth=0.1, s=10,
            label="Bayes mayor que 1")
plt.scatter(x_0, y_0, color="black", linewidth=0.1, s=40,
            label="Objetos del fichero")
plt.scatter(x_1, y_1, color="red", linewidth=0.1, s=30,
            label="Negrello et al.")
plt.scatter(x_2, y_2, color="blue", linewidth=0.1, s=30,
            label="Gonzalez-Nuevo et al.")
plt.axvline(x=0, linewidth=1, color='k')
plt.axhline(y=0, linewidth=1, color='k')

plt.plot(x, y1, color='k', linestyle='-.')
plt.plot(x, y100, 'k--')

plt.grid(False)
plt.xlabel('$\mathrm{\log_{10}}(\{B\}^{\{p\}}_{12})$', fontsize = 16, color='blue')
plt.ylabel('$\mathrm{\log_{10}}(\{B\}^{\{z\}}_{12})$', fontsize = 16, color='blue')
# Limites de los ejes. [xmin,xmax,ymin,ymax]
plt.axis([-1.8, 9.25, -110, 1.1])

plt.yscale('symlog', linthreshx=0.1)

plt.show()
figura.savefig(outroute + outfile + 'factor_bayes_log_semilog', dpi=150
              , transparent=True)

[id_halo_h_espec,id_halo_h_fot,id_halo_negrello_gonzalez]= \
    get.selecion_candidatos(outfile_array_S250,outfile_array_S350,
        outfile_array_S500,flag_h,zh,zg,bayes_posicional, bayes_conjunto)

print("Hay",len(id_halo_h_espec),len(id_halo_h_fot),
      "candidatos segun nuestro criterio y",len(id_halo_negrello_gonzalez),
      "cumplen ambos criterios")

#=====
# Selección de los candidatos a lente gravitatoria .
#=====

# Identificador de todos los emparejamientos que cumplen nuestro criterio
id_halo=id_halo_h_espec+id_halo_h_fot

idg_halo=get.ordena_Array(id_halo,idg)
idh_halo=get.ordena_Array(id_halo,idh)

#-----
# Creamos los arrays que faltan del fichero de salida
#-----

```

```

nombre_gama_halo=get.ordena_Array(id_halo,outfile_array_GAMA_IAU_ID)
nombre_hatlas_halo=get.ordena_Array(id_halo,outfile_array_HATLAS_DR1_CATALOGUE)

z_h_halo=get.ordena_Array(id_halo,zh)
ez_h_halo=get.ordena_Array(id_halo,szh)
flag_h_halo=get.ordena_Array(id_halo,flag_h)

# Los flujos se encuantran en mJy,
s250_halo=get.ordena_Array(id_halo,outfile_array_S250)*u.Jy
s350_halo=get.ordena_Array(id_halo,outfile_array_S350)*u.Jy
s500_halo=get.ordena_Array(id_halo,outfile_array_S500)*u.Jy
s250_halo=s250_halo.to(u.mJy)
s350_halo=s350_halo.to(u.mJy)
s500_halo=s500_halo.to(u.mJy)

z_g_halo=get.ordena_Array(id_halo,zg)
ez_g_halo=get.ordena_Array(id_halo,szg)
flag_g_halo=get.ordena_Array(id_halo,flag_g)

bayes_posicional_halo=get.ordena_Array(id_halo,bayes_posicional)
bayes_z_halo=get.ordena_Array(id_halo,bayes_z)
bayes_conjunto_halo=get.ordena_Array(id_halo,bayes_conjunto)

#-----
#   Creamos las columnas del fichero de salida.
#-----
cl0 = fits.Column(name='iddxg', format='14A', array=idg_halo)
cl1 = fits.Column(name='GAMA_IAU_ID', format='23A', array=nombre_gama_halo)
cl2 = fits.Column(name='iddxh', format='14A', array=idh_halo)
cl3 = fits.Column(name='HATLAS_DR1_CATALOGUE', format='23A'
                  , array=nombre_hatlas_halo)

cl4 = fits.Column(name='S250', format='D', array=s250_halo, unit='mJy')
cl5 = fits.Column(name='S350', format='D', array=s350_halo, unit='mJy')
cl6 = fits.Column(name='S500', format='D', array=s500_halo, unit='mJy')

#   Desplazamiento al rojo.
cl7 = fits.Column(name='Z_H', format='D', array=z_h_halo)
cl8 = fits.Column(name='Error_Z_H', format='D', array=ez_h_halo)
cl9 = fits.Column(name='flag_H', format='D', array=flag_h_halo)

cl10 = fits.Column(name='Z_G', format='D', array=z_g_halo)
cl11 = fits.Column(name='Error_Z_G', format='D', array=ez_g_halo)
cl12 = fits.Column(name='flag_G', format='D', array=flag_g_halo)

#   Factores de bayes.
cl13 = fits.Column(name='Bayes_posicional', format='D'
                  , array=bayes_posicional_halo)
cl14 = fits.Column(name='Bayes_z', format='D', array=bayes_z_halo)
cl15 = fits.Column(name='Bayes_conjunto', format='D'

```

```

        , array=bayes_conjunto_halo)

cls = fits.ColDefs([cl0, cl1, cl2, cl3, cl4, cl5, cl6, cl7, cl8, cl9, cl10
                    , cl11, cl12, cl13, cl14, cl15])

tbhdu = fits.BinTableHDU.from_columns(cls)

tbhdu.writeto(outroute + outfile + '_candidatos.fits',overwrite=True)

```

C.6 Módulo main_matching_gh_simulacion.py

```

# -*- coding: utf-8 -*-
"""
Este es el ejecutable con el que se realizan las simulaciones. Para ejecutarse es necesario disponer de los ficheros de salida de los ejecutables main_gama.py y main_hatlas.py . Para realizar la simulación a cada observación se le ha reasignado una posición aleatoria sobre una región con un área equivalente al área del catálogo al que pertenece. Después estas dos regiones se han solapado en un área igual al área de intersección de los catálogos originales y se ha procedido a realizar un conteo de las contrapartidas con la función "get.seleccion_candidatos".

    Python 3.6.0      numpy 1.11.3      astropy 1.3

@author: Javier Gutiérrez Solórzano
"""

#=====
#   Nombres de los ficheros de entrada/salida y su localización.
#=====
#-----
#   Directorio en el que se encuentran los módulos.
#-----
import sys
sys.path.append("../packages/")
#-----
#   Directorios de los ficheros.
#-----
inroute_g = './outroute_gama/'
inroute_h = './outroute_hatlas/'
outroute = './outroute_matching/'
#-----
#   Nombre de los ficheros
#-----
infile_catálogo1 = "Gama.fits"
infile_catálogo2 = "HAtlas.fits"
infile_objetos_interesantes="nombres_candidatos.txt"
outfile = "matching_gh"

#   Soporte de unidades físicas.

```

```

from astropy import units as u

# Para el cálculo del factor de bayes espectroscópico.
z_max=3.501

import numpy as np

fwhm_sigma=2*np.sqrt(2*np.log(2))

# Separación angular máxima entre los objetos que forman el "matching".
sep=50*u.arcsec

import get
import rojo

# Límites de las áreas en las que se realiza la simulación
# GAMA
DEC_a_gama = 0*u.deg
DEC_b_gama = 12*u.deg

RA_a_gama = 0*u.deg

area_gama = 144*u.deg*u.deg
area_gama_sr = (area_gama.to(u.sr)).value
RA_b_gama_rad = (area_gama_sr/np.sin(DEC_b_gama))*u.rad
RA_b_gama = RA_b_gama_rad.to(u.deg)

# HATLAS
DEC_hatlas = 12*u.deg

area_hatlas = 161*u.deg*u.deg
area_hatlas_sr = (area_hatlas.to(u.sr)).value
RA_hatlas_rad = (area_hatlas_sr/np.sin(DEC_hatlas))*u.rad
RA_hatlas = RA_hatlas_rad.to(u.deg)

# Intersección
DEC_interseccion = 12*u.deg

area_interseccion = 130*u.deg*u.deg
area_interseccion_sr = (area_interseccion.to(u.sr)).value
RA_interseccion_rad = (area_interseccion_sr/np.sin(DEC_interseccion))*u.rad
RA_interseccion = RA_interseccion_rad.to(u.deg)

DEC_a_hatlas = 0*u.deg
DEC_b_hatlas = 12*u.deg

RA_a_hatlas = RA_b_gama-RA_interseccion
RA_b_hatlas = RA_a_hatlas+RA_hatlas

```

```

#-----
#   Preparación del catálogo GAMA.
#-----
#   Todos los valores del redshift de los objetos de GAMA.
z_g=get.columna_fits(inroute_g + infile_catalogo1,5)

#   Nos devuelve el identificador (posición en el fichero) de aquellos objetos
#   de GAMA de los que disponemos de una medida del redshift válida.
idg_catalogo1 = rojo.id_z_validos(z_g)

#-----
#   Preparación del catálogo HATLAS.
#-----
#   Todos los valores del redshift de los objetos de HATLAS.
z_h=get.columna_fits(inroute_h + infile_catalogo2,14)

#   Nos devuelve el identificador de aquellos objetos de HATLAS de los que
#   disponemos de una medida del redshift válida.
idh_catalogo2 = rojo.id_z_validos(z_h)

id_halo_h_espec_media=[]
id_halo_h_fot_media=[]
id_halo_h_media=[]
id_halo_negrello_gonzalez_media=[]

from astropy.coordinates import SkyCoord
import factor_bayes

for i in range(0,1000,1):

    [ra_catalogo_GAMA,dec_catalogo_GAMA]= get.uniform_spherical_distribution(
        RA_a_gama.value,RA_b_gama.value,DEC_a_gama.value,DEC_b_gama.value,
        len(idg_catalogo1))

    [ra_catalogo_HATLAS,dec_catalogo_HATLAS] = \
        get.uniform_spherical_distribution(RA_a_hatlas.value,RA_b_hatlas.value,
        DEC_a_hatlas.value,DEC_b_hatlas.value,len(idh_catalogo2))

    #=====
    #   Matching para identificar aquellos objetos del catálogo H-Atlas que se
    #   encuentran a una distancia angular inferior a "sep" segundos de arco de
    #   un objeto de GAMA.
    #=====
    h = SkyCoord(ra=ra_catalogo_HATLAS*u.degree,
                  dec=dec_catalogo_HATLAS*u.degree)
    g = SkyCoord(ra=ra_catalogo_GAMA*u.degree, dec=dec_catalogo_GAMA*u.degree)

    # d2d   Separación angular entre los objetos que forman el "matching".
    # _ indica que no queremos realizar ese cálculo.
    [idg_arrays, idh_arrays, d2d, _] = h.search_around_sky(g, sep)

```

```

# Posiciones de los objetos en el catalogos de entrada (idxg,idxh):
idxg=get.ordena_Array(idg_arrays, idg_catalogo1)
idxh=get.ordena_Array(idh_arrays, idh_catalogo2)

# El primer elemento del array en Python es un 0 mientras que el la tabla
# fits comienza por un 1. Las dos líneas siguientes resuelven ésto.
idg=get.posicion_fits(idxg)
idh=get.posicion_fits(idxh)

#=====
# Factores de Bayes: posicional, fotométrico y conjunto
#=====
# GAMA
# Errores posicionales del catálogo GAMA.
sg=(0.7/fwhm_sigma)*u.arcsec
sg=sg.to(u.radian)
sg=sg.value

zg = get.ordena_Array(idxg,z_g)
szg = get.ordena_Array(idxg,
                        get.columna_fits(inroute_g + infile_catalogo1,6))
flag_g = get.ordena_Array(idxg,
                        get.columna_fits(inroute_g + infile_catalogo1,7))

# HALTAS
# Errores posicionales del catálogo HALTAS.
sh=(17.98/fwhm_sigma)*u.arcsec
sh=sh.to(u.radian)
sh=sh.value

zh = get.ordena_Array(idxh,z_h)
szh = get.ordena_Array(idxh,
                        get.columna_fits(inroute_h + infile_catalogo2,15))
flag_h = get.ordena_Array(idxh,
                        get.columna_fits(inroute_h + infile_catalogo2,16))
# Distancias angulares en radianes (necesario para calcular el factor de
# bayes posicional).
d2d = d2d.to(u.radian)
d2d = d2d.value

# Cálculo de los factores de Bayes
[bayes_posicional, bayes_z, bayes_conjunto] = factor_bayes.bayes_conjunto(
    sh,sg,d2d,zh,szh,zg,szg,z_max)

outfile_array_S250=get.ordena_Array(idxh
    ,get.columna_fits(inroute_h + infile_catalogo2,3))
outfile_array_S350=get.ordena_Array(idxh
    ,get.columna_fits(inroute_h + infile_catalogo2,4))
outfile_array_S500=get.ordena_Array(idxh
    ,get.columna_fits(inroute_h + infile_catalogo2,5))

```

```

[id_halo_h_espec,id_halo_h_fot,id_halo_negrello_gonzalez]= \
    get.selecion_candidatos(outfile_array_S250,outfile_array_S350,
        outfile_array_S500,flag_h,zh,zg,bayes_posicional,bayes_conjunto)

print(len(id_halo_h_espec)+len(id_halo_h_fot),len(id_halo_h_espec),
    len(id_halo_h_fot),len(id_halo_negrello_gonzalez))

id_halo_h_espec_media.append(len(id_halo_h_espec))
id_halo_h_fot_media.append(len(id_halo_h_fot))
id_halo_h_media.append(len(id_halo_h_espec)+len(id_halo_h_fot))
id_halo_negrello_gonzalez_media.append(len(id_halo_negrello_gonzalez))

print("-Número total de candidatos:",np.mean(id_halo_h_media),chr(177),
    np.std(id_halo_h_media),"n-HATLAS fotométrico:",
    np.mean(id_halo_h_espec_media), chr(177),np.std(id_halo_h_espec_media),
    "n-HATLAS espectroscópico:",
    np.mean(id_halo_h_fot_media),chr(177),np.std(id_halo_h_fot_media),
    "n-Candidatos que cumplen los criterios de Gonzalez y/o Negrello:",
    np.mean(id_halo_negrello_gonzalez_media),chr(177),
    np.std(id_halo_negrello_gonzalez_media))

```

C.7 Módulo factor_bayes.py

```

# -*- coding: utf-8 -*-
"""

```

Contiene las funciones que se utilizan para calcular los factores de Bayes.

Python 3.6.0 numpy 1.11.3

```

@author: Javier Gutiérrez Solórzano
"""

```

```

import math
import numpy as np

```

```

def bayes_objeto(s1,s2,psi,z1,sz1,z2,sz2,z_max):
    """

```

Permite el cálculo del factor de Bayes posicional, fotométrico y conjunto para cada emparejamiento encontrado. Esta función contiene dos funciones; una para calcular el factor de Bayes posicional y otra para el factor de Bayes fotométrico. El factor de Bayes conjunto se calcula como el producto de ambos.

Parámetros:

Se explican en el comentario de cada sub-función.

Return:

bp ----- Factor de Bayes posicional.

```

bz ----- Factor de Bayes fotométrico.
bp*bz -- Factor de Bayes conjunto.
"""
def bayesPosicional(s1,s2,psi):
    """
    s1 --- Desviación estándar de las medidas posicionales de uno de los
           catálogos.
    s2 --- Desviación estándar de las medidas posicionales del segundo
           catálogo.
    psi -- Distancia angular a la que se encuentran los objetos que
           forman el emparejamiento.
    """
    x_1=(s1**2+s2**2)
    bayes_posicional=(2/x_1)*math.exp(-( psi**2)/(2*x_1) ))

    return bayes_posicional

def bayesZ(sz1,sz2,z_max,z1,z2):
    """
    sz1 ---- Desviación estándar asociada a z1.
    sz2 ---- Desviación estándar asociada a z2.
    z_max --
    z1 ----- Valor del redshift de los objetos de uno de los catálogos.
    z2 ----- Valor del redshift del segundo catálogo.
    """
    if sz1>0 and sz2>0 and z_max>0 and z1>0 and z2>0 and z_max>z1 \
        and z_max>z2:

        sz12=sz1**2
        sz22=sz2**2

        f_error_1_numerador=math.erf((sz12*z2+sz22*z1)/(sz1*sz2*\
            math.sqrt(2*(sz12+sz22))))
        f_error_2_numerador=math.erf((sz12*(z2-z_max)+sz22*(z1-z_max))/\
            (sz1*sz2*math.sqrt(2*(sz12+sz22))))
        numerador=math.sqrt(2/math.pi)*z_max*math.exp(-((z1-z2)**2)/\
            (sz12+sz22))*(f_error_1_numerador-f_error_2_numerador)

        dif_error1_denominador=math.erf(z1/sz1*math.sqrt(2))- \
            math.erf((z1-z_max)/(sz1*math.sqrt(2)))
        dif_error2_denominador=math.erf(z2/sz2*math.sqrt(2))- \
            math.erf((z2-z_max)/(sz2*math.sqrt(2)))
        denominador=math.sqrt(sz12+sz22)* \
            dif_error1_denominador*dif_error2_denominador

        if np.any(denominador==0):
            denominador=1e-80

        bayes_z=numerador/denominador

```



```

        return bayes_z
    else:
        # Si no se cumplen las condiciones, Bz no nos aporta información
        return 1

bp=bayesPosicional(s1,s2,psi)
bz=bayesZ(sz1,sz2,z_max,z1,z2)

return bp, bz, bp*bz

def bayes_conjunto(s1,s2,psi,z1,sz1,z2,sz2,z_max):
    """
    Aplica la función bayes_objeto a un conjunto de emparejados.

    Parámetros:
    Se explican en el comentario de las funciones "bayesPosicional" y "bayesZ".

    Return:
    bp ----- Array que contiene los factores de Bayes posicionales.
    bz ----- Array que contiene los factores de Bayes fotométricos.
    bp*bz -- Array que contiene los factores de Bayes conjuntos.

    """
    array_bayes_posicional=[]
    array_bayes_z=[]
    array_bayes=[]
    for i in range(0,len(psi),1):
        [bayes_posicional, bayes_z, conjunto] = bayes_objeto(s1,s2,psi[i]
        ,z1[i],sz1[i],z2[i],sz2[i],z_max)
        array_bayes_posicional.append(bayes_posicional)
        array_bayes_z.append(bayes_z)
        array_bayes.append(conjunto)

    return array_bayes_posicional, array_bayes_z, array_bayes

```

C.8 Módulo rojo.py

```

# -*- coding: utf-8 -*-

"""
Este módulo contiene las funciones con las que se ha obtenido el redshift
de las observaciones de los catálogos H-ATLAS y GAMA.

Python 3.6.0      scipy 0.18      numpy 1.11.3

@author: Javier Gutiérrez Solórzano
"""

from scipy.optimize import curve_fit

```

```

from scipy.interpolate import interp1d
import numpy as np

def z_phot_objeto(x_sed, y_sed, x_experimental, y_experimental_i, y_error_i,
                  abs_s=True):
    """
    Ajusta (utilizando dos parámetros) la SED teórica de la galaxia modelo
    a las medias experimentales del flujo espectroscópico.

    Parámetros:
    x_sed ----- Array con los valores 'x' de la SED teórica.
    y_sed ----- Array con los valores 'y' de la SED teórica.
    x_experimental --- Array de tres elementos con los valores de la longitud
                        de onda a la que se realizan las medidas del flujo.
                        Nosotros siempre utilizaremos el mismo array que esta
                        definido en el modulo 'main.py':
                        [250,350,500]*u.micron
    y_experimental_i-- Array que contiene las medidas del flujo un objeto. En
                        En este trabajo se dispone de tres medidas para cada
                        objeto, asi que cada array contendrá 3 elementos
    y_error_i----- Array que contiene error del flujo asociado al array
                        'y_experimental'.

    Return:
    popt----- Valores óptimos del ajuste.
    np.sqrt(np.diag(pcov))-- Las diagonales proporcionan la varianza de la
                            de los parámetros de ajuste.

    """
    #-----
    # Definición de la funcion desplazada.
    #-----
    def func_desplazada(x,K,C):
        """
        Obtiene una función mediante interpolación lineal del los puntos y
        multiplica los valores asociados a cada uno de los ejes de coordenadas
        'x' e 'y' por un factor ('K' y 'C' respectivamente).

        Parámetros:
        K -- Factor multiplicativo para el eje de abscisas.
        C -- Factor multiplicativo para el eje de ordenadas.

        Return:
        y -- Valores de la función obtenida por interpolación lineal
            asociados al eje de ordenadas tras el ajuste.

        """
        x_desplazada = x_sed*(K)
        f1 = interp1d(x_desplazada,y_sed,bounds_error=False,
                      fill_value=np.min(y_sed))
        y = f1(x)*C

```

```

    return y
    # p0=[1.8, 1e-3] Limita el los valores en los que se realiza
    # el ajuste, aunque en principio no es necesario.

popt, pcov = curve_fit(lambda x, K, C: func_desplazada(x, K, C),
                        x_experimental, y_experimental_i, p0=[1.8, 1e-4],
                        sigma=y_error_i)

return popt, np.sqrt(np.diag(pcov))

def z_phot_hatlas(x_sed,y_sed,x_experimental,y_experimental,y_error):
    """
    Hace uso de la función 'objeto' para obtener el redshift de todo el
    catálogo HATLAS.

    Parámetros:
    x_sed ----- Array con los valores 'x' de la SED teórica.
    y_sed ----- Array con los valores 'y' de la SED teórica.
    x_experimental -- Array de tres elementos con los valores de la longitud
                        de onda a la que se realizan las medidas del flujo.
                        Nosotros siempre utilizaremos el mismo array que está
                        definido en el modulo 'main.py':
                        [250,350,500]*u.micron
    y_experimental -- Array que contiene las medidas del flujo de un conjunto
                        de objetos. Cada elemento del array debe de ser un array
                        con las medidas de flujo de un objeto.
    y_error ----- Array que contiene error del flujo asociado al array
                        'y_experimental'.

    Return:
    z -- Array con los desplazamientos al rojo del catálogo HATLAS completo.
    S -- Desviación estándar como medida de la calidad del ajuste.

    """
    z=[]
    S=[]
    # Cambiar len(y_experimental) por otro valor inferior para las pruebas si
    # el catálogo es muy extenso.
    for i in range(0,len(y_experimental),1):
        [z_objeto,S_objeto] = z_phot_objeto(x_sed, y_sed, x_experimental,
        y_experimental[i], y_error[i])

        z.append(z_objeto[0]-1)
        S.append(S_objeto[0])

    return z, S

def id_z_validos(z_catalogo):
    """

```

Proporciona el identificador de aquellas observaciones del catálogo HATLAS con z "válido".

Parámetros:

$z_catalogo$ ----- Redshifts proporcionados por HATLAS.

Return:

$contador_h$ ----- Posición que ocupa la observación en el catálogo HATLAS.

"""

```
contador_h=[]
for i in range(0,len(z_catalogo),1):
    if np.any(z_catalogo[i]!=-99):
        contador_h.append(i)
return contador_h
```

```
def z_hatlas(gsq_flag,z_spec,z_qual,z_ajuste,z_min,z_max):
```

"""

Esta función se utiliza para asignar los valores definitivos a los redshifts a las observaciones realizadas por HATLAS. Además añade una etiqueta que permite identificar el origen de esa medida.

Parámetros:

gsq_flag ---- Clasificación del tipo de objeto proporcionada por HATLAS.

z_spec ----- Redshift proporcionado por HATLAS.

z_qual ----- Etiqueta de calidad asignada por HATLAS.

z_ajuste ---- Estimación del redshift proporcionado por el ajuste de la función " z_phot_hatlas ".

z_min ----- Valor mínimo aceptable para el z del ajuste.

z_max ----- Valor máximo aceptable para el z del ajuste.

Return:

z_hatlas ----- Array con los valores definitivos de z asignados a las observaciones de HATLAS.

σ_z_hatlas -- Errores asignados a las medias de " z_hatlas ".

$flag$ ----- Etiqueta de "calidad" sobre la medida del redshift.

* 11, 12, 13: medidas "espectroscópicas" procedentes del catálogo HATLAS.

* 2: redshift fotométrico generado con ANNZ.

* 3: z obtenido mediante la función " z_phot_hatlas ".

* -99 para el resto de casos.

"""

```
z_hatlas=[]
sigma_z_hatlas=[]
flag=[]
for i in range(0,len(z_ajuste),1):
    #0=gal, 1=star, 2=quasar, 3=quasar candidate based on colour selection.
    if np.any(gsq_flag[i]==0) or np.any(gsq_flag[i]==2) \
```

```

        or np.any(gsq_flag[i]==3):
# >=3 is OK to use (redshift espectroscópico)
if np.any(z_qual[i]>=3):
    z_hatlas.append(z_spec[i])
    # Puede haber redshifts negativos, asi que abs()!
    sigma_z_hatlas.append(0.00011064*(1+np.abs(z_spec[i])))
    if np.any(z_qual[i]==5):
        flag.append(11)
    elif np.any(z_qual[i]==4):
        flag.append(12)
    else:
        flag.append(12)
# <3 (redshift fotométrico generado con ANNZ)
elif (np.any(z_qual[i]==2) or np.any(z_qual[i]==1)) \
      and (np.any(z_spec[i]>=0) and np.any(z_spec[i]<=0.7)):
    z_hatlas.append(z_spec[i])
    sigma_z_hatlas.append(0.023)
    flag.append(2)
# Es una galaxia de la que no se ha podido obtener el redshift por
# ninguno de los métodos anteriores.
elif np.any(gsq_flag[i]==0) and np.any(z_ajuste[i] >= z_min) \
      and np.any(z_ajuste[i] <= z_max) :
    z_hatlas.append(z_ajuste[i])
    sigma_z_hatlas.append(0.115*(1+z_ajuste[i]))
    flag.append(3)
else:
    z_hatlas.append(-99)
    sigma_z_hatlas.append(-99)
    flag.append(-99)
else:
    z_hatlas.append(-99)
    sigma_z_hatlas.append(-99)
    flag.append(-99)

return z_hatlas, sigma_z_hatlas, flag

```

```

def z_gama(z_HELIO,z_quality):
    """

```

Esta función se utiliza para obtener los valores del redshift espectroscópico que nos proporciona el proyecto GAMA. Los valores de z asociados a los objetos del catálogo GAMA se encuentran en una columna del fichero 'GamaCoreDR1_v1.fits', pero resulta que no todos los valores que aparecen aquí son válidos. El proyecto indica los z no válidos del siguiente modo:

- Si el corrimiento al rojo de objeto es desconocido, '9999'*
- Si se divulgará en una publicación posterior, ' 2'*
- Si z se ha observado pero es una medida pobre, ' 0.9'*

Esta función "filtra" aquellos valores no válidos y proporciona el array con los valores que si podemos utilizar.

Por otra parte, resulta necesario asociar un identificador a cada observación del catálogo GAMA; un entero que indica la fila que ocupaba la observación en el fichero 'GamaCoreDR1_v1.fits'.

Parámetros:

z_HELIO ----- Valores del redshift proporcionados por GAMA.

z_quality ----- Factor de calidad asignado a z por GAMA.

Return:

z_gama ----- Redshifts que se consideramos válidos.

sigma_z_gama ----- Error que hemos asignado a cada valor de "z_gama".

flag ----- Etiqueta de calidad sobre la medida del redshift,

** 11, 12, 13 para medidas espectroscópicas*

** -99 resto de casos.*

"""

z_gama=[]

sigma_z_gama=[]

flag=[]

for i in range(0,len(z_HELIO),1):

if np.any(z_HELIO[i]!=-2) and np.any(z_HELIO[i]!=9999):

if np.any(z_quality[i]>=3):

z_gama.append(z_HELIO[i])

Puede haber redshifts negativos, asi que abs()!

sigma_z_gama.append(0.00011064(1+np.abs(z_HELIO[i])))*

if np.any(z_quality[i]==5):

flag.append(11)

elif np.any(z_quality[i]==4):

flag.append(12)

else:

flag.append(13)

else:

z_gama.append(-99)

sigma_z_gama.append(-99)

flag.append(-99)

else:

z_gama.append(-99)

sigma_z_gama.append(-99)

flag.append(-99)

return z_gama, sigma_z_gama, flag

C.9 Módulo get.py

-- coding: utf-8 -*-*

"""

Módulo del programa que contiene la mayoría de las funciones que utilizan en programas (salvo las relativas a la obtención del redshift y de la obtención

de los factores de Bayes). Se ha decidido hacer así porque muchas de estas funciones se utilizan en varios módulos; además, se facilita la comprensión de los otros módulos y se evita que el programador se pierda en los detalles.

Python 3.6.0 numpy 1.11.3 astropy 1.3

@author: Javier Gutiérrez Solórzano

"""

```
from astropy.io import fits
import numpy as np
```

```
def sed_func(fichero_sed):
```

"""

Función para leer el fichero que contiene los puntos SED teórica de la galaxia 'SMMJ2135-0102' que consiste en dos columnas: la primera con los valores de longitudes de onda y la segunda valores de la densidad espectral de flujo F_{λ} .

Parámetros:

fichero_sed -- Nombre del fichero.

Return:

x_teorica ---- Array, primera columna del fichero.

y_teorica ---- Array, segunda columna del fichero.

"""

```
x_teorica=[]
y_teorica=[]
```

```
file=open(fichero_sed,'r')
```

```
contenido=file.readlines()
```

```
# Obtenemos el número de líneas que tiene el fichero (¡cuidado con
# ficheros muy grandes, pueden dar problemas con el uso de la memoria!)
```

```
lineas = len(open(fichero_sed).readlines())
```

```
for i in range (0,lineas,1):
```

```
    dato1=contenido[i].find(' ')# (separador espacio ' ')
```

```
    dato2=contenido[i].find('\n',dato1+1) # (salto de línea '\n')
```

```
    # Cambia el tipo de cada elemento.
```

```
    valor1=float(contenido[i][0:dato1])
```

```
    valor2=float(contenido[i][dato1+1:dato2])
```

```
    x_teorica.append(valor1)
```

```
    y_teorica.append(valor2)
```

```
file.close()
```

```
return x_teorica, y_teorica
```

```

def columna_fits(catalogo,n_columna):
    """
    La lectura de las columnas individuales de un fichero .fits

    Parámetros:
    catalogo ---- Nombre del fichero.
    n_columnas -- Posición de la columna

    Return:
    Array con el contenido de la columna 'n_columna' del fichero 'catalogo'.
    """

    return fits.open(catalogo, memmap=True)[1].data.field(n_columna)

def datos_experimentales(inroute_infile_datos_experimentales):
    """
    Lee las columnas que contienen las medidas del flujo del
    catálogo 'HATLAS_DR1_CATALOGUE_V1.2.FITS' y devuelve los datos en un
    formato adecuado para usarse por las funciones definidas en el módulo
    'rojo.py'.

    Cada objeto astronómico del catálogo contiene tres medidas del flujo, en
    las longitudes de onda 250, 350 y 500 micrómetros (columnas 4, 5 y 6) y con
    un error asociado a cada una de ellas (columnas 7, 8 y 9).

    Parámetros:
    inroute_infile_datos_experimentales -- Nombre del fichero.

    Return:
    y_experimental -- Array cuyos elementos son un array de 3 elementos con
                    las medidas del flujo de la observación.
    y_error ----- Array cuyos elementos son un array de 3 elementos con el
                    "error" del flujo.
    """

    # Carga archivo fits.
    fitsFile = fits.open(inroute_infile_datos_experimentales, memmap=True)

    # Seleccionamos toda la tabla.
    tbdata = fitsFile[1].data

    # Transformamos una columna del archivo de entrada, infile, en un
    # array. Éste nos proporciona las condiciones de ejecución (a partir del
    # tamaño del array) al bucle for.
    longitud_infile_array=len(np.array(tbdata.field(5)))

    #-----
    # ARRAYS DE DATOS EXPERIMENTALES.

```



```
#-----
```

```
y_experimental=[]
for i in range(0,longitud_infile_array,1):
    y_experimental_objeto=[]
    y_experimental_objeto.append(tbddata[i].field(4))
    y_experimental_objeto.append(tbddata[i].field(5))
    y_experimental_objeto.append(tbddata[i].field(6))

    y_experimental.append(y_experimental_objeto)
```

```
y_error=[]
for i in range(0,longitud_infile_array,1):
    y_error_objeto=[]
    y_error_objeto.append(tbddata[i].field(7))
    y_error_objeto.append(tbddata[i].field(8))
    y_error_objeto.append(tbddata[i].field(9))

    y_error.append(y_error_objeto)
```

```
return y_experimental, y_error
```

```
def ordena_Array(idxa,array_datos):
```

```
    """
```

Reordena un array a partir de los valores numéricos de otro que contiene números enteros. El elemento que ocupa la posición [i] en el array toma la posición array_datos[idxa[i]] en el array de salida. La longitud de array_datos debe ser mayor o igual que el mayor de los idxa[i].

Debe cumplirse siempre que: len(array_datos)-1 <= max(idxa)

Ejemplo:

idxa = [0, 6, 3, 4, 5, 5, 4, 5]

array_datos = ["a","b","c","d","e","f","g"]

ordena_Array(idxa,array_datos) = ['a', 'g', 'd', 'e', 'f', 'f', 'e', 'f']

Parámetros:

idxa ----- Array que contiene números enteros.

array_datos -- Array con los elementos que queremos ordenar.

Return:

nuevo_orden -- Array que contiene los elementos de array_datos ordenados según el orden del array idxa.

```
    """
```

```
nuevo_orden=[]
for i in range(0,len(idxa),1):
    nuevo_orden.append(array_datos[idxa[i]])
```

```
return nuevo_orden
```

```
def elimina_elementos_repetidos(idx):
```

```
    """
```

Ordena los valores numéricos de un array de menor a mayor y deja un solo valor de aquellos que se repiten.

Ejemplo:

idx=[0, 9, 3, 4, 5, 5, 4, 9]

elimina_elementos_repetidos(idx)=[0, 3, 4, 5, 9]

Parámetros:

idx ----- Array con números enteros.

Return:

b ----- Array que contiene los mismos números que idx ordenado de menor a mayor y sin elementos repetidos.

```
    """
```

```
    idx_sort=sorted(idx)
```

```
    b=[]
```

```
    if len(idx)==0:
```

```
        return b
```

```
    else:
```

```
        b.append(idx[0])
```

```
        for i in range(0,len(idx),1):
```

```
            if idx_sort[i] != b[len(b)-1]:
```

```
                b.append(idx_sort[i])
```

```
        return b
```

```
def id_pos(lista):
```

```
    """
```

Permite obtener un array de número naturales de la misma longitud que el array de entrada.

Parámetros:

lista ----- Array cualquiera.

Return:

contador_i -- Array de números naturales de longitud len(lista).

```
    """
```

```
    contador_i=[]
```

```
    for i in range(0,len(lista),1):
```

```
        contador_i.append(i)
```

```
    return contador_i
```

```

def posicion_fits(idx):
    """
    Sirve para encontrar la fila que ocupa un objeto en uno de los catálogos
    astronómicos. Esta función es necesaria porque el primer elemento de un
    array en Python ocupa la posición 0 mientras que la primera fila del
    catálogo, tiene asignado el valor 1.

    Parámetros:
    idx ----- Array de posiciones, siendo el primer elemento 0.

    Return:
    id_fits -- Array de posiciones, siendo el primer elemento 1.

    """
    id_fits=[]
    for i in range(0,len(idx),1):
        id_fits.append(idx[i]+1)

    return id_fits


def frange(start, end, step):
    """
    El comportamiento de esta función es el mismo que el de la función "range"
    pero, a diferencia de ésta, permite que cualquiera de los parámetros de
    entrada no sea un entero.

    Parámetros:
    start -- Limite inferior de la secuencia (incluido en la misma).
    end ---- Límite superior de la secuencia (no está incluido).
    step --- Diferencia entre cada número de la secuencia.

    """
    tmp = start
    while(tmp < end):
        yield tmp
        tmp += step


from astropy.coordinates import SkyCoord
from astropy import units as u


def area_region(x_inf,x_sup,y_inf,y_sup,ra_catologo_GAMA,dec_catologo_GAMA
                ,ra_catologo_HATLAS,dec_catologo_HATLAS,paso):
    """
    Permite obtener el ángulo sólido de las regiones en las que se encuentran
    los objetos del catálogo HATLAS, GAMA y la zona de intersección. Para ello
    se crea una cuadrícula cuyos elementos están separados a una distancia del
    orden de la separación angular de los objetos de los catálogos; después se
    realiza un conteo de aquellos puntos de la cuadrícula que tienen objetos de

```

un determinado catálogo (o ambos, para determinar la zona de intersección) a una distancia la mitad de la diagonal de cuadrado que conforma la celda ($\text{paso} * (\text{np.sqrt}(2)) * (-1)$) que es la distancia mínima posible para que todo el espacio quede cubierto. Para obtener el ángulo sólido de la región, se multiplica el número de objetos por el área de cada celda.

Parámetros:

x_inf ----- Límite horizontal inferior de la cuadrícula.
x_sup ----- Límite horizontal superior de la cuadrícula.
y_inf ----- Límite vertical inferior de la cuadrícula.
y_sup ----- Límite vertical superior de la cuadrícula.
ra_catálogo_GAMA ----- Ascensión recta de los objetos de GAMA.
dec_catálogo_GAMA ----- Declinación de los objetos de GAMA.
ra_catálogo_HATLAS --- Ascensión recta de los objetos de HATLAS.
dec_catálogo_HATLAS -- Declinación de los objetos de HATLAS.
paso ----- Longitud del lado de las celdas que forman la cuadrícula.

Return:

area_hatlas ----- Ángulo sólido que cubren los objetos de HATLAS.
area_gama ----- Ángulo sólido que cubren los objetos de GAMA.
area_interseccion ---- Ángulo sólido que cubren los objetos que forman la intersección de ambos catálogos.
 """

```
def regilla(x_inf,x_sup,y_inf,y_sup,paso):
    # Creo una regilla.
    px=[]
    py=[]
    for y in frange(y_inf,y_sup,paso):
        for x in frange(x_inf,x_sup,paso):
            py.append(y)
            px.append(x)

    return px, py
[px,py]=regilla(x_inf,x_sup,y_inf,y_sup,paso)
# Área de cada de cada elemento del grid.
area=(paso*u.degree)**2
# De esta forma nos aseguramos de cubrir todo el espacio del grid.
sep=paso*((np.sqrt(2))*(-1))*u.degree
# Asignamos unidades a la regilla.
grid = SkyCoord(ra=px*u.degree, dec=py*u.degree)

h = SkyCoord(ra=ra_catálogo_HATLAS*u.degree, dec=dec_catálogo_HATLAS*u.degree)
[id_grid_h, _, _, _] = h.search_around_sky(grid, sep)
area_hatlas=len(elimina_elementos_repetidos(id_grid_h))*area

g = SkyCoord(ra=ra_catálogo_GAMA*u.degree, dec=dec_catálogo_GAMA*u.degree)
[id_grid_g, _, _, _] = g.search_around_sky(grid, sep)
area_gama=len(elimina_elementos_repetidos(id_grid_g))*area
```

```

area_interseccion=len(list(set(id_grid_g).intersection(id_grid_h)))*area

return area_hatlas,area_gama,area_interseccion

def numero_halos(idXH,s250,s350,s500):
    """
    Realiza un conteo de todos aquellos objetos del catálogo HATLAS que cumplen
    los criterios de Gonzalez-Nuevo et al. y de Negrello et al. para ser
    lente gravitatoria partiendo de las medidas del instrumento SPIRE.

    Parámetros:
    idXH -- Array con las posiciones de los objetos de HATLAS.
    s250 -- Medidas del flujo en 250 microm.
    s350 -- Medidas del flujo en 350 microm.
    s500 -- Medidas del flujo en 500 microm.

    Return:
    len(id_halo_negrello) -- Número de objetos tipo Negrello et al.
    len(id_halo_gonzalez) -- Número de objetos tipo Gonzalez-Nuevo et al.
    len(id_halo_ambos) ----- Número de objetos que cumplen ambas condiciones.

    """
    id_halo_negrello=[]
    id_halo_gonzalez=[]
    id_halo_ambos=[]
    for i in range(0,len(idXH),1):
        negrello=False
        gonzalez=False
        # Para evitar divisiones por 0 (menor que la precisión de la
        # medida del instrumento SPIRE)
        if np.any(s250[i] == 0):
            s250[i]=1e-6
        if np.any(s350[i] == 0):
            s350[i]=1e-6
        s350_s250 = s350[i]/s250[i]
        s500_s350 = s500[i]/s350[i]

        if np.any(s500[i] >= 0.1):
            negrello=True
        if np.any(s350[i] >= 0.085) and np.any(s250[i] >= 0.035) \
            and np.any(s350_s250 > 0.6) and np.any(s500_s350 > 0.4):
            gonzalez=True

        if negrello==True:
            id_halo_negrello.append(idXH[i])
        if gonzalez==True:
            id_halo_gonzalez.append(idXH[i])
        if negrello==True and gonzalez==True:

```

```

        id_halo_ambos.append(idXH[i])

    return len(id_halo_negrello), len(id_halo_gonzalez), len(id_halo_ambos)

def obj_by_flag_hatlas(idx, flag, tipo):
    """
    Realiza un conteo de los objetos del catálogo HATLAS en función de la
    calidad de la medida del redshift de que se dispone y nos devuelve un array
    o un número dependiendo de la variable "tipo" que hemos escogido. (la
    medida de la calidad, para cada objeto, se indica por el valor de flag[i]).

    Parámetros:
    idx --- Array con las posiciones de los objetos de HATLAS.
    flag -- Array con la etiqueta sobre la calidad del redshift.
    tipo -- Toma el valor "array" si queremos que nos devuelva los
            identificadores de los objetos o bien el valor "numero" si estamos
            interesados en el número de objetos con una determinada medida del
            factor de calidad.

    Return:
    return 1 -- Devuelve las longitudes de los arrays "return 2".
    return 2 -- Contiene los identificadores de los objetos, en 4 arrays
                diferentes, dependiendo de si su redshift ha sido obtenido
                a partir de una medida espectroscópica (id_spec), si se ha
                obtenido a partir de ANNZ (id_annz), ha sido obtenida mediante
                el ajuste propuesto en este trabajo (id_phot) o bien no está
                disponible (id_disabled). El último elemento que devuelve es
                el array con los identificadores de todos los objetos
                considerados.
    return 3 -- En caso de haber introducido un valor incorrecto en el
                parámetro "tipo".

    """
    id_spec=[]
    id_annz=[]
    id_phot=[]
    id_disabled=[]

    for i in range(0, len(flag), 1):
        if flag[i]==11 or flag[i]==12 or flag[i]==13:
            id_spec.append(idXH[i])
        elif flag[i]==2:
            id_annz.append(idXH[i])
        elif flag[i]==3:
            id_phot.append(idXH[i])
        else:
            id_disabled.append(idXH[i])

    if tipo=="numero":
        return len(elimina_elementos_repetidos(id_spec)), \

```

```

        len(elimina_elementos_repetidos(id_annz)), \
        len(elimina_elementos_repetidos(id_phot)), \
        len(elimina_elementos_repetidos(id_disabled)), \
        len(elimina_elementos_repetidos(idx))
elif tipo=="array":
    return elimina_elementos_repetidos(id_spec), \
           elimina_elementos_repetidos(id_annz), \
           elimina_elementos_repetidos(id_phot), \
           elimina_elementos_repetidos(id_disabled), \
           elimina_elementos_repetidos(idx)
else:
    return "la variable tipo debe ser 'numero' o 'array' "

def obj_by_flag_gama(idx,flag,tipo):
    """
    Realiza un conteo de los objetos del catálogo GAMA en función de la
    calidad de la medida del redshift de que se dispone y nos devuelve un array
    o un número dependiendo de la variable "tipo" que hemos escogido.

    Parámetros:
    idx --- Array con las posiciones de los objetos de GAMA.
    flag -- Array con la etiqueta sobre la calidad del redshift.
    tipo -- Toma el valor "array" si queremos que nos devuelva los
            identificadores de los objetos o bien el número de observaciones
            con una determinada medida del factor de calidad.

    Return:
    return 1 -- Devuelve las longitudes de los arrays "return 2".
    return 2 -- Contiene los identificadores de los objetos, en 4 arrays
                diferentes, dependiendo de si su redshift ha sido obtenido
                a partir de una medida espectroscópica (id_spec11), si se ha
                obtenido a partir de ANNZ (id_spec12), ha sido obtenida mediante
                el ajuste propuesto en este trabajo (id_spec13) o bien no está
                disponible (id_disabled). El último elemento que devuelve es
                el array con los identificadores de todos los objetos
                considerados.
    return 3 -- En caso de haber introducido un valor incorrecto en el
                parámetro "tipo", nos indica los valores posibles que puede
                tomar.
    """
    id_spec11=[]
    id_spec12=[]
    id_spec13=[]
    id_disabled=[]

    for i in range(0,len(flag),1):
        if flag[i]==11:
            id_spec11.append(idx[i])
        elif flag[i]==12:

```

```

        id_spec12.append(idx[i])
    elif flag[i]==13:
        id_spec13.append(idx[i])
    else:
        id_disabled.append(idx[i])

if tipo=="numero":
    return len(elimina_elementos_repetidos(id_spec11)), \
           len(elimina_elementos_repetidos(id_spec12)), \
           len(elimina_elementos_repetidos(id_spec13)), \
           len(elimina_elementos_repetidos(id_disabled)), \
           len(elimina_elementos_repetidos(idx))
elif tipo=="array":
    return elimina_elementos_repetidos(id_spec11), \
           elimina_elementos_repetidos(id_spec12), \
           elimina_elementos_repetidos(id_spec13), \
           elimina_elementos_repetidos(id_disabled), \
           elimina_elementos_repetidos(idx)
else:
    return "tipo debe ser 'numero' o 'array' "

import copy

def representacion_candidatos(bp,bz_original,bc,nombre_hatlas,
nombres_confirmados,s250_original,s350_original,s500,flag,zh,zg,seleccion):
    """
    Esta función clasifica los emparejamientos y agrupa sus factores de Bayes
    en varios arrays para realizar una representación adecuada de los mismos.

    Parámetros:
    bp ----- Factor de Bayes posicional.
    bz_original ----- Factor de Bayes fotométrico.
    bc ----- Factor de Bayes conjunto.
    nombre_hatlas ----- Nombres de los objetos HATLAS que han participado en
                          el matching.
    nombres_confirmados -- Nombres que de los objetos de los que queremos
                          destacar el pareado en el que participan.
    s250_original ----- Medidas del fujo en 250 microm.
    s350_original ----- Medidas del fujo en 350 microm.
    s500 ----- Medidas del fujo en 500 microm.
    flag ----- Array con la etiqueta sobre la calidad del redshift
                de los objetos de HATLAS.
    zh ----- Redshift de la observación de HATLAS.
    zg ----- Redshift de la observación de GAMA.
    seleccion ----- Nos permite seleccionar solo aquellos objetos del
                     catálogo HATLAS cuyo redshift ha sido obtenido a
                     partir del ajuste propuesto en este trabajo
                     (flag[i]==3).
    """

```

Return:


```

return -- Devuelve varios arrays que contienen el logaritmo en base 10 de
        los factores de Bayes posicional y fotométrico. La casificación
        está basada en el valor de los factores de Bayes y si se cumplen
        o no los criterios de Negrello y Gonzalez-Nuevo.

"""
# Se van a modificar estos arrays. Para no sobrescribir sobre los
# originales se hace una copia. Sin estas lineas de código, el fits se
# genera a partir de bayes_z modificado (bz).
bz = copy.copy(bz_original)
s250 = copy.copy(s250_original)
s350 = copy.copy(s350_original)

# Creo los arrays vacios.
[x_0, y_0, x_1, y_1, x_2, y_2] = [[], [], [], [], [], []]
[x_3, y_3, x_4, y_4, x_5, y_5] = [[], [], [], [], [], []]

# Si seleccion=ajuste representamos las emparejados en los que participa
# una observación de HATLAS con flag=3
if seleccion=="ajuste":
    (a,b,c,d,e)=(3,3,3,3,3)
else:
    (a,b,c,d,e)=(11,12,13,2,3)

for i in range(0,len(nombre_hatlas),1):
    if (flag[i]==a or flag[i]==b or flag[i]==c or flag[i]==d or \
        flag[i]==e) and zh[i]>=1 and zh[i]>zg[i]:

        # Para evitar log(0) posteriormente.
        if np.any(bp[i]<1e-100):
            bp[i]=1e-100
        if np.any(bz[i]<1e-100):
            bz[i]=1e-100

        #-----
        # Selecciona los objetos de cuyos nombres están en el fichero.
        #-----
        for o in range(0,len(nombres_confirmados),1):
            nombre_candidato=False
            if nombre_hatlas[i]==nombres_confirmados[o]:
                nombre_candidato=True
                break
        if nombre_candidato==True:
            x_0.append(bp[i])
            y_0.append(bz[i])
        else:
            #-----
            # Clasificación de los emparejamientos.
            #-----
            if np.any(s500[i] >= 0.1): # Negrello et al.
                x_1.append(bp[i])
                y_1.append(bz[i])

```

```

else:
    # Para evitar divisiones por 0 (menor que la precisión de
    # la medida del instrumento SPIRE)
    if np.any(s250[i] < 1e-6):
        s250[i]=1e-6
    if np.any(s350[i] < 1e-6):
        s350[i]=1e-6
    s350_s250 = s350[i]/s250[i]
    s500_s350 = s500[i]/s350[i]

    if np.any(s350[i] >= 0.085) and np.any(s250[i] >= 0.035) \
        and np.any(s350_s250 > 0.6) \
        and np.any(s500_s350 > 0.4):
        x_2.append(bp[i])    # Condición (Gonzalez-Nuevo et al)
        y_2.append(bz[i])

    elif np.any(bc[i]>100): # Bayes conjunto mayor que 100.
        x_5.append(bp[i])
        y_5.append(bz[i])
    elif np.any(bc[i]>1):   # Bayes conjunto mayor que 1.
        x_4.append(bp[i])
        y_4.append(bz[i])
    else:
        x_3.append(bp[i])   # Resto.
        y_3.append(bz[i])

return np.log10(x_0), np.log10(y_0), np.log10(x_1), np.log10(y_1), \
       np.log10(x_2), np.log10(y_2), np.log10(x_3), np.log10(y_3), \
       np.log10(x_4), np.log10(y_4), np.log10(x_5), np.log10(y_5)

def seleccion_candidatos(s250_original,s350_original,s500,flag_h,zh,zg,
                        bayes_posicional,bayes_conjunto):
    """
    Permite seleccionar aquellos emparejamientos que cumplen nuestro criterio
    para ser considerados candidatos a lente gravitatoria. Los emparejamientos
    que cumplen nuestro criterio se agrupan en dos grupos dependiendo de la
    calidad de la medida del redshift. Por último se cuenta el número de
    emparejamientos que además contienen observaciones en HATLAS que cumplen
    los criterios de Negrello y González-Nuevo para ser candidatos a SLGs.

    Parámetros:
    s250_original ----- Medidas del flujo en 250 microm.
    s350_original ----- Medidas del flujo en 350 microm.
    s500 ----- Medidas del flujo en 500 microm.
    flag_h ----- Etiqueta de calidad sobre la medida del redshift de la
                  observación perteneciente a HATLAS.
    zh ----- Redshift perteneceinte a HATLAS
    zg ----- Redshift perteneceinte a GAMA
    bayes_posicional -- Factor de Bayes posicional
    bayes_conjunto ---- Factor de Bayes conjunto

```

Return:

id_halo_h_espec ----- Identificadores de los emparejamientos que cumplen nuestros criterios para ser candidatos y estan formados por observaciones que tienen medidas espectroscópicas del redhsift.

id_halo_h_fot ----- Identificadores de los emparejamientos que cumplen nuestros criterios para ser candidatos pero la observación de HATLAS tiene un redshift fotométrico.

id_halo_negrello_gonzalez -- Identificadores de los emparejamientos que cumplen nuestros criterios y la observación de HATLAS cumple además los criterios de Negrello y González-Nuevo.

"""

```
s250 = copy.copy(s250_original)
```

```
s350 = copy.copy(s350_original)
```

```
id_halo_h_espec=[]
```

```
id_halo_h_fot=[]
```

```
id_halo_negrello_gonzalez=[]
```

```
for i in range(0,len(s250_original),1):
```

```
    if zh[i]>=1 and zh[i]>zg[i]:
```

```
        negrello=False
```

```
        gonzalez=False
```

```
        bayes=False
```

```
        # Para evitar divisiones por 0 (menor que la precisión de la  
        # medida del instrumento SPIRE)
```

```
        if np.any(s250[i] < 1e-6):
```

```
            s250[i]=1e-6
```

```
        if np.any(s350[i] < 1e-6):
```

```
            s350[i]=1e-6
```

```
        s350_s250 = s350[i]/s250[i]
```

```
        s500_s350 = s500[i]/s350[i]
```

```
        if np.any(s500[i] >= 0.1):
```

```
            negrello=True
```

```
        if np.any(s350[i] >= 0.085) and np.any(s250[i] >= 0.035) \
```

```
            and np.any(s350_s250 > 0.6) and np.any(s500_s350 > 0.4):
```

```
            gonzalez=True
```

```
        if np.any(bayes_conjunto[i]<1) and np.any(bayes_posicional[i]>1):
```

```
            bayes=True
```

```
        if bayes==True:
```

```
            if flag_h[i]==3:
```

```
                id_halo_h_fot.append(i)
```

```
            else:
```

```
                id_halo_h_espec.append(i)
```

```

        if bayes==True and (negrello==True or gonzalez==True):
            id_halo_negrello_gonzalez.append(i)

    return id_halo_h_espec,id_halo_h_fot,id_halo_negrello_gonzalez

import math
import random

def uniform_spherical_distribution(RA_a,RA_b,DEC_a,DEC_b,n_vectors):
    """
    Genera una distribución esférica uniforme de puntos aleatorios comprendida
    entre los meridianos "RA_a" y "RA_b" y los paralelos "DEC_a" y "DEC_b". La
    variable "n_vectors" indica a la función en número de vectores que se
    debe generar.

    Parámetros:

    RA_a y RA_b ----- Meridianos límite.
    DEC_a y DEC_b ----- Paralelos límite.
    n_vectors -- Número de vectores que queremos generar.

    Return:
    RA_random --- Array con la ascensión recta de los vectores generados.
    DEC_random -- Array con la declinación de los vectores generados.

    """
    def random_3D_unit_vector(RA_a,RA_b,DEC_a,DEC_b):
        """
        Genera un vector unitario en un área definida entre "RA_a" y "RA_b" y
        "DEC_a" y "DEC_b" de forma aleatoria
        """
        RA = random.uniform(RA_a,RA_b) #0,360

        theta_a = 90-DEC_a # -90,90
        theta_a = math.radians(theta_a)
        cos_theta_a = math.cos(theta_a)

        theta_b = 90-DEC_b
        theta_b = math.radians(theta_b)
        cos_theta_b=math.cos(theta_b)

        cos_theta = random.uniform(cos_theta_a,cos_theta_b)

        theta = math.acos(cos_theta)
        theta = math.degrees(theta)
        DEC = 90-theta

```

```

        return RA, DEC

RA_random=[]
DEC_random=[]
for i in range(0,n_vectors,1):
    [RA,DEC]=random_3D_unit_vector(RA_a,RA_b,DEC_a,DEC_b)
    RA_random.append(RA)
    DEC_random.append(DEC)

return RA_random,DEC_random

```

C.10 Módulo main_smm.py

```

# -*- coding: utf-8 -*-

"""
Representación gráfica de la SED de la galaxia SMM expresado en  $F_{\lambda}$  y
 $F_{\epsilon}$  .

Python 3.6.0      matplotlib 2.0.0      astropy 1.3

@author: Javier Gutiérrez Solórzano
"""

#=====
#   Nombres de los ficheros de entrada/salida y su localización.
#=====
#-----
#   Directorio en el que se encuentran los módulos.
#-----
import sys
sys.path.append("../packages/")
#-----
#   Directorios de los ficheros.
#-----
inroute = './inroute/'
outroute = './outroute/'
#-----
#   Nombre del fichero de entrada
#-----
infile_template = 'SMM_template_norm.sed'

#=====
#   Desplazamiento al rojo de los objetos H-Atlas
#=====
import get
#-----
#   Leemos el fichero de la SED modelo. El fichero contiene dos columnas; la
# primera representa las longitudes de onda en Ångström y la segunda valores
# valores de la densidad espectral de flujo ( $F_{\lambda}$ ), en unidades de:

```

```

# (erg)* (cm)**(-2)* (s)**(-1)* (Å)**(-1) aunque normalizado en 5570 Ångström.
#-----

[x_sed, y_sed] = get.sed_func(inroute + infile_template)

#=====
#   Grafica F_(lambda)
#=====

import matplotlib.pyplot as plt

figura=plt.figure()
plt.loglog(x_sed,y_sed, color="blue", linewidth=1, linestyle="--",
           label="SMM J2")
plt.grid(False)
plt.title('SMM J2135-0102', fontsize = 16, color='blue')
plt.errorbar(x_sed[42],y_sed[42], fmt='+', label="data", linewidth=1, xerr=0,
             yerr=0, ecolor='yellow')
plt.xlabel('$ \lambda; [\mathrm{\AA}]$', fontsize = 16, color='blue')
plt.ylabel('$ \mathrm{F}_{\lambda}; [ @\,5570\, , \mathrm{\AA} ]$', fontsize = 16,
           color='blue')
figura.savefig(outroute + 'grafica_SMM_F_lambda', dpi=150, transparent=True)
plt.show()

#=====
#   Grafica F()
#=====

#   Soporte de unidades físicas
from astropy import units as u

#   Soporte de magnitudes físicas. Solo se utiliza la velocidad de la luz.
from astropy.constants import c

#   Unidades a las longitudes de onda
x_sed=x_sed*u.AA # En Ångström
#   Unidades de densidad espectral de flujo (F_(lambda))
y_sed=y_sed*((u.erg)* (u.cm)**(-2)* (u.s)**(-1)* (u.AA)**(-1))
#   Cambio de F_(lambda) a F(ípsilon) multiplicando por (lambda**2)/c
y_sed=y_sed*(x_sed**2)*(c**-1)
#   Cambio de unidades
x_sed=x_sed.to(u.micron)
y_sed=y_sed.to(u.mJy)

figura=plt.figure()
plt.loglog(x_sed,y_sed, color="blue", linewidth=1, linestyle="--",
           label="SMM J2")
plt.grid(False)
plt.title('SMM J2135-0102', fontsize = 16, color='blue')
plt.xlabel('$ \lambda; [\mu \mathrm{m}]$', fontsize = 16, color='blue')
plt.ylabel('$ \mathrm{F}_{\epsilon} ]$', fontsize = 16, color='blue')
figura.savefig(outroute + 'grafica_SMM_F_ípsilon', dpi=150, transparent=True)
plt.show()

```

C.11 Módulo main_main_radio_Einstein_D_L.py

```
# -*- coding: utf-8 -*-
```

```
"""
```

```
Gráfica de radio de Einstein frente a la distancia observador-lente.
```

```
Python 3.6.0      matplotlib 2.0.0      astropy 1.3
```

```
@author: Javier Gutiérrez Solórzano
```

```
"""
```

```
#=====
```

```
# Nombres de los ficheros de entrada/salida y su localización.
```

```
#=====
```

```
#-----
```

```
# Directorio en el que se encuentran los módulos.
```

```
#-----
```

```
import sys
```

```
sys.path.append("../packages/")
```

```
#-----
```

```
# Directorios de los ficheros.
```

```
#-----
```

```
outroute = './outroute/'
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
from astropy.cosmology import FlatLambdaCDM
```

```
cosmo = FlatLambdaCDM(H0=70, Om0=0.3)
```

```
print(cosmo)
```

```
from astropy import units as u
```

```
# Genera 100 valores igualmente espaciados entre 1 y 3.5
```

```
x_z= np.linspace(1e-5, 0.6, 100, endpoint=True)
```

```
x_pc=cosmo.luminosity_distance(np.array(x_z))
```

```
x_pc=x_pc.to(u.pc)
```

```
x_pc=x_pc.value
```

```
r_e_9=0.1*np.sqrt((1e9/np.array(x_pc)))#*u.arcsec
```

```
r_e_12=0.1*np.sqrt((1e12/np.array(x_pc)))#*u.arcsec
```

```
x_z_eje= np.linspace(0.1, 0.6, 6, endpoint=True)
```

```
x_pc_eje=np.round(cosmo.luminosity_distance(np.array(x_z_eje)).value, decimals=0, out=None)
```

```

#-----
#   Figura radio de Einstein vs distancia observador-lente.
#-----
figura = plt.figure(num = None, figsize = (9, 4), dpi = 80, facecolor = 'w',
                    edgecolor = 'k')
#   Cambiamos los límites de los ejes.
plt.axis([0, 0.6, 4e-2, 50])

#   Graficas para cada masa
plt.plot(x_z,r_e_12, linewidth=0.7, color="red", linestyle='--')
plt.plot(x_z,r_e_9, linewidth=0.7, color="red")

#   Líneas horizontales
plt.axhline(y=7.63, linewidth=0.5, color='k', linestyle='-.')
plt.axhline(y=0.297, linewidth=0.5, color='k', linestyle=':')

#   Escala de cada eje
plt.yscale('log', lenthreshx=0.1)
#plt.xscale('log', lenthreshx=0.1)

#   Sin rejilla de fondo.
plt.grid(False)

ax = plt.gca()
#   Etiqueta eje y.
ax.set_ylabel(r" $\theta_{\rm E}$  [\mathrm{arcsec}]$", fontsize=16, color='blue')

#   Etiqueta eje x.
plt.xlabel(r" $z_{\rm L}$ ", fontsize = 16, color='blue')

ax2 = ax.twinx()
ax2.set_xlabel(r" $D_{\rm L}$  [\mathrm{Mpc}]$", fontsize=16, color='blue')
ax2.set_xlim(0, 0.6)
ax2.set_xticks([0.1, 0.2, 0.3, 0.4, 0.5, 0.6])
ax2.set_xticklabels([460,980,1553,2172,2833,3530])

plt.show()
#   Guarda gráfica.
figura.savefig(outroute + "radio_einstein", dpi=150, transparent=True)

```

C.12 Módulo main_ajuste_error_espec.py

```
# -*- coding: utf-8 -*-
```

```
"""
```

Se realiza un ajuste lineal para obtener los errores asociados a las medidas espectroscópicas del redshift para los proyectos HALTAS y GAMA.

Python 3.6.0 scipy 0.18 matplotlib 2.0.0 numpy 1.11.3

@author: Javier Gutiérrez Solórzano

"""

#=====

Nombres de los ficheros de entrada/salida y su localización.

#=====

#-----

Directorio en el que se encuentran los módulos.

#-----

import sys

sys.path.append("../packages/")

#-----

Directorios de los ficheros.

#-----

outroute = './outroute/'

Redshift espectrocópico

z=[0.312005,0.385261,0.133342,0.125497,0.126204,0.130568,0.123867,0.103692,
0.223991,0.385721,0.374708,0.081022,0.394029,0.134499,0.476045,0.291690,
0.305359,0.038159,0.326157,0.293476]

Errores asociados a las medidas espectroscópicas (desviación estándar)

ez=[0.000102,0.000103,0.000092,0.000079,0.000100,0.000133,0.000136,0.000048,
0.000089,0.000117,0.000120,0.000130,0.000175,0.000119,0.000180,0.000098,
0.000119,0.000065,0.000351,0.000351]

import matplotlib.pyplot as plt

import numpy as np

from scipy.optimize import curve_fit

def func(x, A):

"""

Función que relaciona el redshift espectrocópico con su error.

"""

return A*(1+x)

pcov - Matriz covarianza

popt - Vector con valores de ajuste

popt, pcov = curve_fit(func, z, ez)

def recta_sup(x,opt_par,cov_matrix,nsigmas): # *función que calcula el*
extremo superior de la banda, a NSIGMAS sigmas

ymean = opt_par[0]*(x+1)

sa = cov_matrix[0]

y = ymean+nsigmas*np.sqrt(sa)

return y

def recta_inf(x,opt_par,cov_matrix,nsigmas): # *función que calcula el*
extremo inferior de la banda, a NSIGMAS sigmas

ymean = opt_par[0]*(x+1)

sa = cov_matrix[0]

y = ymean-nsigmas*np.sqrt(sa)

```

return y

#-----
# Grafica z_articulo vs z_programa
#-----
figura = plt.figure(num = None, figsize = (9, 6), dpi = 80, facecolor = 'w',
                    edgecolor = 'k')
# Cambiamos los limites de los ejes.
plt.axis([0, 0.6, 0, 0.0004])
# Genera 100 valores igualmente espaciados entre 1 y 3.5
x = np.linspace(0, 0.6, 10, endpoint=True)
# Recta del ajuste
plt.plot(x, popt[0]*(x+1), 'r', linewidth=0.5)
# Sombreado amarillo para la banda 1 sigma
plt.fill_between(x, recta_inf(x, popt, pcov, 2), recta_sup(x, popt, pcov, 2),
                facecolor='yellow', alpha=0.8)
# Sombreado amarillo para la banda 2 sigma
plt.fill_between(x, recta_inf(x, popt, pcov, 1), recta_sup(x, popt, pcov, 1),
                facecolor='orange', alpha=0.8)
# Dibuja los puntos
plt.errorbar(z, ez, fmt='+', label="data", linewidth=0.15, ecolor='black')
# Sin rejilla de fondo.
plt.grid(False)
# Etiqueta eje x.
plt.xlabel('$z^{\mathrm{spec}}$', fontsize = 16, color='blue')
# Etiqueta eje y.
plt.ylabel('$\sigma^z$', fontsize = 16, color='blue')
# Muestra la figura.
plt.show()
# Guarda una imagen de la gráfica.
figura.savefig(outroute + "ajuste_errores_z", dpi=150, transparent=True)

#-----
# Muestra datos del ajuste por pantalla
#-----
print('Errores z_spec vs z_spec:\n',
      '--- Best fit    y = ({0}'.format(popt[0]), chr(177) ,
      '{0})'.format(np.sqrt(pcov[0,0])), chr(215), '(1+x)')

```

Bibliografía

- N. Bourne, L. Dunne, S. Maddox, S. Dye, C. Furlanetto, C. Hoyos, D. Smith, S. Eales, M. Smith, E. Valiante, M. Alpaslan, E. Andrae, I. Baldry, M. Cluver, A. Cooray, S. Driver, J. Dunlop, M. Grootes, R. Ivison, T. Jarrett, J. Liske, B. Madore, C. Popescu, A. Robotham, K. Rowlands, M. Seibert, M. Thompson, R. Tuffs, S. Viaene, and A. Wright. The herschel-atlas data release 1 paper ii: Multi-wavelength counterparts to submillimetre sources. *Monthly Notices of the Royal Astronomical Society*, July 2016. doi: 10.1093/mnras/stw1654. URL <http://researchonline.ljmu.ac.uk/3894/>.
- T. Budavári and A. S. Szalay. Probabilistic cross-identification of astronomical sources. *The Astrophysical Journal*, 679:301-309, May 2008. doi: 10.1086/587156. URL <http://adsabs.harvard.edu/abs/2008ApJ...679..301B>.
- T. Budavári. Probabilistic cross-identification of cosmic events. *The Astrophysical Journal*, 736(2):155, 2011. URL <http://stacks.iop.org/0004-637X/736/i=2/a=155>.
- C. M. Casey. Far-infrared spectral energy distribution fitting for galaxies near and far. *Monthly Notices of the Royal Astronomical Society*, 425(4):3094–3103, 2012. doi: 10.1111/j.1365-2966.2012.21455.x. URL [+http://dx.doi.org/10.1111/j.1365-2966.2012.21455.x](http://dx.doi.org/10.1111/j.1365-2966.2012.21455.x).
- J. Cepa. *Cosmología física*. Ediciones Akal, S. A., 28760 Tres Cantos, Madrid, 2007.
- A. A. Collister and O. Lahav. Ann z : Estimating photometric redshifts using artificial neural networks. *Publications of the Astronomical Society of the Pacific*, 116(818):345, 2004. URL <http://stacks.iop.org/1538-3873/116/i=818/a=345>.
- S. Driver and t. GAMA team. Galaxy and mass assembly (gama). *International Astronomical Union*, 4, 08 2008. URL https://www.researchgate.net/publication/1739987_Galaxy_And_Mass_Assembly_GAMA.
- S. P. Driver, P. Norberg, I. K. Baldry, S. P. Bamford, A. M. Hopkins, J. Liske, J. Loveday, J. A. Peacock, D. T. Hill, L. S. Kelvin, A. S. G. Robotham, N. J. G. Cross, H. R. Parkinson, M. Prescott, C. J. Conselice, L. Dunne, S. Brough, H. Jones, R. G. Sharp, E. van Kampen, S. Oliver, I. G. Roseboom, J. Bland-Hawthorn, S. M. Croom, S. Ellis, E. Cameron, S. Cole, C. S. Frenk, W. J. Couch, A. W. Graham, R. Proctor, R. De Propriis, I. F. Doyle, E. M. Edmondson, R. C. Nichol, D. Thomas, S. A. Eales, M. J. Jarvis, K. Kuijken, O. Lahav, B. F. Madore, M. Seibert, M. J. Meyer, L. Staveley-Smith, S. Phillipps, C. C. Popescu, A. E. Sansom, W. J. Sutherland, R. J. Tuffs, and S. J. Warren. GAMA: towards a physical understanding of galaxy formation. *Astronomy and Geophysics*, 50(5):5.12–5.19, Oct. 2009. doi: 10.1111/j.1468-4004.2009.50512.x. URL <http://adsabs.harvard.edu/abs/2009A%26G...50e..12D>.
- S. P. Driver, D. T. Hill, L. S. Kelvin, A. S. G. Robotham, J. Liske, P. Norberg, I. K. Baldry, S. P. Bamford, A. M. Hopkins, J. Loveday, J. A. Peacock, E. Andrae, J. Bland-Hawthorn, S. Brough, M. J. I. Brown, E. Cameron, J. H. Y. Ching, M. Colless, C. J. Conselice, S. M. Croom, N. J. G. Cross, R. de Propriis, S. Dye, M. J. Drinkwater, S. Ellis, A. W. Graham, M. W. Grootes, M. Gunawardhana, D. H. Jones, E. van Kampen, C. Maraston, R. C. Nichol, H. R. Parkinson, S. Phillipps, K. Pimbblet, C. C. Popescu, M. Prescott, I. G. Roseboom, E. M. Sadler, A. E. Sansom, R. G. Sharp, D. J. B. Smith, E. Taylor, D. Thomas, R. J. Tuffs, D. Wijesinghe, L. Dunne, C. S. Frenk, M. J. Jarvis, B. F. Madore, M. J. Meyer, M. Seibert, L. Staveley-Smith, W. J. Sutherland, and S. J. Warren. Galaxy and Mass Assembly (GAMA): survey

- diagnostics and core data release. *Monthly Notices of the Royal Astronomical Society*, 413:971–995, May 2011. doi: 10.1111/j.1365-2966.2010.18188.x. URL <http://adsabs.harvard.edu/abs/2011MNRAS.413..971D>.
- GAMA. Proyecto GAMA, 2018. URL <http://www.gama-survey.org>. Accedido: Enero, 2018.
- A. M. Garcia. *Estudio e implementación de una herramienta probabilística para la Cross-identificación de objetos extragalácticos*. PhD thesis, Universidad de Cantabria (España), 2015. URL <https://repositorio.unican.es/xmlui/handle/10902/7805?show=full>.
- J. González-Nuevo, A. Lapi, S. Fleuren, S. Bressan, L. Danese, G. D. Zotti, M. Negrello, Z.-Y. Cai, L. Fan, W. Sutherland, M. Baes, A. J. Baker, D. L. Clements, A. Cooray, H. Dannerbauer, L. Dunne, S. Dye, S. Eales, D. T. Frayer, A. I. Harris, R. Ivison, M. J. Jarvis, M. J. Michałowski, M. López-Caniego, G. Rodighiero, K. Rowlands, S. Serjeant, D. Scott, P. van der Werf, R. Auld, S. Buttiglione, A. Cava, A. Dariush, J. Fritz, R. Hopwood, E. Ibar, S. Maddox, E. Pascale, M. Pohlen, E. Rigby, D. Smith, and P. Temi. Herschel-atlas: Toward a sample of 1000 strongly lensed galaxies. *The Astrophysical Journal*, 749(1):65, 2012. URL <http://stacks.iop.org/0004-637X/749/i=1/a=65>.
- Herschel*. Instrumentos de *Herschel*, 2018. URL <http://herschel.cf.ac.uk/mission/instruments>. Accedido: Enero, 2018.
- H-ATLAS. Proyecto H-ATLAS, 2018. URL www.h-atlas.org. Accedido: Enero, 2018.
- HIFI. Proyecto *Herschel*-HIFI, 2018. URL <https://www.sron.nl/missions-herschel-hifi>. Accedido: Enero, 2018.
- R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90 (430):773–795, 1995. ISSN 01621459. URL <http://www.jstor.org/stable/2291091>.
- A. Lapi, J. González-Nuevo, L. Fan, A. Bressan, G. D. Zotti, L. Danese, M. Negrello, L. Dunne, S. Eales, S. Maddox, R. Auld, M. Baes, D. G. Bonfield, S. Buttiglione, A. Cava, D. L. Clements, A. Cooray, A. Dariush, S. Dye, J. Fritz, D. Herranz, R. Hopwood, E. Ibar, R. Ivison, M. J. Jarvis, S. Kaviraj, M. López-Caniego, M. Massardi, M. J. Michałowski, E. Pascale, M. Pohlen, E. Rigby, G. Rodighiero, S. Serjeant, D. J. B. Smith, P. Temi, J. Wardlow, and P. van der Werf. Herschel-atlas galaxy counts and high-redshift luminosity functions: The formation of massive early-type galaxies. *The Astrophysical Journal*, 742(1):24, 2011. URL <http://stacks.iop.org/0004-637X/742/i=1/a=24>.
- P. Murdin. *The Encyclopedia of Astronomy and Astrophysics*. (Web version) NY and London: Nature Publishing Group and Institute of Physics Publishing Co., Dirac House, Temple Back, Bristol, BS1 6BE, UK, 2001.
- M. Negrello, R. Hopwood, G. De Zotti, A. Cooray, A. Verma, J. Bock, D. Frayer, M. Gurwell, A. Omont, R. Neri, H. Dannerbauer, L. Leeuw, E. Barton, J. Cooke, S. Kim, E. Da Cunha, G. Rodighiero, P. Cox, D. Bonfield, M. Jarvis, S. Serjeant, R. Ivison, S. Dye, I. Aretxaga, D. Hughes, E. Ibar, F. Bertoldi, I. Valtchanov, S. Eales, L. Dunne, S. Driver, R. Auld, S. Buttiglione, A. Cava, C. Grady, D. Clements, A. Dariush, J. Fritz, D. Hill, J. Hornbeck, L. Kelvin, G. Lagache, M. Lopez-Caniego, J. Gonzalez-Nuevo, S. Maddox, E. Pascale, M. Pohlen, E. Rigby, A. Robotham, C. Simpson, D. Smith, P. Temi, M. Thompson, B. Woodgate, D. York, J. Aguirre, A. Beelen, A. Blain, A. Baker, M. Birkinshaw, R. Blundell, C. Bradford, D. Burgarella, L. Danese, J. Dunlop, S. Fleuren, J. Glenn, A. Harris, J. Kamenetzky, R. Lupu, R. Maddalena, B. Madore, P. Maloney, H. Matsuhara, M. Michaowski, E. Murphy,

- B. Naylor, H. Nguyen, C. Popescu, S. Rawlings, D. Rigopoulou, D. Scott, K. Scott, M. Seibert, I. Smail, R. Tuffs, J. Vieira, P. Van Der Werf, and J. Zmuidzinas. The detection of a population of submillimeter-bright, strongly lensed galaxies. *Science*, 330(6005):800–804, 11 2010. ISSN 0036-8075. doi: 10.1126/science.1193420. URL https://www.astro.umd.edu/rareas/lma/publications/2010/science_2010_Negrello_800_4.pdf.
- PACS. Proyecto *Herschel*-PACS, 2018. URL <http://www.mpe.mpg.de/ir/Pacs>. Accedido: Enero, 2018.
- E. E. Rigby, S. J. Maddox, L. Dunne, M. Negrello, D. J. B. Smith, J. González-Nuevo, D. Herranz, M. López-Caniego, R. Auld, S. Buttiglione, M. Baes, A. Cava, A. Cooray, D. L. Clements, A. Dariush, G. De Zotti, S. Dye, S. Eales, D. Frayer, J. Fritz, R. Hopwood, E. Ibar, R. J. Ivison, M. Jarvis, P. Panuzzo, E. Pascale, M. Pohlen, G. Rodighiero, S. Serjeant, P. Temi, and M. A. Thompson. *Herschel-atlas: first data release of the science demonstration phase source catalogues*. *Monthly Notices of the Royal Astronomical Society*, 415(3):2336–2348, 2011. doi: 10.1111/j.1365-2966.2011.18864.x. URL [+http://dx.doi.org/10.1111/j.1365-2966.2011.18864.x](http://dx.doi.org/10.1111/j.1365-2966.2011.18864.x).
- R. J. Serrano. *Detección y caracterización de cúmulos de galaxias con alto corrimiento al rojo*. PhD thesis, Universidad de La Laguna (España), 2010. URL <https://documat.unirioja.es/servlet/tesis?codigo=21622>.
- SPIRE. Proyecto *Herschel*-SPIRE, 2018. URL <http://www.astro.cf.ac.uk/research/astro/instr/projects/?page=spire>. Accedido: Enero, 2018.
- A. M. Swinbank, I. Smail, S. Longmore, A. I. Harris, A. J. Baker, C. D. Breuck, J. Richard, A. Edge, R. J. Ivison, R. Blundell, K. Coppin, P. Cox, M. Gurwell, L. J. Hainline, M. Krips, A. Lundgren, R. Neri, B. Siana, G. Siringo, D. P. Stark, D. Wilner, and J. D. Younger. Intense star formation within resolved compact regions in a galaxy at $z=2.3$. *Nature*, 2010. URL <https://arxiv.org/pdf/1003.3674v1.pdf>.
- E. Valiante, M. W. L. Smith, S. Eales, S. J. Maddox, E. Ibar, R. Hopwood, L. Dunne, P. J. Cigan, S. Dye, E. Pascale, E. E. Rigby, N. Bourne, C. Furlanetto, and R. J. Ivison. The *Herschel*-ATLAS data release 1 - I. Maps, catalogues and number counts. *Monthly Notices of the Royal Astronomical Society*, 462:3146–3179, Nov. 2016. doi: 10.1093/mnras/stw1806. URL <http://adsabs.harvard.edu/abs/2016MNRAS.462.3146V>.
- S. Weinberg. *Gravitation And Cosmology: Principles And Applications Of The General Theory Of Relativity*. John Wiley & Sons, Inc., 1972. ISBN 0471925675. URL <https://archive.org/details/WeinbergS.GravitationAndCosmology.PrinciplesAndApplicationsOfTheGeneralTheoryOf>.