

Práctica de detección de fuentes con SExtractor

Cedric Prieels

Marzo 2017

Resumen

Esta práctica consiste en analizar una imagen astronómica de formato FITS, usando la herramienta SExtractor. Se estudia el número de objetos detectados en esta imagen, cambiando diferentes parámetros como por ejemplo, el umbral de detección y poniendo cortes en el número mínimo de píxeles conectados activados, con el objetivo final de poder seleccionar solamente los objetos de verdad en la imagen, quitando el máximo número posible de objetos que aparecen en la imagen pero que provienen solamente del ruido. También se usan diferentes tipos de filtros para ver el impacto que pueden tener a la hora de quitar ruido en la imagen.

1. Desarrollo del ejercicio

Primero, queremos abrir el fichero de tipo FITS que tenemos para poder estudiar después la foto. Para esto se usa la herramienta *Javascript FITS viewer* disponible en <http://www.clearskyinstitute.com/fits/>, que permite visualizar la imagen que vamos a estudiar en este ejercicio. El resultado así obtenido se enseña en la figura 1.

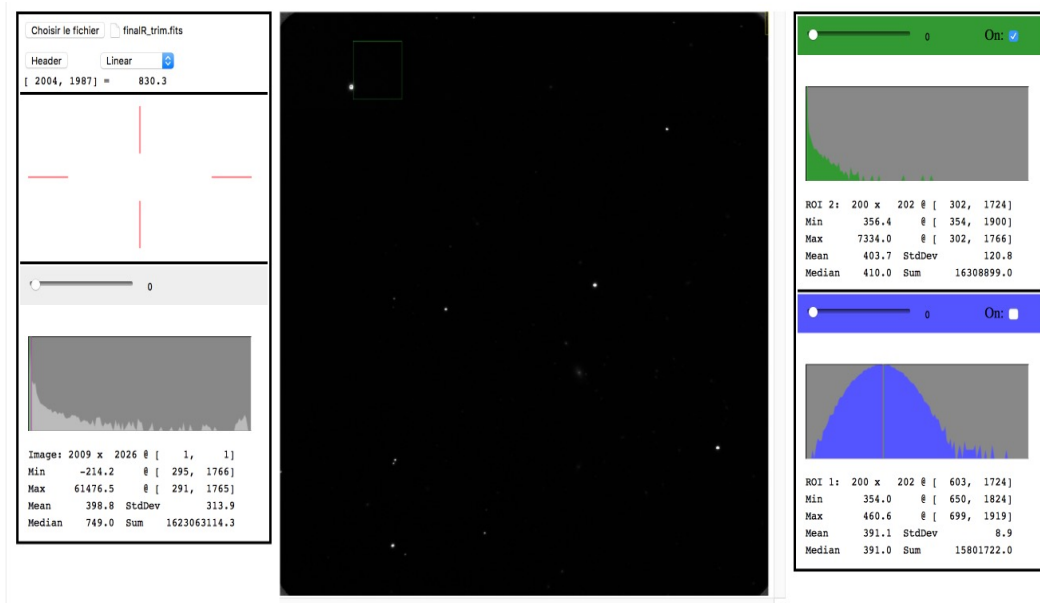


Figura 1: Imagen que vamos a estudiar en este ejercicio

Ya que hemos visualizado la imagen del ejercicio, podemos empezar a estudiarla con el programa SExtractor, para intentar analizarla sus características y el número de objetos físicos presentes.

2. Resultados obtenidos

Para obtener resultados, lo primero consiste en descargar e instalar el software SExtractor, y en encontrar por internet (en <https://github.com/Starlink/sextractor/tree/master/config>) ficheros de configuración y de parámetros por defecto.

2.1. Umbrales de detección

Una vez hecho, podemos empezar el estudio de la foto que tenemos, yendo probando diferentes umbrales de detección (2σ , 3σ ,...) para estudiar el número de objetos detectados en la imagen, dejando todos los otros parámetros por defecto. Los resultados se enseñan en la tabla siguiente (resultados obtenidos con la configuración por defecto del programa, modificando solamente cada vez el parámetro llamado DETECT_THRESH) :

Número de sigma	Objetos detectados	Objetos <i>sextractados</i>	Percentage
1.5	1394	992	71,1 %
2.0	842	696	82,7 %
3.0	503	463	92,0 %
4.0	383	360	94,0 %
5.0	308	297	96,4 %

Vemos claramente que cada vez que vamos subiendo el número de sigma, el número de objetos detectados va bajando pero que la diferencia entre el número de objetos que aparece en cada columna va bajando (y por lo tanto, el percentage en la última columna va subiendo). Esto significa que estamos seleccionando cada vez menos objetos, pero objetos mejores.

2.2. Píxeles conectados

Podemos ahora ir cambiando el parámetro DETECT_MINAREA para probar diferentes cantidades mínimas de píxeles conectados. Un poco lo mismo que en el paso precedente va a pasar : cuando pedimos que haya más píxeles activados, tendremos menos objetos seleccionados pero los que pasan este corte van a ser mejores (porque no esperemos que el ruido active diferentes píxeles cercanos). Los resultados están en la tabla siguiente (estos resultados se obtienen con la configuración por defecto de SExtractor, considerando un umbral de detección de 2 sigmas) :

Número de píxeles	Objetos detectados	Objetos <i>sextractados</i>	Percentage
1	1008	749	74,3 %
2	842	696	82,7 %
5	668	605	90,5 %
10	566	523	92,4 %
20	452	423	93,6 %

Como ya comentado y como se esperaba, lo mismo pasa que en la sección precedente : al aumentar el número de píxeles activados que pedimos, se seleccionan cada vez menos objetos, pero objetos que tienen una probabilidad más alta de no provenir del ruido del detector.

2.3. Tamaño de la reticula

Un otro parámetro que vamos a ir cambiando es el tamaño de la reticula en la que se calcula el fondo (parámetro llamado `BACK_SIZE`). Lo que esperemos es ver que poner un tamaño menor que los objetos más grandes de la imagen implica que el fondo esté muy sobreestimado, mientras que poner un tamaño demasiado grande resultará en una estimación de fondo demasiado pequeña. Hasta ahora, hemos considerado el valor por defecto de 64.

Tamaño	Objetos detectados	Objetos <i>sextractados</i>	Percentage
8	673	636	94,5 %
16	680	634	93,2 %
32	661	616	93,2 %
64	668	605	90,6 %
128	645	595	92,2 %

Vemos claramente que con los valores elegidos para este parámetro, el número de objetos obtenido en cada columna cambia bastante poco. Esto significa que todos los valores considerados están correctos, no parece que estemos considerando valores demasiado pequeños o grandes que podrían influir en las medidas y en el número de objetos detectados.

2.4. Parámetro de deblending

El último parámetro que vamos a ir cambiando es el *deblending*, que nos ayuda a determinar si en un conjunto de puntos se ha detectado más de una fuente (es por lo tanto importante si hay muchos objetos en la imagen, o si consideramos un umbral de detección muy pequeño y entonces no debería jugar un papel muy importante en este ejercicio). Este estudio se hace mediante dos parámetros : `DEBLEND_NTHRESH` (define el número de niveles que queremos entre el umbral definido y el número máximo, que toma en general valores menores que 32, si nos referimos al manual del usuario de `SExtractor`), y `DEBLEND_MINCONT` (corresponde al contraste mínimo, lo que significa que si lo ponemos a 0, incluso los picos locales más débiles van a aparecer como objetos separados).

Primero, vamos a ir cambiando el parámetro `DEBLEND_NTHRESH`, dejando `DEBLEND_MINCONT` a 0.005, su valor por defecto.

Deblending umbral	Objetos detectados	Objetos <i>sextractados</i>	Percentage
4	656	594	90,6 %
8	660	598	90,6 %
16	666	603	90,6 %
32	668	605	90,6 %

Como se puede ver y como se esperaba, este parámetro tiene un impacto pequeño. Después, vamos cambiando el segundo parámetro considerado, `DEBLEND_MINCONT`.

Deblending contraste	Objetos detectados	Objetos <i>sextractados</i>	Percentage
0.0	800	668	83,5 %
0.25	631	579	91,8 %
0.5	630	578	91,8 %
0.75	630	578	91,8 %
1.0	630	578	91,8 %

Como se esperaba en este caso también, si ponemos un valor diferente de 0 al parámetro `DEBLEND_MINCONT`, no observemos casi ningún cambio en la eficiencia de la detección.

2.5. Filtros

Poner cortes en las variables precedentes no es la única manera de seleccionar objetos de manera a mejorar el ratio entre señal y ruido : también se pueden usar filtros. Vamos a simular el cambio introducido por el uso de diferentes filtros (filtros tophat y gaussianos). Los filtros se introducen como ficheros .conv y se pasan como argumento al comando general *sex* de SExtractor. Diferentes filtros listos para el uso se pueden ver en la página <https://github.com/Starlink/sextractor/tree/master/config>.

Filtro Top-Hat

Este tipo de filtro puede ser aplicado para suprimir posibles surestimaciones locales debidas a estrellas muy brillantes (se utilizan en general para detectar objetos extendidos y poco brillantes). Podemos estudiar el impacto de este tipo de filtro sobre el número de detecciones, dejando todos los parámetros a sus valores por defecto.

Filtro	Objetos detectados	Objetos <i>sextractados</i>	Percentage
tophat_1.5_3x3	840	691	82,2 %
tophat_2.0_3x3	708	630	89,0 %
tophat_2.5_3x3	665	604	90,1 %
tophat_3.0_3x3	641	595	92,8 %
tophat_4.0_5x5	571	537	94,0 %
tophat_5.0_5x5	531	508	95,7 %

Vemos claramente que el cambiar el filtro utilizado, la eficiencia de selección también cambia bastante. El mejor filtro parece ser el último con esta imagen.

Filtro gaussiano

También podemos usar filtros gaussianos para ver el impacto del uso de diferentes filtros sobre el número de objetos detectados. Estos filtros se usan en general para detectar objetos débiles. Vamos a considerar filtros que tengas valores de FWHM en píxeles diferentes.

Filtro	Objetos detectados	Objetos <i>sextractados</i>	Percentage
gauss_1.5_3x3	697	629	90,2 %
gauss_2.0_3x3	665	602	90,5 %
gauss_2.0_5x5	618	575	93,0 %
gauss_3.0_5x5	562	530	95,7 %
gauss_3.0_7x7	539	514	95,4 %
gauss_4.0_7x7	488	462	94,7 %
gauss_5.0_9x9	418	404	96,7 %

En este caso también, vemos que usar filtros diferentes cambia mucho el número de objetos detectados, y puede mejorar bastante la eficiencia de selección.

3. Ejemplos de selección

Primero, podemos representar a la izquierda una imagen obtenida con todos los parámetros por defecto y a la derecha, la misma imagen pero cambiando unos parámetros (el umbral de detección, el número de píxeles conectados y el tamaño de retícula). Estas imágenes son el resultado del uso de la opción CHECKIMAGE_TYPE igual a SEGMENTATION.

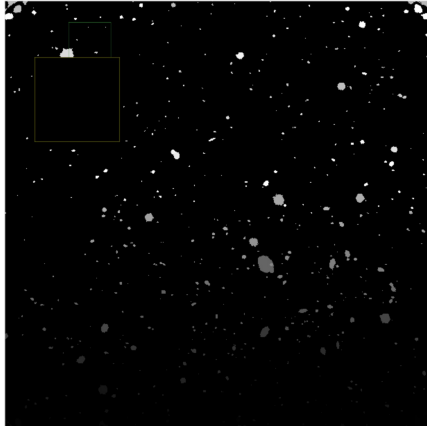


Imagen obtenida con el umbral de detección puesto a 2 sigmas, el número de píxeles conectados a 5 y el valor de tamaño de la retícula de 64.



Imagen obtenida con el umbral de detección puesto a 3 sigmas, el número de píxeles conectados a 20 y el valor de tamaño de la retícula de 8.

También podemos comparar el resultado obtenido con y sin filtro de tipo gaussiano.

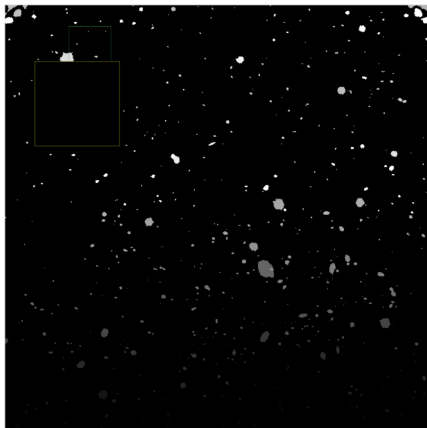


Imagen obtenida sin aplicar ningún filtro.

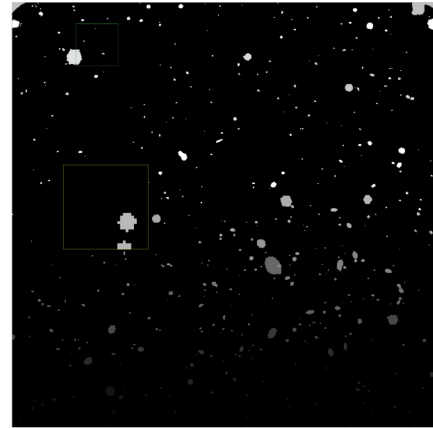


Imagen obtenida con el filtro gaussiano gauss_3.0_7x7.conv

4. Conclusiones

En conclusión, esta práctica nos permitió usar por primera vez el programa SExtractor que nos permite analizar imágenes astronómicas y seleccionar en estas imágenes candidatos a ser objetos de verdad, por el uso de diferentes parámetros que permiten diferenciar el ruido de una observación real. Estudiamos el impacto de diferentes parámetros sobre el valor de objetos detectados por SExtractor, como el valor del umbral en sigmas, el número de píxeles conectados activados y el tamaño de la retícula, y intentamos simular filtros de diferentes tipos para ver el impacto que tienen sobre la detección. Lo que hemos observado en todos los casos es que el uso de estos filtros y parámetros permite cambiar el número de objetos detectados y cambiar la diferencia que existe entre el número de objetos detectados y el número de objetos *sextractados*, lo que significa que podemos ir seleccionando cada vez menos objetos de interés, aumentando la probabilidad de que sean objetos de verdad, y no solamente ruido.