PySex
Manual de usuario

Christopher Añorve

1 Introducción

1.1 ¿Que es pysex?

Pysex es una rutina escrita python que permite obtener un catálogo con la fotometría de los objetos en una imagen.

Para obtener una mejor reliabilidad y tamaño de todos los objetos en una imagen, pysex ejecuta **sextractor** Bertin and Arnouts [1996] con dos configuraciones diferentes y combina estos dos catálogos en uno final.

En el caso de la rutina pysex3 y pysexbcg **sextractor** es ejecutado 3 veces.

1.2 ¿Por que ejecutar dos (o más) veces sextractor?

El programa **sextractor** ya realiza esta rutina de estimar los tamaños y fotometría de los objetos que se encuentran en una imagen. Sin embargo, el problema con este es que no hay una sola configuración en la cual se pueda obtener toda la detección de galaxias para imágenes que contengan grandes cantidades de objetos.

Por ejemplo, para una determinada configuración, **sextractor** puede detectar satisfactoriamente galaxias pequeñas que se encuentren agrupadas, sin embargo si hay una galaxia grande está puede ser dividida incorrectamente en varios objetos pequeños Häussler et al. [2007].

Por el otro lado, para una distinta configuración, la buena detección de galaxias grandes puede presentar dificultades en la detección de galaxias pequeñas. Para ilustrar mejor esto, véase la figura 1.

Además de esto existe otro problema adicional: en las regiones donde hay más densidad objetos, **sextractor** detecta las fuentes débiles como objetos de tamaño grande, dando la apariencia de que son objetos con bajo brillo superficial (ver figura 2). Este es un problema común con **sextractor** en las imágenes con cúmulos de galaxias.

Todo esto da como resultado un catálogo con objetos que son divididos en pequeñas partes, objetos débiles con tamaños sobredimensionados y objetos que no son detectados ya sea porque son débiles o están cerca de objetos con brillos grandes.

Una mejor solución es ejecutar **sextractor** con dos configuraciones optimizadas: una para objetos grandes y otra para objetos pequeños. Una vez que se obtengan estos, el catálogo final sería simplemente el resultado de la combinación de estos dos catálogos. Esto es lo que hace **pysex**.

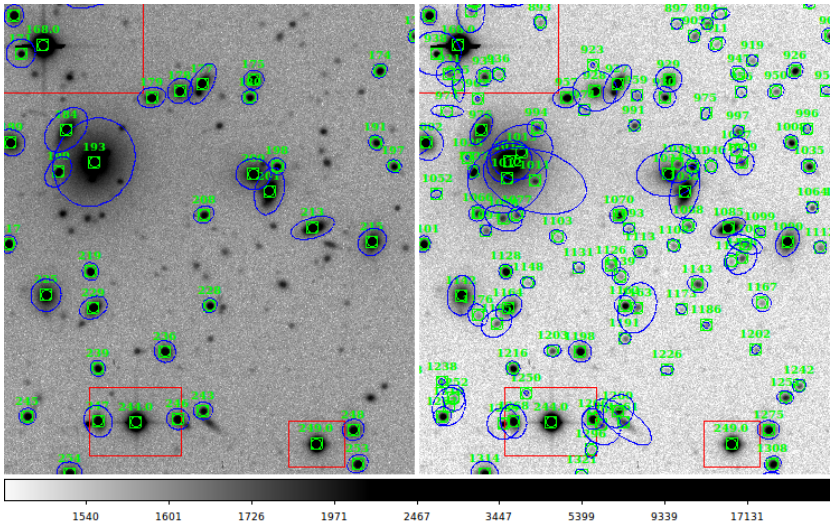

Figure 1: Resultados de ejecutar **sextractor** para detectar objetos grandes, pero no pequeños (imagen izquierda). Detección de objetos pequeños pero no lo hace bien para los objetos grandes y en ciertos casos llega a haber subdivisión (imagen derecha).

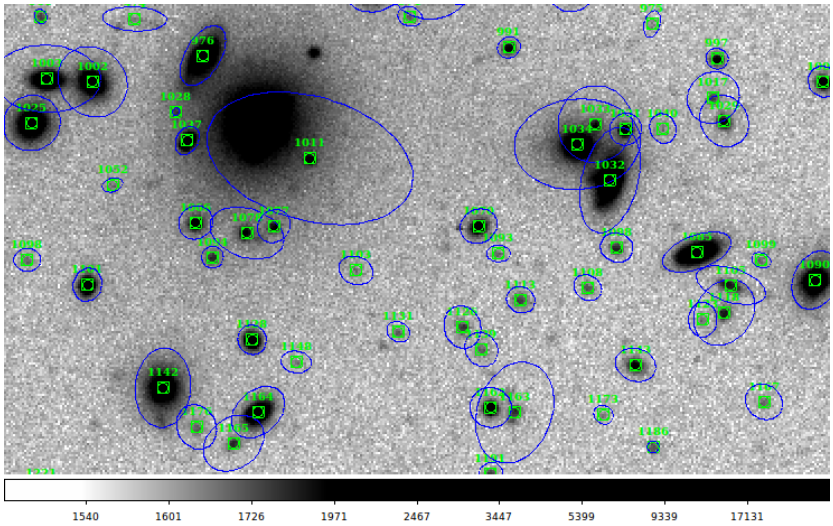

Figure 2: Resultados de ejecutar **sextractor**. En este caso sobredimensiona objetos pequeños, cuando se encuentra alrededor de otros, como si fueran objetos de bajo brillo superficial. Vease el objeto número 1011 junto a la galaxia grande.

2 Funcionamiento

Pysex une dos catálogos de **sextractor**. Llamaremos a estos catálogos **hot** y **cold**. La manera en que lo hace es tomar **todos** los objetos del catálogo **hot** y añadir aquellos objetos de **cold** pero que no están en **hot**.

Cuando **sextractor** se ejecuta para generar el catálogo **hot**, este se encargá de detectar todos los objetos grandes, brillantes y extendidos. Por el otro lado, en la configuración **cold**, **sextractor** está enfocado en detectar objetos débiles y cercanos entre sí.

Para decidir que objetos del catálogo **cold** serán incluidos en el catálogo final, **sextractor** incluye los objetos del catálogo **cold** que no contengan en su área los centros de cualquiera de los objetos que pertenezcan al catálogo hot.

Para calcular el área de cada objeto pysex calcula una elipse para cada objeto basado en los parámetros de **sextractor**. El eje mayor es calculado por medio de la siguiente ecuación:

$$Rkron = scale \times ai \times kr + offset \tag{1}$$

Donde *scale* es el factor de escala dado por el cual se agranda la elipse. *ai* es *A_IMAGE* del parámetro de **sextractor** (ver manual). *kr* es Kron radius. offset es el número de pixeles que se agrega al valor del eje mayor.

Detalles sobre que configurar los catalogos hot y cold son explícados en 6.

3 El programa

El paquete *pysex* contiene los siguientes archivos:

pysex.py Programa principal

config.txt Archivo de configuración de pysex

default.conv Archivo de **sextractor** para convolución

default.sex Archivo en el cual se basa pysex para configurar **sextractor**

default.nnw Archivo neural networks de **sextractor**

sex.param Archivo de salida de parámetros de **sextractor**

pysex3.py Pysex que ejecuta **sextractor** 3 veces

pysexbcg.py Pysex que ejecuta **sextractor** 3 veces para detectar BCG

makemask.py Crea máscaras a partir del catálogo de pysex

remove.py Programa que remueve mascaras particulares de la imagen creada por makemask.py

MaskBox.py Programa que añade o remueve mascaras a partir de una region de DS9

LICENSE Licencia del programa

4 Requerimientos

Para ejecutar pysex se necesita tener instalados los siguientes programas: **DS9**, **sextractor** y **python 3**. En el caso de python, se necesitan las siguientes librerías: *numpy*, *sys*, *os*, *subprocess*, *astropy*, *os*, *scipy*.

Para instalar DS9, y **sextractor** en *ubuntu* se puede ejecutar la siguiente instrucción:

```
> sudo apt install saods9
> sudo apt install \textbf{sextractor}
```

Para instalar python 3 se recomienda hacerlo a través del paquete anaconda.

5 Manual

Para ejecutar pysex se necesita ejecurar el siguiente comando (linux):

```
>  ./pysex.py [ConfigFile] [ImageFile]
```

ImageFile es la imagen en la cual pysex va a trabajar y ConfigFile es el archivo de parámetros que pysex necesita para ejecutar **sextractor**. Cada uno de los parámetros se explica en la siguiente sección.

5.1 Archivo de parámetros

A continuación se muestra los parámetros

FirstRun	Permite ejecutar **sextractor** para la primera configuración
SecondRun	Permite ejecutar **sextractor** para la segunda configuración
Scale	Factor de escala para el cual las elipses de cada objeto son agrandadas
SatDs9	Archivo de regiones para DS9 que indica los objetos saturados
SatScale	Factor de escala para el cual las regiones saturadas de cada objeto son agrandados
SatOffset	Cantidad de pixeles que se pueden alargar para las regiones para los objetos saturados
MakeMask	Bandera que indica si pysex va a realizar una imagen de máscaras (valor 1 para hacerlo)
OutCatalog	Nombre final de catálogo de salida
RegDs9	Archivo de regiones para DS9 que muestra los objetos de los catálogo final.

DEBLEND_NTHRESH1 Número de separadores para umbral. Parámetro para **sextractor** para ejecución hot. Ver DEBLEND_NTHRESH del manual de **sextractor**.

DEBLEND_MINCONT1 . Mínimo valor de contraste para separació.n Parámetro para **sextractor** para ejecución hot. Ver DEBLEND_MINCONT del manual de **sextractor**.

ANALYSIS_THRESH1 Valor sigma o umbral para realizar análisis. Parámetro para **sextractor** para ejecución hot. Ver ANALYSIS_THRESH del manual de **sextractor**

DETECT_THRESH1 . Valor sigma o umbral para detectar fuentes. Parámetro para **sextractor** para ejecución hot. Ver DETECT_THRESH del manual de **sextractor**

DETECT_MINAREA1 . Valor mínimo de área por encima del umbral para detectar fuentes. Parámetro para **sextractor** para ejecución hot. Ver DETECT_MINAREA del manual de **sextractor**

BACK_SIZE1 . Malla para el fondo. Parámetro para **sextractor** para ejecución hot. Ver BACK_SIZE del manual de **sextractor**

BACK_FILTERSIZE1 . Tamaño de filtro para el fondo. Parámetro para **sextractor** para ejecución hot. Ver BACK_FILTERSIZE del manual de **sextractor**

ANALYSIS_THRESH2 Parámetro para **sextractor** para ejecución cold. Ver ANALYSIS_THRESH del manual de **sextractor**

DETECT_THRESH2 Parámetro para **sextractor** para ejecución cold. Ver DETECT_THRESH del manual de **sextractor**

DETECT_MINAREA2 Parámetro para **sextractor** para ejecución cold. Ver DETECT_MINAREA del manual de **sextractor**

DEBLEND_NTHRESH2 Parámetro para **sextractor** para ejecución cold. Ver DEBLEND_NTHRESH del manual de **sextractor**

DEBLEND_MINCONT2 Parámetro para **sextractor** para ejecución cold. Ver DEBLEND_MINCONT del manual de **sextractor**

BACK_SIZE2 Parámetro para **sextractor** para ejecución cold. Ver BACK_SIZE del manual de **sextractor**

BACK_FILTERSIZE2 Parámetro para **sextractor** para ejecución cold. Ver BACK_FILTERSIZE del manual de **sextractor**

Un ejemplo de como se vería el archivo de configuración para pysex se muestra a continuación:

```
FirstRun  1   # Enable first run (1 = run)
SecondRun 1  # enable second run   (1 = run)
Scale  1  # factor scale which ellipses are enlarged
```

```
SatDs9 sat.reg
SatScale 3
SatOffset  1
MakeMask  0
OutCatalog  hotcold.cat
RegDs9   hotcold.reg


DEBLEND_NTHRESH1 64          # Number of debl|ending sub-thresholds
DEBLEND_MINCONT1 0.001         # Minimum contrast parameter for deblending
ANALYSIS_THRESH1 20          # <sigmas> or <threshold>,<ZP> in mag.arcsec-2
DETECT_THRESH1   20          # <sigmas> or <threshold>,<ZP> in mag.arcsec-2
DETECT_MINAREA1  10          # minimum number of pixels above threshold
BACK_SIZE1      100
BACK_FILTERSIZE1 11
ANALYSIS_THRESH2 1.5          # <sigmas> or <threshold>,<ZP> in mag.arcsec-2
DETECT_THRESH2   1.5          # <sigmas> or <threshold>,<ZP> in mag.arcsec-2
DETECT_MINAREA2  10      # minimum number of pixels above threshold
DEBLEND_NTHRESH2 16           # Number of deblending sub-thresholds
DEBLEND_MINCONT2 0.01         # Minimum contrast parapymeter for deblending
BACK_SIZE2       10
BACK_FILTERSIZE2 2
```

5.2 Archivos de salida

Pysex produce principalmente 4 archivos: 1) catálogo final de **sextractor**, 2) archivo de regiones DS9 de para fuentes saturadas, 3) archivo de regiones DS9 de las fuentes del catálogo final, y 4) una imagen máscara que contiene cada uno los objetos detectados. La descripción de cada archivo se muestra a continuación:

1) El catálogo final de **sextractor** indicado por el nombre dado en *OutCatalog* contiene los parámetros de cada una de los objetos detectados en la fuente. Las columnas que contendrá el catálogo final estará indicado por el archivo que está descrito bajo la variable PARAMETERS_NAME del archivo *default.sex* (en nuestro ejemplo *sex.param*).

2) El archivo nombrado por la variable *SatDs9* contiene las regiones de los objetos con píxeles saturados. Para ver estás regiones se necesita DS9 y ejecutarlo en la línea de comandos:

```
>  ds9 imagen -regions SatDs9
```

3) El archivo de regiones DS9 de los objetos finales contiene a todos los objetos que están en el catálogo final. El nombre de este archivo está dado por la variable *RegDs9*. Para ver estas regiones igualmente se necesita tener DS9 y ejecutarlo en la línea de comandos:

```
>  ds9 imagen -regions RegDs9
```

4) El archivo de salida "mask.fits" es una imagen fits que muestra cada uno de los objetos del catálogo final. Cada objeto es mostrado por una elipse. Cada una tiene una cantidad de flujo que corresponde exactamente al número del objeto del catálogo final (exactamente como lo hace **sextractor** en la segmentation check image). Así de esta manera es posible identificar el área de cada objeto.

6 Como obtener catálogos óptimos

Para detectar la mayor cantidad de objetos en una imagen, se necesita ajustar **pysex** para que ejecute **sextractor** con dos configuraciones diferentes. A continuación se muestran recomendaciones de configuracion de **pysex** para una buena detección de los objetos.

La idea de la primera ejecución con **sextractor** (catálogo **hot**) es detectar los objetos grandes y brillantes. Para esto, se recomienda poner el valor de *DEBLEND_NTHRESH1* con valor pequeño (abajo de 16) y *DEBLEND_MINCONT1* con valores grandes (arriba de 0.1)[1] para evitar que objetos sean divididos en sub-objetos.

Como se desea que el catálogo **hot** contenga solamente a los objetos brillantes y grandes, se recomienda poner los valores de *ANALYSIS_THRESH1*, *DETECT_THRESH1* y *DETECT_MINAREA1* con valores grandes (20, 20 y 100 respectivamente) muy por arriba de la señal a ruido.

Los parámetros *BACK_SIZE1* y *BACK_FILTERSIZE1* ayudan a obtener una mejor estimación sobre los límites de cada objeto por lo que se recomienda poner con valores grandes (alrededor de 100 y 11 respectivamente).

Cabe señalar que los valores exactos de las variables mencionadas arriba variarán entre las diferentes imágenes. Lo recomendable es estimar mediante observación con DS9 y el archivo de regiones de los objetos finales (archivo dado por RegDs9).

Finalmente, un ejemplo de los valores de los parámetros para una ejecución de **sextractor** para obtener el catálogo **hot** se muestra a continuación:

```
DEBLEND_NTHRESH1 32
DEBLEND_MINCONT1 0.01

ANALYSIS_THRESH1 20
DETECT_THRESH1   20
DETECT_MINAREA1  10

BACK_SIZE1      100
BACK_FILTERSIZE1 11
```

Por otro lado, la idea de la segunda ejecución con **sextractor** (el catálogo **cold**) es detectar los objetos débiles, pequeños y que probablemente están

[1] checar manual de **sextractor** para una mayor interpretación de estos parámetros

alrededor de objetos grandes. Para lograr esto, se recomienda poner el valor de *DEBLEND_NTHRESH2* con valor grande y *DEBLEND_MINCONT2* con valor pequeño[2] (64 y 0.001 respectivamente).

Para garantizar la detección de objetos débiles y pequeños, los valores de *ANALYSIS_THRESH2*, *DETECT_THRESH2* y *DETECT_MINAREA2* deben tener valores pequeños apenas por encima de los valor de la señal a ruido (valores de 1.5, 1.5 y 10 respectivamente).

Es recomendable tener los parámetros de *BACK_SIZE2* y *BACK_FILTERSIZE2* con valores pequeños (10 y 2 respectivamente).

Finalmente, un ejemplo de valores de los parámetros para una ejecución de **sextractor** para obtener el catálogo **cold** se muestra a continuación:

```
ANALYSIS_THRESH2 1.5          # <sigmas> or <threshold>,<ZP> in mag.arcsec-2
DETECT_THRESH2   1.5          # <sigmas> or <threshold>,<ZP> in mag.arcsec-2
DETECT_MINAREA2  10      # minimum number of pixels above threshold

DEBLEND_NTHRESH2 64           # Number of deblending sub-thresholds
DEBLEND_MINCONT2 0.001         # Minimum contrast parapymeter for deblending

BACK_SIZE2       10
BACK_FILTERSIZE2 2
```

Al terminar de ejecutarse pysex, este abrirá ds9 con la imagen y junto con esta se cargará el archivo de las regiones del catálogo final junto con el de los objetos saturados (archivos señalados por los parámetros RegDs9 y SatDs9). En este punto el usuario puede verificar si los objetos fueron correctamente detectados.

Por el otro lado, si se desea verificar que los catalogos **hot** y **cold** detectaron las fuentes deseadas uno puede accionar las banderas *FirstRun* y *SecondRun* para que ds9 muestre solamente el catálogo **hot** (bandera FirstRun) o **cold** (bandera SecondRun).

Los valores de los parámetros explicados en está sección pueden variar para diferentes tipos de imágenes. Sin embargo, la idea principal es la misma.

7 pysex3 y pysexbcg

Pysex3 y pysexbcg tienen el mismo funcionamiento que pysex, solamente que estos ejecutan **sextractor** 3 veces, esto con la finalidad de garantizar una mejor recuperación de las fuentes. pysex3 y pysexbcg se recomienda para aquellas ocasiones en las cuales pysex no pudo detectar satisfactoriamente todos los objetos a pesar de utilizar cualquier configuración.

Para el caso de **pysex3**, se adicionan los siguientes parámetros:

ThirdRun

[2]checar manual de **sextractor** para una mayor interpretación de estos parámetros

DEBLEND_NTHRESH3

DEBLEND_MINCONT3

ANALYSIS_THRESH3

DETECT_THRESH3

DETECT_MINAREA3

BACK_SIZE3

BACK_FILTERSIZE3

El principio de funcionamiento es el mismo: todos los objetos de la primera ejecución son añadidos, y después se agregan todos aquellos de la segunda ejecución que no están en el primer catalogo. Finalmente, todos aquellos objetos en el tercer catálogo que no están en la combinación de los primeros dos.

Para el caso de **pysexbcg** el objetivo es usarlo en imágenes de cúmulos de galaxias. En estas normalmente existe una galaxia que sobresale de las demás en cuanto a tamaño y brillo y se llama bright cluster member(BCG). La configuracion de la tercera ejecución de **sextractor** debe estar diseñada para detectar la o las galaxias BCG y añadirlas al catálogo final que resultó de la combinación de las dos primeras ejecuciones de **sextractor**.

Los parámetros para **pysexbcg** son los mismos que **pysex3** con la excepción que existe un parámetro adicional que es *BCG*. Este le indica al programa cuantas galaxias de la ejecución para detectar BCG son agregadas al catálogo final. Normalmente este valor es 1 o 2, dependiendo de la cantidad de BCG que se encuentren en la imagen.

8 Ajuste Manual

Si por alguna razón, no se puede encontrar alguna configuración de **pysex** para obtener una mascara para algunos objetos en una imagen, se puede hacer ciertos ajustes manuales con *MaskBox.py* y *remove.py*.

MaskBox.py se utiliza para poner (o quitar) una máscara en una imagen. Primero se usa la region box de DS9 para seleccionar el área de la imagen que se quiera remover. La forma de usar *Maskbox* es la siguiente:

```
./MaskBox.py [ImageFile] [RegFile] [Value]
```

ImageFile es la imagen máscara, RegFile es el archivo de región de DS9. Value es el valor númerico que se le quiere dar al valor de flujo que va a cubrir el área de la máscara

Por otro lado, *remove.py* remueve objetos de la imagen máscara. Es algo parecido, a *MaskBox*, pero este remueve máscaras elipse de la imagen creada por **pysex**. Para usarlo:

```
./remove.py [Mask] [SexCatalog] [Number] [Scale optional]
```

Mask es la imagen máscara creada por **pysex.py** (o *makemask.py*), SexCatalog es el catálogo de sextractor creado por **pysex** (o sextractor), Number es el número del objeto que se quiere remover, este número debe ser el mismo que le corresponde en el catálogo de sextractor.

Finalmente, si solo se desea obtener imágenes máscara el programa *makemask.py* lo puede hacer a partir del catálogo generado por **pysex** (o sextractor si contiene las mismas columnas). Para ejecutarlo:

```
> ./makemask.py [SexFile] [ImageFile] [MaskFileOut] [scale]
```

SexFile es el archivo de Sextractor, *ImageFile* es la imagen original y *MaskFileout* es el archivo máscaras de salida. *scale* es el factor de escala el cual serán agrandadas las máscaras de cada fuente.

9 ¿Preguntas? ¿Sugerencias?

Cualquier pregunta, sugerencia o reporte de un bug mandar mail a canorve@gmail.com

References

E. Bertin and S. Arnouts. SExtractor: Software for source extraction. *AA*, 117: 393–404, June 1996. doi: 10.1051/aas:1996164.

B. Häussler, D. H. McIntosh, M. Barden, E. F. Bell, H.-W. Rix, A. Borch, S. V. W. Beckwith, J. A. R. Caldwell, C. Heymans, K. Jahnke, S. Jogee, S. E. Koposov, K. Meisenheimer, S. F. Sánchez, R. S. Somerville, L. Wisotzki, and C. Wolf. GEMS: Galaxy Fitting Catalogs and Testing Parametric Galaxy Fitting Codes: GALFIT and GIM2D. *APj*, 172:615–633, October 2007. doi: 10.1086/518836.