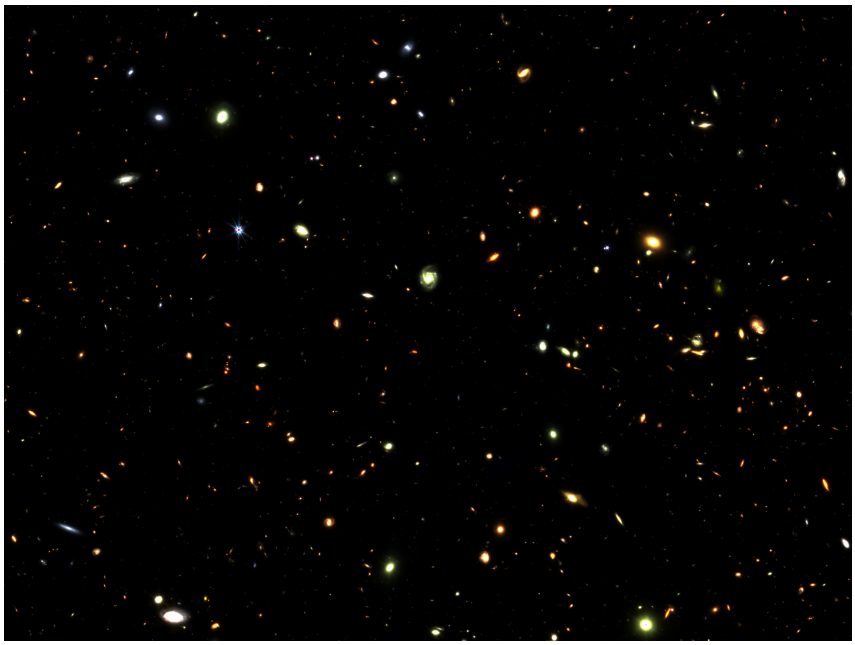
Research internship report

Morphology of Galaxies using SourceXtractor++ on the DAWN JWST Archive



Aurélien Genin
Cycle Ingénieur Polytechnicien - X2021

Cosmic DAWN Center
04/2024 - 08/2024

Supervised by Marko Shuntov

Image : JWST composite image in the GOODS-S field by the JADES collaboration

Abstract

Understanding how galaxies evolve through cosmic times can bring light on the history of our universe. An axis of study is to look at the morphology of galaxies at various redshifts and how it changed. During my 4 months internship at the Cosmic DAWN Center, I used SourceXtractor++ to run brightness profile model fitting on major fields observed by the James Webb Space Telescope and made available on the DAWN JWST Archive (DJA): CEERS, GOODS, PRIMER-UDS, PRIMER-COSMOS. This enabled me to measure the morphology of more than 340k galaxies at redshifts $0 < z_{phot} < 12$. The model fitting was done independently for both with Sérsic profiles and Bulge+Disk models. It made use of Amazon Web Services to run automatically SourceXtractor++ on the different fields with enough computing power. Using the redshifts measurements produced by EAZY and available on the DJA, we were able to study the size evolution of galaxies through cosmic times, and also the evolution of their disk and bulge. We recover well some morphological scaling relations that are well established in the more local Universe, which serves as a validation of our methodology. Furthermore we investigate the redshift evolution of the UVJ diagram and its dependence on galaxy morphology. We find that at lower redshifts $z < 4$, there is a clear bimodality between bulge-dominated quiescent population and disk-dominated star-forming, with this bimodality becoming less prominent at $z > 4$, potentially reflecting a transition period from star-formation to quiescence. We show that the bulge-dominated population grows later than the disk-dominated one, and from the bulge and disk at the same time. The disk-dominated population seems to grow only by the disk, with a bulge even shrinking. This work is a highly valuable addition to the DJA, adding a morphological dimension to this rich dataset and thus enabling a wider scientific application.

Keywords: Astronomy, SourceXtractor++, Galaxy morphology, Catalogs, Size evolution, Quiescent galaxies

Contents

1 Introduction

The James Webb Space Telescope (JWST) is revolutionizing astronomy by offering observations deeper, and earlier, than ever before. To measure how far an object is, it is common to talk about its redshift z. It is simply calculated by $1 + z = \lambda_{obsv}/\lambda_{rest}$, where λ_{rest} is the wavelength of the light emitted by the source, in its rest frame of reference, and λ_{obs} is the observed wavelength from Earth (or space telescopes). Because of the expansion of the universe [Hubble, 1929], the higher the redshift is, the farther, and therefore the older, this object is.

Before the JWST, the furthest known object was GN-z11, detected by the Hubble Space Telescope (HST), at a redshift $z = 11.09$ [Oesch et al., 2016], observed as it existed 400 million years after the Big Bang. The JWST, after only two years of scientific observations, has already detected and spectroscopicaly confirmed a $z = 14.32$ galaxy [Carniani et al., 2024], only 290 million years after the Big Bang, and has already showed potential $z \sim 16$ candidates in photometry [Atek et al., 2022].

The Cosmic DAWN Center is specialized in studying the extremely early universe to try to understand it better. This epoch is usually called the epoch of reionization (EoR) or the cosmic dawn, hence the name of the center. DAWN is an international research center supported by the Danish Research Foundation. It is located at the Niels Bohr Insitute, University of Copenhagen (KU), and at the National Space Institute, Technical University of Denmark (DTU Space). To study how and when the very first galaxies, stars and black holes formed, the researchers from DAWN can rely on observations with the best telescopes of the time : ALMA, HST, JWST, ELT, Euclid...

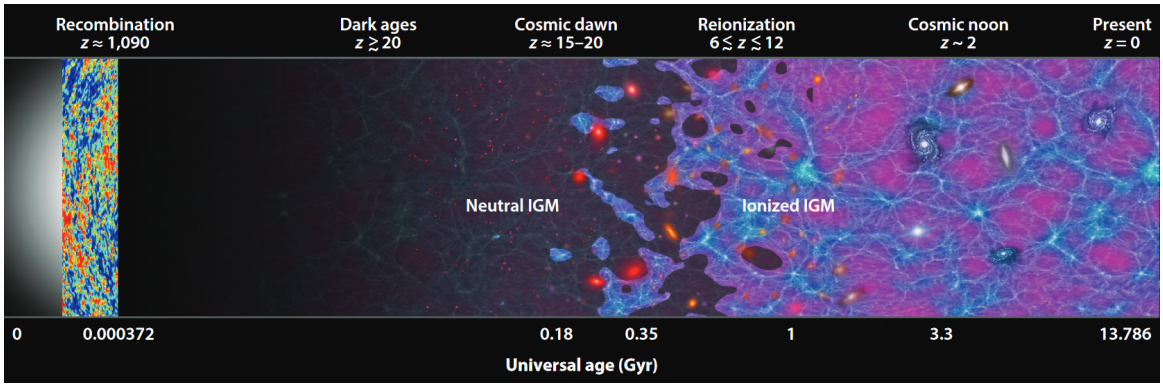


Figure 1: Overview of the history of the universe from the Big Bang to the present, highlighting the epoch of reionization. This epoch is the transition between a time in each all the intergalactic medium (IGM) was neutral, to a time in which it is fully ionized by stars, galaxies and AGNs. Figure from Robertson [2022].

To help researchers around the world exploit the data of the JWST, the Space Telescope Science Institute (STScI), an organisation founded by NASA to lead science with HST and JWST, developed differents tools and pipelines for data reduction [Bushouse et al., 2024]. This encompasses the different stages of image calibration : corrections for dark current, flagging bad pixels, convert pixel values (resp. coordinates) to photometric flux (resp. sky coordinates), etc. More details can be found on the JWST documentation.

These pipelines are made to work for all cases with minimal parameters changes. This means that almost the same methods are used to reduce images of nebulaes, deep extra-galactic fields, planets, exoplanets... Although these pipelines work generally very well, it's possible to develop more specialized pipelines for specific use cases. This lead DAWN

to develop the DAWN JWST Archive (DJA), a repository of public JWST galaxy data, released for use by anyone [Brammer, 2023].

The DJA hosts data from all the major deep (high z) surveys of the JWST : COSMOS, JADES, CEERS, FRESCO... Furthermore, it contains photometric data as well as spectroscopic data. The photometric data comes from the NIRCam instrument [Rieke et al., 2005] and is made of images taken with different color filters and mosaiced to cover the whole field. DAWN has also produced photometric catalogs using SExtractor from Bertin and Arnouts [1996].

These catalogs offer researchers valuable ready-to-use science data such as the flux and magnitude through different filters of all the sources in a field. The DJA notably uses these fluxes to estimate the redshift z of the different sources. For that, it uses the `EAZY` package made by Brammer et al. [2008] to fit a galaxy spectrum template to the photometric fluxes in different wavelength bands and estimate a so-called photometric redshift. Though less precise than a spectroscopic redshift (directly measuring the difference between the observed wavelength of a spectral feature and its rest-frame wavelength), it can be done on a scale many orders of magnitude bigger.

The goal of my research internship at DAWN was to expand the DJA by adding valuable morphology measurements to the catalogs. This will enable researchers to perform statiscal studies on the morphology of galaxies at high redshift and learn how it evolved during the history of the universe. To measure these morphologies, we chose to use SourceXtractor++ from Bertin et al. [2020].

2 JWST and DJA

2.1 The James Webb Space Telescope

The James Webb Space Telescope (JWST) is the latest big space telescope, launched in space on 25/12/2021. It has been developed by the National Aeronautics and Space Administration (NASA), with contributions by the European Space Agency (ESA) and the Canadian Space Agency (CSA). This new space observatory is designed to conduct infrared astronomy and is stationed at the Sun-Earth L_2 Lagrange point to avoid Earth eclipses.

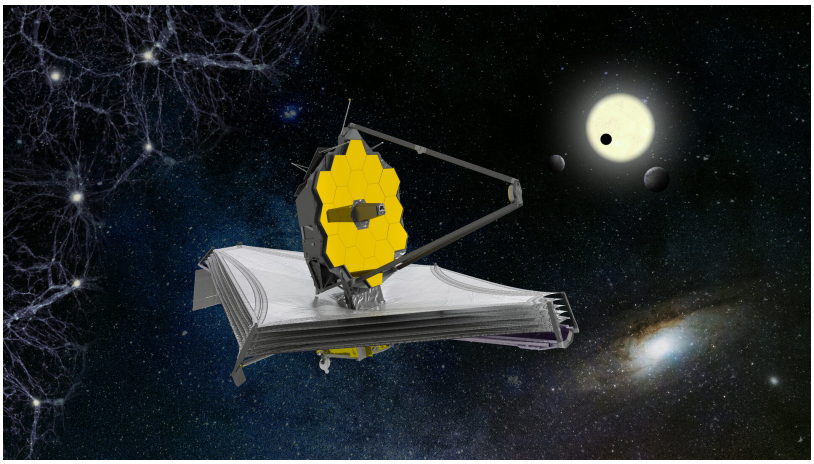


Figure 2: Artist's impression of the James Webb Space Telescope.
Credit: ESA/ATG medialab

Its giant 6.5m-diameter mirror (compared to Hubble's 2.4m-diameter mirror) consists of 18 hexagonal segments that can be adjusted to work as a single one. The JWST can observe in the near- and mid-infrared wavelengths, from 0.6µm to 27µm. It can also perform spectroscopy, making it perfectly suited from a vast range of research subjects.

Webb hosts four scientific instruments, all made for astronomy:

- **NIRCam** (Near Infrared Camera) is the main imager of the JWST, developed by NASA. It has 2 identicals sets of 5 sensors, 4 for wavelengths between 0.6µm and 2.3µm (SW channel) and 1 for wavelengths between 2.4µm and 5.0µm (LW channel). The field of view of NIRCam is 2x2.2′x2.2′. To enable photometric measurements, it has many different filters, centered at different wavelengths and of different bandwidths. More details are provided in Rieke et al. [2005]. For my work, I used only NIRCam data, although it can very easily be extended to the other photometric instruments.

- **MIRI** (Mid Infrared Instrument) provides imagery and spectroscopy from 4.9µm to 27.9µm. It has been developed jointly by NASA and ESA and has an imaging field of view of 1.2′x1.9′. Its resolution is much smaller than the one of NIRCam, but it provides important insight in the characterisation of galaxies at redshift $z > 7$. More details are provided in Rieke et al. [2015].

- **NIRSpec** (Near Infrared Spectrograph) enables spectroscopy in the near infrared (0.6µm to 5.3µm) with resolving powers from 100 to 2700. It was developed by ESA. Thanks to its micro-shutter array, it can measure simultaneously the spectrum of

up to 100 objects. It also features an Integral-Field Unit (IFU) to acquire spatially resolved spectroscopy over a 3″x3″ region. More details are available in Jakobsen et al. [2022].

- **NIRISS** (Near Infrared Imager and Slitless Spectrograph) is an imager provided by the CSA which offers an identical wavelength range and field-of-view as NIRCam (with only one sensor). It enables slitless spectroscopy, by using a grism, an optical element than disperse the light from a whole image. This allows to gather low resolution spectrum from all the sources in a frame, for example for batch redshift estimation. NIRCam also has grisms. More details are available in Doyon et al. [2012].

2.2 JWST fields and the DJA

Research with the JWST works with a system of project and observation proposals. For now, there have been three cycles of call of proposals, where scientists from all around the world can present a project of research to be conducted with the JWST. All the projetcs are then reviewed to select the ones that will actually be realised. Principal Investigators have the choice to disclose the data acquired by the space telescope (typically the raw images) immedialty after they were taken so that the whole scientific community can study them, or request an exclusive access period of up to 12 months. The DJA only focuses on public data, either from the first case or after the exclusive access period, to share public science-ready data to the whole scientific community.

On the DJA, one can find many mosaics of different surveys performed by the JWST. These surveys are observations of a specific area of the sky, called a field, to study a vast number of sources (typically galaxies in our case). This enables statistical studies over tens or hundreds of thousands of galaxies at various redshifts. It is also the best way to find new interesting sources for follow-up observations, for example with spectroscopy. These surveys are generally done in fields also covered by other telescopes (both on ground and in space). Because each telescope is specialized in a specific range of the electromagnetic spectrum, having common fields allow for obsversations in a wide range of wavelengths (from radio waves to X-rays), in term allowing to see and understand different things.

The widest field for extragalactic studies is the COSMOS field. It covers a 2deg^2 region of the sky, and contains informations in all wavelengths, thanks to a huge collaboration of telescopes from all around the globe, and in space:

- X-ray: XMM-Newton and Chandra space telescopes;
- Ultraviolet: Galex space telescope;
- Visible light: Hubble space telescope, SDSS, Subaru, CFHT, VISTA... ground telescopes;
- Near and mid infrared: Spitzer and JWST space telescopes, Keck ground telescope;
- Far infrared: Herschel space telescope;
- Sub-millimeter: Alma ground telescope;
- Radio: VLA, VLBI, GMRT ground telescopes;

These observations provide very valuable data for research in the extragalactic field by giving insights into all the different parts of the spectrum of many different sources, at many different times of the history of the universe.

To avoid bias that might be caused by using a single area of the sky, other fields exist. The most famous, and the one hosting all the record-breaking sources in term of redshift

is the Great Observatories Origins Deep Survey, also knwow as GOODS (or Hubble Deep Field, as it has been observed for the first time in details by the Hubble Space Telescope). Much smaller than COSMOS, it consists of two regions, GOODS-S and GOODS-N, of 160arcmin^2, or 0.044deg^2. However, it looks deeper in the universe and allows for very exciting discoveries. GOODS is the target of multiple surveys, such as JADES (JWST Advanced Deep Extragalactic Survey, Rieke et al. [2023]) and FRESCO (First Reionization Epoch Spectroscopically Complete Observations, Oesch et al. [2023]). The footprints of the main fields and surveys observed by JWST are presented in figure 3.

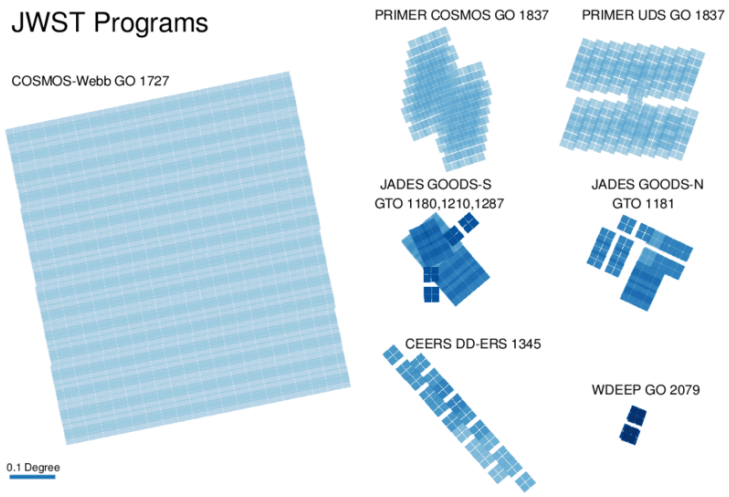


Figure 3: Footprints of the major extragalactic fields observed by JWST at the same scale. This figure shows the NIRCam coverage. Figure from Robertson [2022].

The DJA publishes mosaics of these different surveys, in the different filter bands used. These surveys are conducted using wide or medium filters primarly, I therefore focused on these. Their transmission curves and names can be found on figure 4. The use of multiple filters, also know as multifilter photometry, has many interests. Maybe the least scientific one, but the most important for science communication and raising interest for astronomy in the general public, is the possibility of creating color images out of the raw, black and white, frames. For scientists, the filters allow to determine the color (difference in brightness in different filter bands) of the observed sources. This can be used for example to classify galaxies and determine the ones forming new stars (which tend to be bluer thanks to the young hot stars) and the ones which stopped star formation, known as quiescent galaxies.

Another major use of multifilter photometry is to use the flux measured in different bands as a very low resolution spectrum of a source. It's then possible to fit a model of a spectrum (many templates exist depending on the type of source: star, galaxy, Active Galaxy Nuclei (AGN), etc) to these points to calculate more information about the target, such as redshift, metallicity, dust content, mass, star formation rate (SFR)...

An even simpler way to understand how multifilter photometry can be used to estimate redshift (for high z sources), is to look at the Lyman break. This denotes a distinctive step in the spectral energy distribution (SED) of a galaxy at 912Å, the wavelength corresponding to the energy needed to ionise an hydrogen atom from the ground state. Any radiation at wavelengths lower than that is almost completely absorbed by neutral gas in the galaxy, or in the path to us. If the redshift is high enough, this Lyman limit can be pushed into

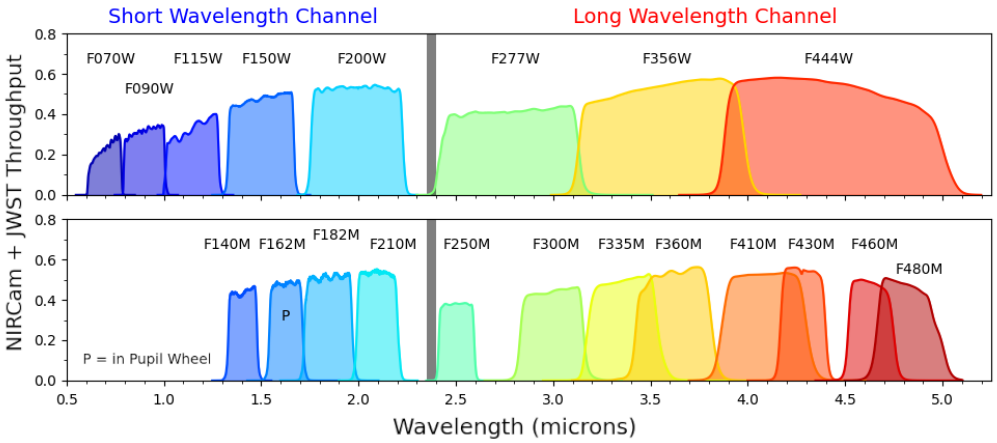


Figure 4: Transmission curves of the wide and medium filters of NIRCam. Adapted from the STScI/JWST documentation.

the visible domain or even infrared (at $z > 9$, a galaxy disappears in JWST's F090W band). A similar effect appears at 3646Å, the Balmer limit, caused by young stars. By being redder, this limit can be used for lower redshifts. These effects are illustrated on figure 5.

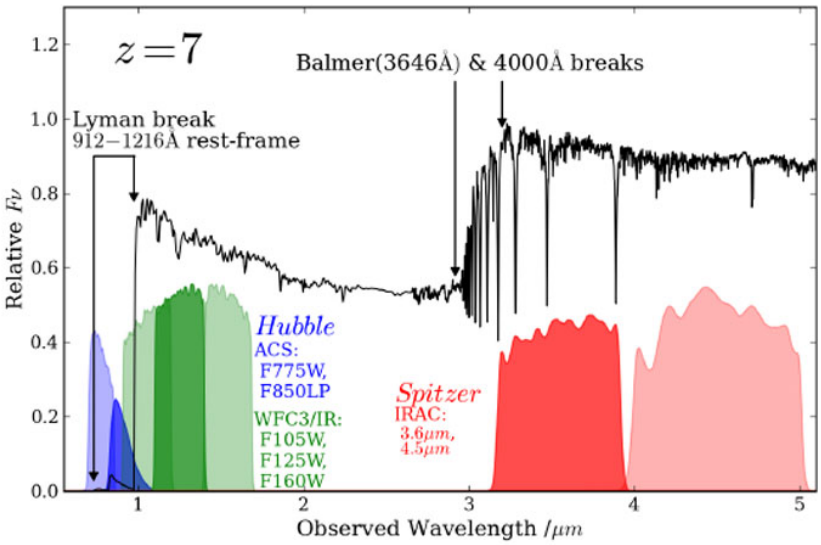


Figure 5: Redshifted SED of a young galaxy at $z = 7$, showing the Lyman and Balmer breaks. Here, the galaxy would be invisible to Hubble/ACS but visible to Hubble/WFC3. The JWST samples the whole range of wavelengths shown on this figure. This figure is the courtesy of S. Rogers.

3 DJA and SourceXtractor++

3.1 Galaxy morphology

Astronomical images have a lot more information than just the measure of flux and magnitude[1]. Thanks to the spatial resolution of the images, we can look at the morphology of the sources, a fancy way to talk about their shape. This is especially useful for galaxies as the morphology can teach us a lot about the things happening in it, and about its history.

First, let's review the basics about galaxies. They are dynamically bound systems consisting of stars, gas and dust, embedded in a dark matter halo. By definition, the dark matter can't be directly observed, but its gravitational effects (which led to its theoretical creation) can be seen. The most famous is the galaxy rotation curve. With classical Newtonian gravitation, we would except the tangential velocity of stars further from the galaxy's center to decline. However, we observe that this speed stays pretty constant with distance, which would require much more mass than the one we can observe (from stars, gas and dust). This extra mass is called the dark matter, and gives rise to the Λ-CDM model of the universe. Other theories such as MOND (Modified Newtonian Dynamics) or AQUAL (AQUAdratic Lagrangian) try to tackle this issue by modifying the law of dynamics, but are not yet as good as Λ-CDM to match the observations in very different situations.

Galaxies exist in various sizes, masses, colors, and... shapes. They can be usually classified in two general classes: elliptical galaxies (system of old stars, not producing new ones) and disk galaxies (complex structures with star forming regions and possibily AGN (Active Galaxy Nucleus)). Historically, the morphology study of galaxies was done with qualitative classifications. The two main schemes were the Hubble's one and the de Vaucouleurs' one, which associated a type to a galaxy depending on the features that could be seen in it (see figure 6).

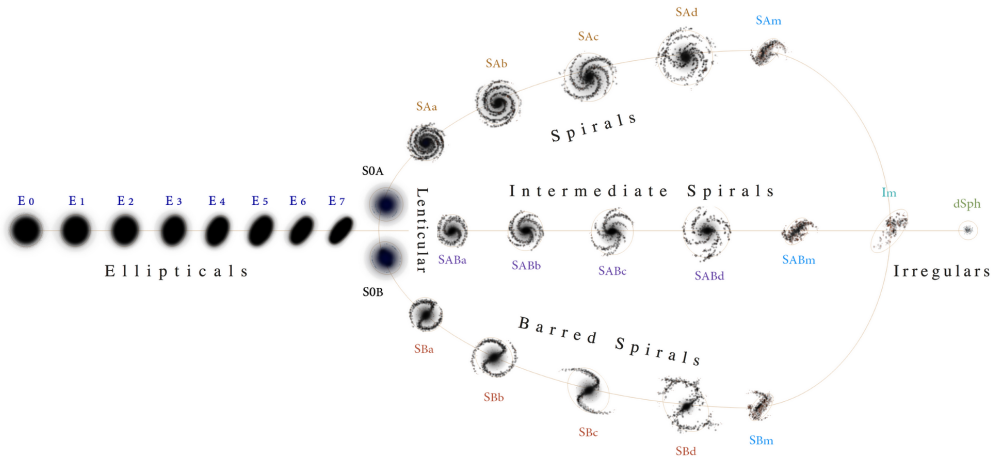


Figure 6: Hubble-de Vaucouleurs galaxy classification scheme.
Adapated from Antonio Ciccolella / M. De Leo.

[1]The magnitude is the flux on a logarithmic scale, where lower values are the brighter objects. It's calculated by $M = -2.5 * \log_{10}(f) + cst$

A more quantitative way to define the morphology of a galaxy is to look at its surface brightness profile $I(r)$. This is a measure of the flux concentration (mag/arcsec^2) as a function of the distance to the galactic center. It is then possible to define different models to quantify by one (or more) number(s) the morphology of a galaxy. The simplest, with only one number, is the Sérsic profile : $I(r) = I_e \exp\left\{-b_n\left[(r/r_e)^{1/n} - 1\right]\right\}$. The shape parameter n is therefore called the Sérsic index of the galaxy. However, galaxies, especially disk galaxies, generally don't fit this profile very well because of a central bulge, as depicted on figure 7. In this case, it is possible to combine two Sérsic (or other) profiles, so that one will model the bulge, and the other one the disk. This gives shape parameters, that can give a disk and bulge radii and the ratio between them to show the relative importance between them. These two models are discussed further in section 3.3, as they are the two I used for my work.

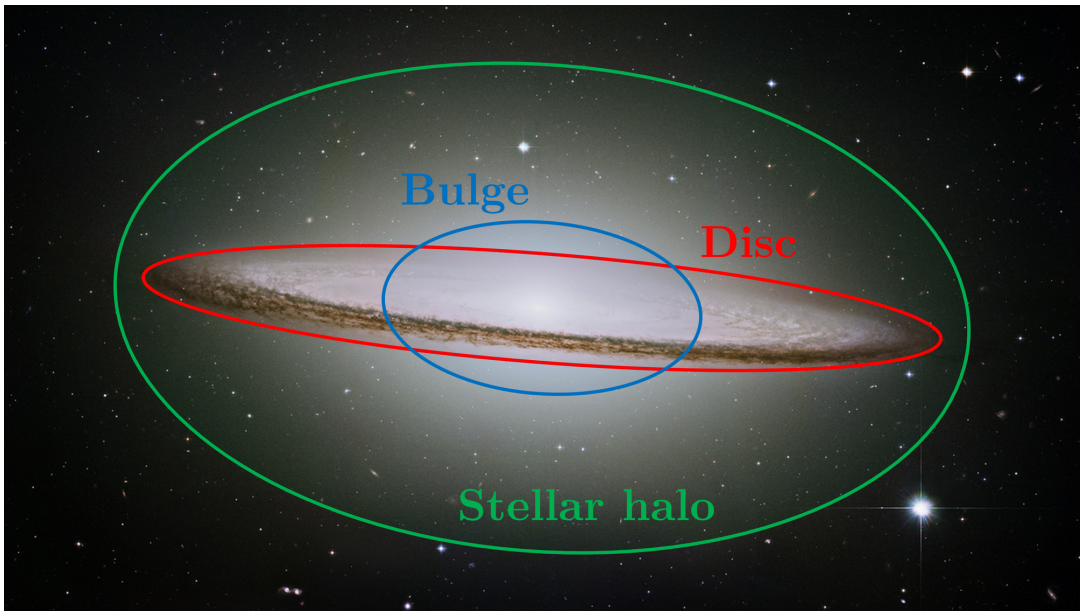


Figure 7: Classic anatomy of a galaxy. Base image is M104, the Sombrero galaxy, by NASA / Hubble Heritage Team (STScI/AURA).

3.2 PSF estimation

There exists different programs made by researchers all around the world to fit these models to galaxies. One of the most used is GALFIT by Peng et al. [2002]. It can fit very different models and even be used to analyse different components of a galaxy such as the nucleus, the stellar halo or the spiral arms. However, GALFIT is made for fitting only a few galaxies at a time, not the tens of thousands present in a field. It's designed for detailed studies of a few galaxies rather than statistical studies over full fields.

Two tools that allow for batch fitting of galaxies are `The Farmer` by Weaver et al. [2023] and SourceXtractor++ by Bertin et al. [2020]. Their differences will be presented in section 3.3. A common point with all these programs, fitting models to sources, is that they need to know the Point Spread Fucntion (PSF) of the telescope.

3.2.1 Point Spread Function

The PSF is the response of an optical instrument to an impulse. It shows how a perfectly point-like source appears on an image taken with the instrument. Because of diffraction or defects in the optical path (irregularities on the optical elements, dust or ice...), point-like sources such as distant stars never appear as perfect points on astronomical images. They usually appear spreaded and more blurred. The diffraction also explains the famous spikes that can be seen on bright stars observed by telescopes: it is caused by the supports of the secondary mirror which are in the path of the incoming light and cause diffraction (see figure 8). The spikes in the JWST's images are even more important because of the hexagonal mirror segments, also causing diffraction.

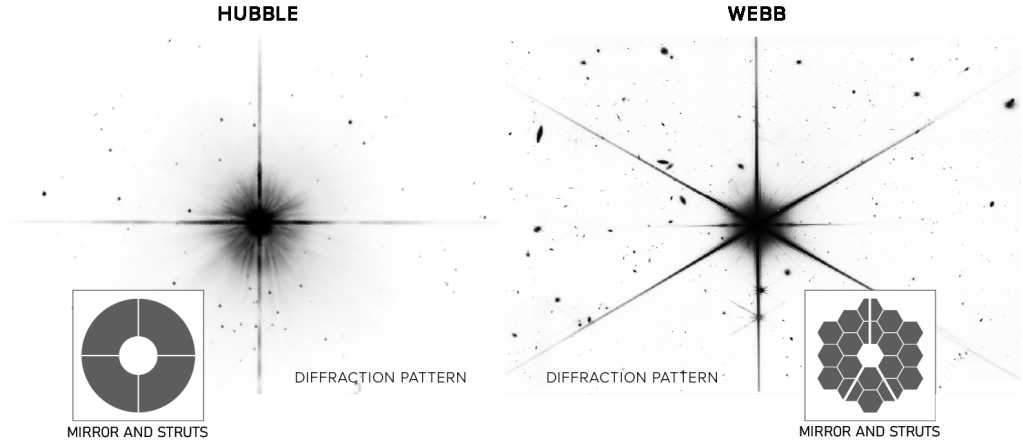


Figure 8: Point Spread Functions of HST and JWST observed on bright stars.
Adapted from NASA, ESA, CSA, Leah Hustak (STScI), Joseph DePasquale (STScI).

Knowing this PSF is crucial for model fitting sources. Mathematically, the acquired image is the convolution of the real astronomical sources and the PSF. Therefore, to fit a brightness model to a source, the program needs to calculate a model and then convolve it with the PSF in order to simulate what the telescope would see of this modeled source.

In order to get the PSF of a telescope, it's possible to simulate it. The easiest method uses a 2D Fourier transform of the aperture and mirrors of the telescope, known as the Fraunhofer diffraction model. More advanded simulations also exist and give more accurate results by taking many more physical effects into account, such as mirror imperfections, multiple optical pupils or masks. For example, for the JWST, STScI developed WebbPSF [Perrin et al., 2012] to help researchers with observation planification and data analysis.

However, it is also possible to calculate an empirical PSF from real observations. This can give better results as it's directly based on the images taken by the telescope and doesn't require prior modeling of the defects of the instrument. The JWST regularly image a bright star for that purpose. Howwever, it's also possible to use regular images, from scientific observation campaigns, to estimate this PSF. This has the advantage of giving the PSF in the exact same observation conditions (rotation, flextures, light leaks...) as the scientific images.

For this, I used the PSFEx software by Bertin [2011] in combination with SExtractor. SExtractor (short name for Source-Extractor) is a software that can detect sources (stars, galaxies, asteroids...) in astronomical images and perform photometric measurements (flux and magnitude) of such sources. Moreover, it's possible to use it to generate vignets of

a few tens of pixels (151x151px for my work) of all the sources. These vignets are then stacked by PSFEx to generate the PSF of the optical instrument used.

3.2.2 Point-like sources selection

As explained before, the PSF is the response of the telescope to a point-like source. It's therefore crucial to identify the point-like sources before giving the vignets to PSFEx. PSFEx has an integrated filter to distinguish point-like (e.g. stars) from extended (e.g. galaxies) sources. It is based on the radius of the source on the image and on its signal-to-noise ratio (SNR). The hypothesis is that point-like sources have a low radius and a high SNR (by being brighter so that the PSF, e.g. diffraction spikes, is very visible). This filter is good, but can sometimes be contaminated by bright extended sources or miss some dimmer but still great stars.

To improve PSFEx's point-like sources selection, I developed my own filter. It is based on [Leauthaud et al., 2007, Section 3.6]. The idea is to look at the `MU_MAX/MAG_AUTO` plane where `MU_MAX` and `MAG_AUTO` are two measurements performed by SExtractor for every source. `MU_MAX` is the maximal pixel value of the source, converted to mag. `MAG_AUTO` is one of the measurements of magnitude performed by SExtractor and expressed in mag/arcsec^2. What we observe is that point-like sources lie on a very well defined line (which I call the star-line, see figure 9), whereas the extended sources form a cloud of points. An improvment I have made compared to Leauthaud et al. [2007] is to add a maximal threshold for `MAG_AUTO`. In their work, the star-line becomes flat for smaller values of `MAG_AUTO` which is in fact the consequence of saturation on the images. It is not desirable to keep saturated sources for the calculation of the PSF as they will bias the PSF by having a truncated maximum.

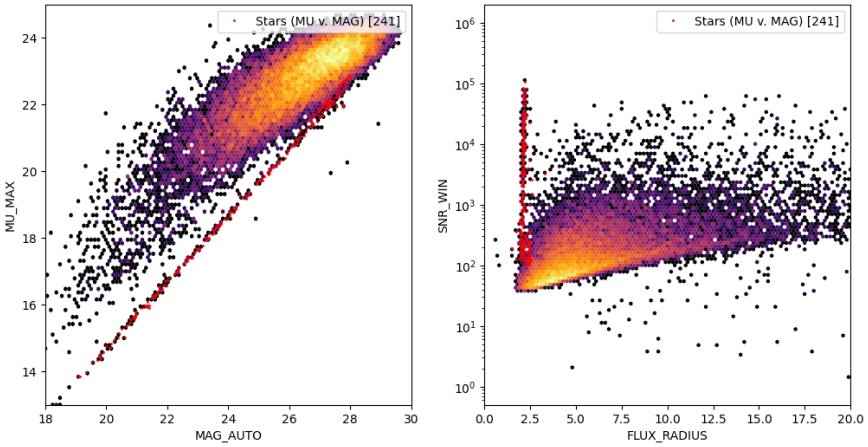


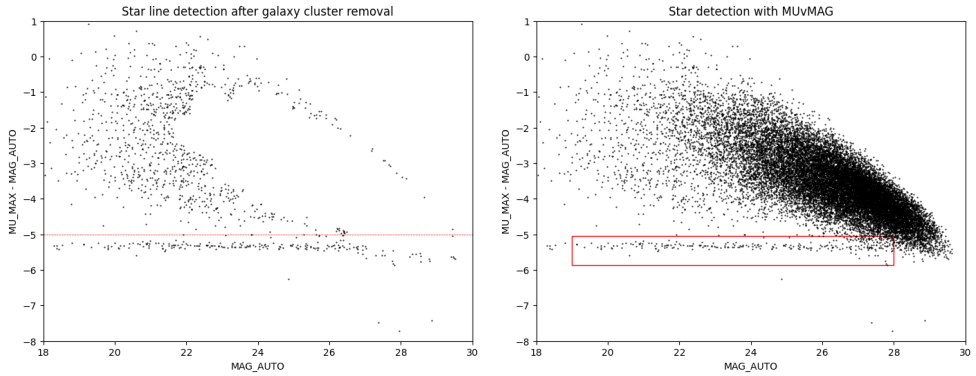
Figure 9: Point-like sources selection for PSF estimation. The left plot shows the `MU_MAX/MAG_AUTO` plane and the star-line is shown with red dots. The right plot shows the `SNR/RADIUS` plane used by PSFEx's selection. The separation between point-like sources and extend sources is clearer on the `MU_MAX/MAG_AUTO` plane. The data used for these plots comes from the GOODS-S field observed with JWST in the F200W band.

To go in more technical details into how this star-line is found in the `MU_MAX/MAG_AUTO` plane, the goal was to use a linear regression to extract the star-line. However, a first step for that is to "un-bias" the plane by removing most of the extended sources cloud. To do that, I used `DBSCAN` from Ester et al. [1996] implemented in the `scikit-learn` Python package. `DBSCAN` stands for *Density-Based Spatial Clustering of Applications with Noise*.

It is a clustering algorithm very good at finding cores of high density (in our case, the extended sources cloud) while excluding points considered as too noisy (in our case, the star-line). The goal is therefore to find a singular cluster, that can then be removed to exclude the extended sources.

After this first clean-up step, it is possible to find the star-line by linear regression. However, a classic linear-regression using least squares would be plagued by all the extended sources left out by `DBSCAN`. Therefore, I used the `RANSAC` (Random Sample Consensus) algorithm by Fischler and Bolles [1981] implemented in `scikit-learn`. `RANSAC` is a noise-robust non-deterministic algorithm for linear regression. It performs really good even with outliers, which is exactly the case here. To help it further, I added a threshold to the `MU_MAX-MAG_AUTO` value in order to exclude most outliers. The star-line has a slope of 1 in the `MU_MAX/MAG_AUTO` plane, and is therefore represented by a single ordinate value in the `MU_MAX-MAG_AUTO/MAG_AUTO` plane. The RANSAC linear regression is therefore performed in this plane, after thresholding, as depicted on figure 10.

Once the star-line ordinate has been found by RANSAC, the point-like sources are selected using a box around this star-line. Its width around the star-line and length (mininum and maximum of `MAG_AUTO`) are manually chosen depending on the saturation and sensitivity of the telescope used.



(a) (b)

Figure 10: Process of star-line and point-like sources detection. (a) shows the removal of most extended sources using `DBSCAN` and the thresholding (dotted red line) using for `RANSAC`. Only the points under the threshold are given to `RANSAC` for linear regression. (b) shows the point-like sources selection box after the star-line detection.

This selection gives a catalog of point-like sources in the considered field, that can be used to empirically model the PSF. This selection should physically be the same for all the avalaible color bands. However, this algorithm can give slightly different results, and more concerning, the threshold used to exclude the outliers from the extended sources cloud can be different between the different bands. To account for that and produce a more robust program, I looked at the results for all the wide bands of JWST. I concluded that F200W gives the best and most consistent results. This can be understood by the fact that it is the reddest channel on the short-wavelength channel, therefore giving higher resolution compared to the long-wavelength channel, while also giving more details on extended sources (especially galaxies) compared to bluer channels. For the following work, I always used the F200W band as the point-like sources selection channel for the different images of a same field.

3.2.3 PSF results

This point-like sources selection allows a better PSF estimation using PSFEx. Some very technical issues with the very specific `FITS_LDAC` file format of the catalog required for PSFEx had to be overcame, but they are outside the scope of this report.

PSFEx, using the vignets of the point-like sources, creates an empirical PSF of the telescope, in the given field and with given color filter. The figure 11 shows the comparison of the PSFs obtained with the different color filters on the GOODS-S field.

We find, as expected the six branches shape, with an additional branch from the secondary mirror support. The PSF also appears "dotted", which is a direct evidence of the diffraction creating these PSFs as the different patches show the different orders of diffraction. Finally, we see that the PSFs grow with the wavelength, again a direct effect of diffraction as it is stronger for larger wavelengths.

An other interesting effect is to notice that the PSFs coming from the medium channels (FxxxM) are noisier than the ones from the wide channels (FxxxW). This is due to the medium filters allowing less light into NIRCam and therefore leading to a lower signal-to-noise ratio.

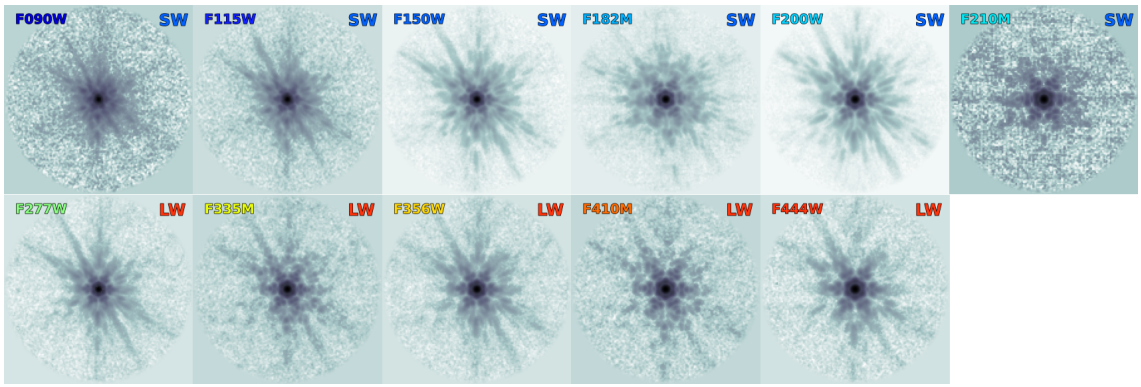
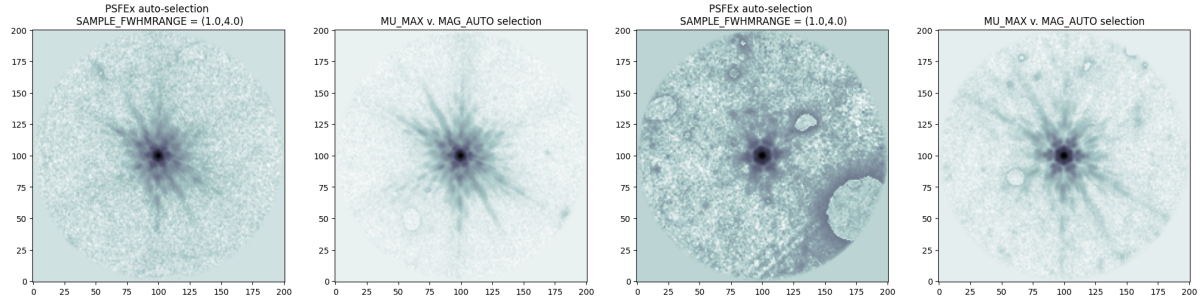


Figure 11: PSF calculated with PSFEx for the different bands of JWST on the GOODS-S field. The name of the filter is written in the top left corner of each image, and the name of the channel (SW=Short-Wavelength, LW=Long-Wavelength) is written in the top right corner. The images are displayed in logarithmic scale and color-inverted.

As a last step of validation of my method, I compared the PSFs I obtained with the ones calculated by PSFEx with its standard auto-selection. The results are generally pretty similar, but some examples show radical improvments in SNR and quality, as depicted in figure 12. These improvments come mainly from the exclusion of extended sources kept by PSFEx and the inclusion of more, lower SNR, point-like sources. Although these are lower SNR individually, by being more numerous, they give a global higher SNR PSF.

3.3 SourceXtractor++ model fitting

Using the PSFs calculated in section 3.2, it is possible to fit brightness profile models to the different sources in the JWST images. To do this on a large scale, there exists two main softwares in the astronomy world : `The Farmer` and SourceXtractor++. Their main differences are on the types of models they fit. With SourceXtractor++, it's possible to manually define any model the researcher want in a Python script. On the contrary, `The Farmer` has pre-defined models (point-source, Sérsic, Bulge+Disk...). However, `The Farmer` has an additional optimization step in which it finds the optimal model to fit each



(a) (b)

Figure 12: Comparisons of the PSF obtained using PSFEx standard selection and the `MU_MAX/MAG_AUTO` selection. These show the most radical examples, with the F115W filter on (a) and the F356W filter on (b). The abnormal patches on the PSFs come from SExtractor segmentation of sources.

specfic source. This way, it fits a star using a point-source and not an extreme Sérsic profile. This is also beneficial to optimize the calculation time by using simple models as much as possible.

3.3.1 Models

The modularity of SourceXtractor++ with the possibility to define any model led us to choose it rather than `The Farmer`. This gives more control on the constraints or modeling we want to use. As briefly introduced in section 3.1, I used two models for my work : Sérsic profile and Bulge+Disk.

Sérsic model The Sérsic model is the most basic, yet quite general, model for the brightness profile of galaxies. It has been published by Sérsic [1963] and is a generalization of one of the very first quantitative model: the de Vaucouleurs' law.

The Sérsic model is very interensting because it has a single parameter that controls the shape (the degree of curvature) of the brightness profile as seen on figure 13. This parameter n is called the Sérsic index. Additionally, two parameters, r_e and I_e, define the scaling of the brightness profile, respectively the half-light radius and the intensity (flux) at the half-light radius. The half-light radius is the radius of a circle centered on the galaxy center and enclosing half of the total flux of the galaxy. It is defined by

$$I(r) = I_e \exp\left\{-b_n\left[\left(\frac{r}{r_e}\right)^{1/n} - 1\right]\right\} \tag{1}$$

where b_n satisfies $\gamma(2n, b_n) = \frac{1}{2}\Gamma(2n)$[2] with γ the lower incomplete Gamma function[3] and Γ the Gamma function[4]. Most galaxies are fit with indices in the range $0.5 < n < 10$. Elliptical galaxies typically have high Sérsic index around 4, whereas spiral galaxy disks have a lower index around 1.

[2]Approximation for b_n good for $n > 0.36$: $b_n \simeq 2n - \frac{1}{3} + \frac{4}{405n} + \frac{46}{25515n^2} + \frac{131}{1148175n^3} - \frac{2194697}{30690717750n^4}$ by Ciotti and Bertin [1999]

[3]Lower incomplete Gamma function: $\gamma(s,x) = \int_0^x t^{s-1}e^{-t}dt$

[4]Gamma function: $\Gamma(s) = \int_0^{+\infty} t^{s-1}e^{-t}dt = \lim_{x\to\infty}\gamma(s,x)$

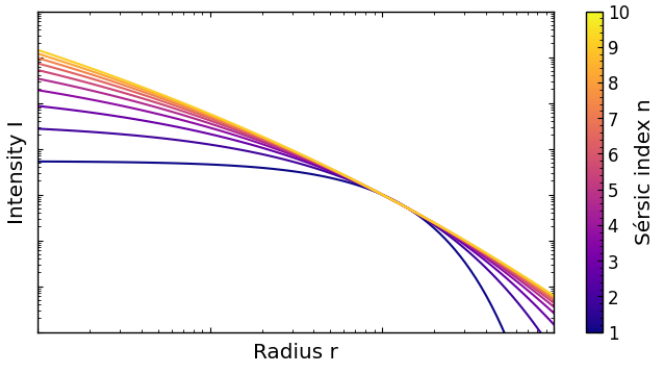


Figure 13: Sérsic profile (1) for $n \in [\![1, 10]\!]$ with same r_e and I_e. Given in logarithmic scale.

Bulge+Disk (B+D) The Sérsic model is quite good to fit a large variety of galaxies, but doesn't generally fit well the low-density cores of very bright elliptical galaxies, or the very bright bulge of other galaxies. To overcome this and get a better fit of the brigthness profile, we can define the Bulge+Disk model. This model is the sum of an exponential profile (Sérsic index $n = 1$) representing the disk, and a de Vaucouleurs profile (Sérsic index $n = 4$) representing the bulge. By adjusting the values of I_e and r_e for each model, this gives more accurate modeling of galaxies and especially their core as can be seen on figure 14. It can therefore be interesting to study the ratios of intensities and radii between the bulge and the disk, for example to detect candidates of AGN (Active Galaxy Nuclei).

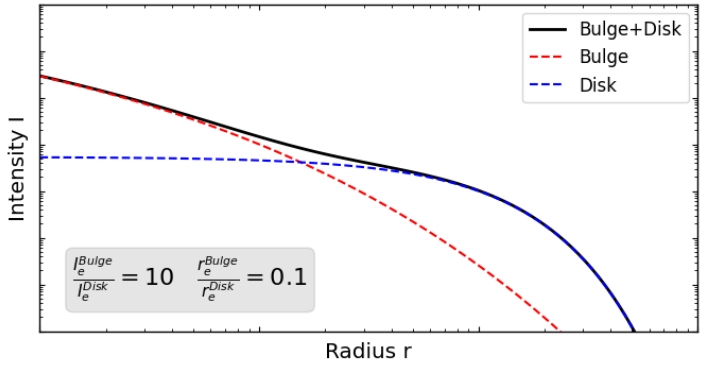


Figure 14: Example of a Bulge+Disk profile which could be used to represent a disk galaxy with a very luminous core. Given in logarithmic scale.

Although the equation 1 is only dependant on radial distance, we used a deformed one to fit ellipses rather than circles. This fits better galaxies that can be viewed inclined or that are intrinsicly elliptical. This therefore gives a measure of their aspect ratio. The same modification is done for the Bulge+Disk model, with two aspect ratios and inclination for the bulge and disk.

3.3.2 Utilization of SourceXtractor++

SourceXtractor++ has a rather complete documentation online. I sum up the main points of its operation in this section. SourceXtractor++ works in three main steps:

- Detection: the software detects the sources in the detection image (see below) and creates a segmentation map associating each pixel of the image to one source (this step also deblends close sources by separating pixels that appear to be part of both sources);
- Collection: nearby sources are grouped together to take into account the mix of fluxes from different sources contributing to a single pixel value;
- Measurement: finally, the software fits models simultaneously to each source in a group of sources and performs measurements such as aperture photometry[5] or isophotal measurements[6] on every measurement image (see below).

One important feature of SourceXtractor++ is the distinction between the detection image and the measurement images. As explained before, it is generally very useful to have photometric measurements in different color channels. For that, SourceXtractor++ enables measurements and model fitting simultaneously on different measurement frames, even if they have different pixel scales (the angular size in the sky represented by one pixel on the image), which is useful to combine data from different telescopes, or different instruments of a one telescope (on JWST, MIRI, NIRCam/SW and NIRCam/LW have different pixel scales.). It requires however one detection frame that is used in the two first steps (detection and collection) to find all the sources that will be fitted and measured in the last step. This detection image can be one of the measurement images, or better, a weighted sum of each measurement images where the weight for each image can be the inverse of the variance of the pixel values (measure of how noisy is an image) to improve the SNR of the detection image and allow better detection of dim sources.

Detection and association mode There exists an alternative mode to create the list of sources and the groups for model fitting and measurements. Instead of using a detection image (detection mode), SourceXtractor++ proposes an association mode, in which the user gives a catalog of sources and groups to the software. This can be especially useful if a catalog of sources in the field already exists, for example with the ones published on the DJA. It allows to expand this catalog with a full coverage and without mis-detection of sources (which can happen in very noisy region such as the edges of frames).

I first tried to use the association mode. To do this, I used the DJA catalog as the initial source catalog. Before giving if to SourceXtractor++, it is necessary to build the groups that will be used for the simultaneous model fitting of nearby sources. For this, the idea was to draw ellipses on a dark frame that represents each source in the catalog, and whose major and minor axes are determined by the size parameters in the catalog[7]. Then, the `photutils` Python package is used to create a segmentation map from this frame. This has the effect of grouping overlapping ellipses, therefore grouping the sources they represent because their individual flux overlap on some pixels. An example of a resulting segmentation map is shown on figure 15.

However, this association mode proved to not work as wanted in SourceXtractor++. It's important to note that SourceXtractor++ (and any model fitting software on such

[5]Flux contained in circle or ellispe of a given size

[6]Area above a certain flux value

[7]The catalogs on the DJA are generated using SExtractor. SExtractor fits an ellipse to each source. This drawing step simply draws these ellipses, expanded to enclose more of the source flux, on a dark empty frame.

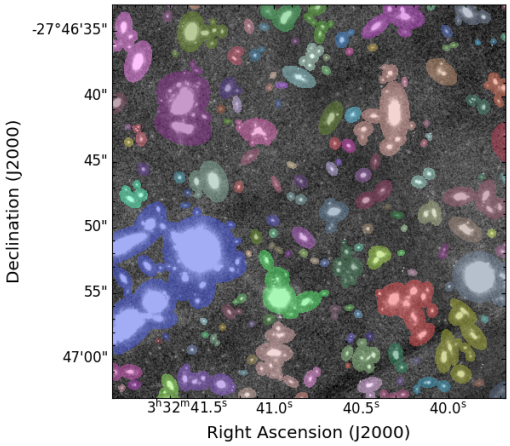


Figure 15: Example of a segmentation map created from a source catalog from the DJA. It is overlayed on a real JWST image from the same field to show how each ellipse corresponds to a real source. The colors of the ellipses show their group (overlapping ellipses share the same color). This frame is a cutout from the GOODS-S field.

large images containing 10-100k sources) is notoriously slow to run. Depending on the number of sources and the complexity of the model used, it can take from a few hours to a few days to run. Some details are given in 3.4.1.

The crucial point is that SourceXtractor++ runtime is governed by the largest group of sources it has to fit. Usually, we tried to keep the number of sources per group under 30-50. However, the way we performed the segmentation and grouping to use the association mode led to very big groups, sometimes reaching 200 sources. This is due to a chaining effet. Because groups are generated by looking at overlapping ellipses, a chain of overllaping ellipses will give a single group. Even though the two most extreme sources in this chain share absolutely no pixels and don't have any influence on each other. A way to avoid this would be downsize the ellipses, but this comes with the risk of missing some sources that should be grouped together.

In the end, we decided to use the detection mode only and let SourceXtractor++ group the sources. The merging with the DJA catalog is therefore done as a post-processing step, by matching the SourceXtractor++ catalog to the DJA one on the sky coordinates (right ascension and declination) using the `astropy` Python package [Astropy Collaboration et al., 2022] and the `match_to_catalog_sky` function.

How to run SourceXtractor++ SourceXtractor++ is not an executable software, with its own window and tabs. It's a program that is started in command-line and runs in a shell terminal. It is configured via a text file to give values to its many arguments (which can also be given directly in the command line), and with a Python script where the user defines the path to the different frames to use (detection and measurement frames, weight maps and PSFs) and the model required for model fitting. Although it is very customizable, it is also very much not user-friendly. Therefore, one of my first task was to simplify its use as much as possible. I detail this more in 5, but in short, I created a Python package and shared all my code on GitHub for anyone to use.

With these configuration files, it is possible to run SourceXtractor++, which, after a long run of many hours or days, produces a catalog (usually in FITS format, a standard file format used for images and catalogs in astronomy [Pence et al., 2010]). In this

catalog, each row corresponds to one source. The different columns contain astrometry (position in the sky), simple morphology (best fit ellipse, radius...), photometry (isophotal measurements, aperture photometry...) and model fitting (model parameters, modeled flux and magnitudes...) values.

3.4 Amazon Web Services

As explained in 3.3, SourceXtractor++ takes a very long time to run. However, it is multi-threaded and can take advantage of multiple CPUs to accelerate its processing. One of my task was therefore to implement SourceXtractor++ on AWS EC2 (Amazon Web Services Elastic Compute Cloud), the cloud computing service of Amazon. AWS is a very vast collection of web services which the two most used are EC2 for cloud computing and S3 (Simple Storage Service) for cloud storage. These services allow anyone and any organization to easily buy cloud services instead of purchasing expensive infrastructures such as servers.

Using AWS for my work made even more sense because it was already used at DAWN, especially for hosting the DJA and the heavy images on it (most images are between 100MB and a few GB). AWS EC2 was also already used for some pipelines for the DJA, or for MOSFIRE, a near-infrared spectrograph at the Keck observatory.

AWS EC2 works on a system of instances. It's possible to launch an instance, a virtual machine, and to connect to it via SSH. From there, it's possible to install any program and run any code in a terminal. I found this quite tedious, especially since I worked mostly with Python and Jupyter notebooks. I wanted to be able to use EC2 as a transparent virtual machine, meaning that on the user side it would not change the way one would code. I therefore developed simple scripts to automatically start a Jupyter server on an EC2 instance and be able to connect to it via VS Code (a versatile and very polyvalent coding environment). That way, it was possible to work with Jupyter notebooks as one would on its local machine, while profiting of the higher computing power of EC2 instances. This also allowed to launch long calculations in Jupyter notebooks, and keep them running even when the local computer was shutdown (very useful for SourceXtractor++). I documented and shared these scripts on GitHub for anyone to use (and some researchers at DAWN have started using them).

3.4.1 Benchmarking

SourceXtractor++ is a multi-threaded software and AWS EC2 allows the use of instances with virtually any number of CPUs wanted. Therefore, one could think that it is possible to run SourceXtractor++ as fast as possible by simply having a very big number of CPUs. My tests show that it's absolutely not the case. As quickly explained in 3.3.2, the runtime of SourceXtractor++ is governed by the size of the largest group of sources. In fact, SourceXtractor++ can only use one CPU core to perform model fitting on one group of sources. Therefore, for large groups (>50 sources), this process alone, on a single core, can take multiple hours. Using a computer with many CPU cores can very quickly accelerate the beginning of the computation for the smallest groups, but the biggest groups always limit the minimal runtime. Even worse, when only the biggest groups of sources remain, the computer uses sub-optimally the available resources: the CPU use is far from 100%, meaning that resources are wasted. Since EC2 instances' hourly cost is based on the available resources (the more CPU an instance has, the more expensive it is to use

per hour), it is important to find the right balance between total computation speed and total computation cost.

In order to find this balance, and better understand how the multi-threading of SourceXtractor++ works, I decided to run a benchmark of SourceXtractor++ on AWS EC2. The idea, was to run SourceXtractor++ with different settings on images of different sizes, on EC2 instances of different powers. Because of the long time of single SourceXtractor++ runs, I decided to only perform one run with a set of these parameters, therefore leading to potentially big uncertainties on the results. The goal here is not to find the absolute optimal parameters, but rather some guidelines to choose them.

The parameters used for this benchmark are:

- SourceXtractor++
  - `thread_count`: Number of threads[8] that SourceXtractor++ uses.
- AWS EC2 instance
  - Instance type and vCPU: The instance type defines the number of CPUs (formally known as vCPUs for virtual CPUs because AWS EC2 instances are virtual machines) and their generation and type. AWS EC2 instance type names are in the format *c6a.4xlarge* where *c* is the general focus type of the instance (focused on memory *m*, computation *c*, general *g*, etc), *6* is the generation of the instance, *a* is the type of CPUs used (*a* for AMD, *i* for Intel, etc), *4xlarge* is the size of the instance, directly linked to the number of CPUs (*4xlarge* has 8 CPUs, *8xlarge* has 16, etc).
  - Hyperthreading: EC2 instances allow by default two threads to run on a single CPU core, in effect, doubling the number of available CPUs. It can sometimes be faster to disable this option to avoid losing computation time on switching between threads.
- Images
  - Number of bands: Number of color channels used.
  - Size: Size of the images used, given in solid angle (arcmin^2).

To evaluate the different parameters choices, I looked at the runtime of SourceXtractor++ divided by the number of sources in the frames and the number of bands. It is expected that the runtime would be linear with the number of sources to fit, as well as with the number of bands as model is fitted for each source, in each band. Therefore, to really compare different runs with different number of sources in the frame and different number of bands, it is necessary to divide the runtime by these values.

The result of this - very incomplete by lack of time - benchmark is given in figure 16. Some conclusions are given here:

- `thread_count` seems to have an optimal value. By being too small, it is the limiting factor of the computation speed by not allowing SourceXtractor++ to use the full resources of the instance. By being too large, it slows down the computation by losing time in switching between the myriad of threads. A rule of thumb is to set `thread_count` to 2-4 times the number of available CPUs.

[8] A thread is a piece of a process (set of instructions). Usually, only one thread can be run at the same time on a CPU core, but the core can switch between different threads.

- Running SourceXtractor++ with multiple bands at the same time seems to be slightly faster than just linear time. This can be explained by the fact that SourceXtractor++ shares some values between the model fitting in different bands, such as astrometry (position of the source center).[9]
- Disabling hyperthreading doesn't seem to improve or deteriorate the runtime.
- Bigger images seem to slightly improve the runtime.
- Bigger instances with more CPUs improve the runtime, but not linearly with the number of CPUs (twice as many CPUs doesn't mean twice as fast), because of the issue of a single group of sources running on a single core.

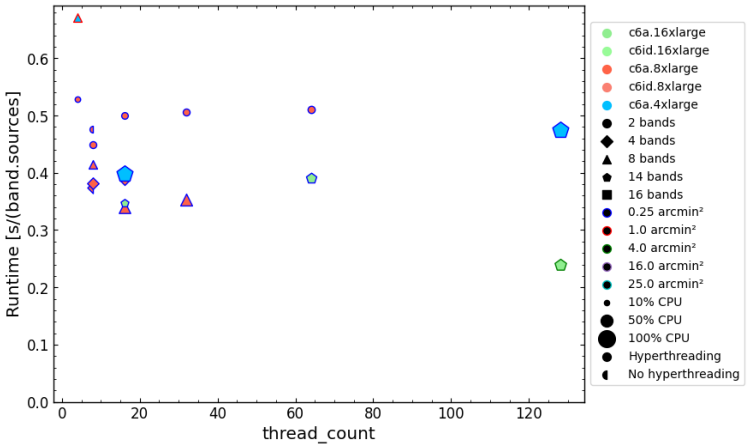


Figure 16: Benchmark of SourceXtractor++ on AWS EC2.

This benchmark led me to the conclusion that it would be very complex to find optimal parameters, and even good guidelines, especially because the runtime of SourceXtractor++ seems very dependant on too many parameters and on the images themselves.

A point not yet adressed is the memory used by SourceXtractor++. Of course, with bigger images, more RAM memory is needed to store the images and the results during calculations. This means that to run SourceXtractor++ on big images, one needs a computer with a large memory. AWS EC2 instances scale in number of CPUs as well as RAM. However, since they also scale in cost, and because of the "tail" issue of the runtime being limited by the largest group of sources, I decided to use medium size instances (*4xlarge* or *8xlarge*) on small cutouts of the complete images. This means that the full images need to be tiled, and each tile is run on one EC2 instance. This also allows to run a full field faster with SourceXtractor++ because it enables parallelization of the computation.

3.4.2 Tiling

Because of memory issues, it is not realist to run SourceXtractor++ on a single 30,000 x 30,000 pixels image, containing ∼100,000 sources. More over, this would take a very long

[9]N.B. It is very important to not run SourceXtractor++ with an empty frame (this can happen when using a cutout or a tile from a bigger frame) as SourceXtractor++ will not run model fitting in this case. This happened to me during the benchmark and led me to initially believe that 16 bands was much faster than 2,4,8 bands before realizing than no model fitting had been done.

time. In order to make this process faster, and also more stable to failures[10], I decided to tile the images: instead of running the program on a single large instance with a single large frame, I run it on multiple instances, each working on a smaller part of the large image.

Doing this tiling allow for parallelization by running all the tiles at once, but also better optimization of resources (CPU and memory) and therefore cost. It is possible to choose smaller instances for some tiles if they don't require as much memory or computational power as others. From experience, some tiles work just fine with a *c6a.4xlarge* instance whereas some crash and require a bigger *c6a.8xlarge*.

To define this tiling, I decided to define an angular size of tiles and a size of overlap. It is indeed crucial to have an overlap between the different tiles in order to recombine them into a single catalog at the end of the process. My program therefore calculates the number of tiles required to cover the whole image by taking these sizes into account. The tiling is summarized on figure 17.

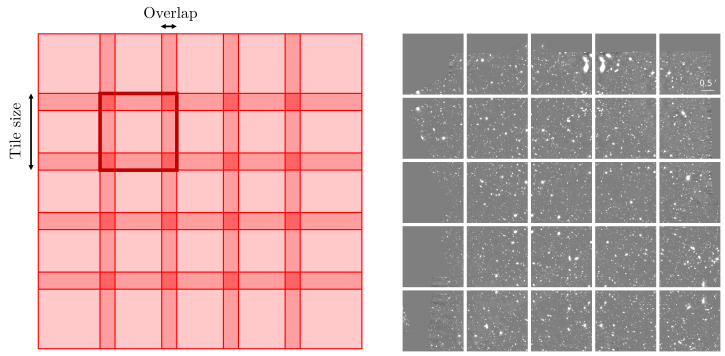


Figure 17: Tiling of images for SourceXtractor++ parallelization. (a) shows the definition of tile size and overlap. (b) is an example of this process on the GOODS-S field with 2′x2′ tiles and 0.5′ overlap.

With this tiling, comes the question of recombination of the results into a single catalog. This is further separated in two problems: catalogs and images. SourceXtractor++ produces a model and a residual images as well as a catalog. The model image is the modelization of the frame according to model fitting of each source. One model image is created for each filter given to SourceXtractor++. The residual image is simply the difference between the base (data) image and the model image. This is useful to see how well the fit has been performed. These images also need to be recombined after the tiling process in order to create unique model and residual images for the whole field.

Catalog In order to merge the catalogs from all the tiles, I make use of the function `match_to_catalog_sky` from `astropy`. This function matches two lists (A and B) of source positions (in my case, right ascension and declination) by associating every source in catalog A to a source in catalog B. It also calculates the angular distance between the two matched sources. This angular distance is used to remove false matches. This is

[10]For reasons I didn't manage to explain, SourceXtractor++ can sometimes get stuck or even crash in the middle of a run. This could be because of a lack of memory, or a random issue. Running a very large tile for tens of hours would mean taking the risk of having it crash in the middle of it and losing everything.

simply done by disqualifying any matches with a distance above a certain threshold - in my case, 0.3″. This treshold has been chosen using the histogram of the angular distances for the matches (figure 18), where there exists a clear separation between real and false matches. The threshold chosen here corresponds to 5 (resp. 10) pixels in the LW (resp. SW) channel and can be associated to differences in model fitting between different tiles or between SExtractor and SourceXtractor++.

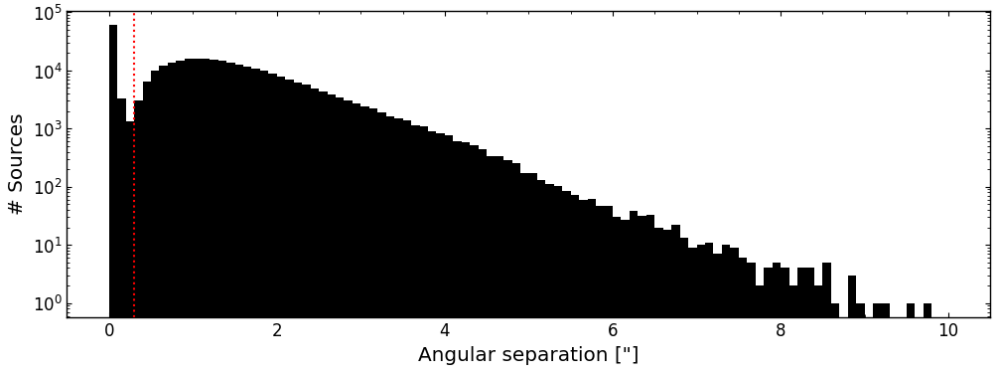


Figure 18: Histogram of the angular distances between matched sources using `match_to_catalog_sky`. The data here is the match between the original DJA catalog for the CEERS field, and the catalog I created using SourceXtractor++. Histrograms have similar shapes between tiles.

To merge catalogs from different tiles, the idea is to do it sequentially, tile after tile. To add a new tile catalog to the full catalog, I match them using `match_to_catalog_sky`. Using the distance threshold, I classify sources between matched (corresponding to the ones in the overlap region between the tiles) and un-matched. In the new catalog, I keep all the un-matched sources for completeness. For the matched sources (same source identified in the two catalogs), I compare their errors on magnitude (F200W band here), computed by SourceXtractor++. I keep only the lowest error one since it is associated with less uncertainty on its measurements. By doing this process, we ensure the completeness of the full catalog compared to the individual ones, while keeping only the best version of the sources in the overlapping regions. An additional post-treatment is performed in 4.1 to remove false detections.

Model images To merge the model images, the idea is to co-addition them. This means aligning them based on their coordinates (given by the data in their header, as a WCS, *World Coordinate System*[11]) and calculating the mean of them on each pixels to create a final image.

The combination step using the mean is needed for regions with overlap were multiple pixel values are available (compared to only one value for pixels covered by only one tile). In theory, reprojection on a different pixel grid is also needed because the pixel grids of all the different tiles may not align exactly. This is because we use sky coordinates to create and recombine the tiles instead of pixel coordinates. There is therefore no guarantee that the angular size translates to an integer number of pixels, or that the sky coordinate of the center falls exactly at the center of one pixel. The reprojection takes care of these inaccuracies by defining a new pixel grid covering all the tiles to combine and mapping their pixel values to it. For speed (because of the high number of pixels involved), I decided

[11]Standard associated to the FITS format to store coordinate and distorsion data with an image: `https://fits.gsfc.nasa.gov/fits_wcs.html`

to use a simple interpolation, but more advanced method such as *drizzling* exists [Fruchter and Hook, 2002]. All these steps are performed directly by the `reproject_and_coadd` function from the `reproject` package in Python [Christoph Deil, 2024].

This function can take as arguments a target WCS and shape of the pixel grid. To be compatible with the images on the DJA, I decided to use the WCS and shape from the images on the DJA. That way, the mosaic created from the model tiled images fits exactly (same pixel scale and orientation) with the data images.

I used `reproject_and_coadd` for model images only and not the residual images. I decided to calculate the residual images myself from the base data images from the DJA and the mosaiced data images. This ensures that the residual images are "true" residuals, especially in the overlapping regions where `reproject_and_coadd` may modify slightly the values to fit the tiles. Using `reproject_and_coadd` with the residual images might create differences in these regions and therefore lead to a misleading residual image, not representing truthfuly the difference between the full data image and the full model image.

3.4.3 Automation

The goal of my project was to make it as simple and integrated as possible to run SourceXtractor++ on the different fields available on the DJA. Because of the tiling step, I needed to automate the process. The full process of morphological measurements with SourceXtractor++ I came up with is the following:

1. Main instance (*m5d.4xlarge* for memory)
   1. Download all the frames of the field from the DJA
   2. Pre-process them (decompress and save to a S3 bucket)
   3. Find point-like sources, estimate the PSFs in all the bands (see 3.2) and save them to a S3 bucket
   4. Tile the images (see 3.4.2) and save them to a S3 bucket
2. For every tile
   1. Start an instance (by default, *c6a.4xlarge*)
   2. Download the necessary files (frames, PSFs, configuration files, code)
   3. Run SourceXtractor++ with the selected model (Sérsic or Bulge+Disk, see 3.3.1)
   4. Save the resulting catalog and images to a S3 bucket
3. Main instance (*m5d.4xlarge* for memory)
   1. Merge tile catalogs and images (see 3.4.2)
   2. Save the full catalog and images to a S3 bucket
4. Main instance (*m5d.4xlarge* for memory)
   1. Merge the full SourceXtractor++ catalogs with the two models (Sérsic and Bulge+Disk) to the DJA catalogs (aperture photometry, and results from SED fitting[12])
   2. Save the complete catalog to the DJA

Each of the sub-steps in this process is coded as a bash and/or Python script. I also developed a package (`dja_sepp`) to help with many of the different operations involved (see 5.2).

The main steps are coded as bash scripts. This is practical because it is possible to launch an AWS EC2 instance using a command line (and therefore it is possible to do

[12]The spectral energy distribution fitting is performed by EAZY [Brammer et al., 2008] and gives data such as redshift, mass, luminosity, flux in different filter bands...

so in a bash script). Even more, it is possible to give a script (called "user data") that is executed at start-up in the AWS EC2 instance. Therefore, in one script, the user can run one of the main steps from the process by sequentially launching an EC2 instance and running a script inside it. This is especially useful to run SourceXtractor++ on all of the tiles: instead of starting the instances one by one, a single `for` loop can launch the process on all the runs at the same time.

Running code with "user data" is however tricky. Indeed, the code is run as `root` (user with administrator privileges) on the instance. This means that it will run code from the root of the computer, and not as a user. This can lead to the code not finding the correct version of Python or other softwares or packages. In my case, I set up the instance manually by installing Python, SExtractor, PSFEx and SourceXtractor++. Then, I could create an image of this instance, enabling anyone (or a script) to launch an instance with these tools already installed. However, these were installed as a user, and not as `root`. Therefore, in the "user data" script, I add to give the path to these softwares for it to run correctly[13]

To make it possible to debug or follow the progress of SourceXtractor++ on the EC2 instances, I used the `screen` command. This allows to create a background terminal which can be detached or reattached to see it or not. Detaching a screen has the benefit of not stopping the code currently executing. Therefore, one can connect to the instance via SSH and reattach the screen to follow the progress of SourceXtractor++, or find what happened wrong if there was an issue.

About issues, I encountered two. First, for some reason that remains a mystery, SourceXtractor++ can get stuck during the detection/segmentation step. This manifests by the progress bar no longer progressing before it reached 100%. I isolated the issue to an error with `LAPACK` and the `dlevmar_pseudoinverse` function, used by SourceXtractor++. This is very similar to a knwon issue which doesn't have a solution. Sometimes, restarting SourceXtractor++ solves it, but fortunately, it happens only to few tiles (during my project, it happened with 3 tiles on $\sim$100). Another issue that can happen is SourceXtractor++ crashing during the measurement step. This is generally caused by a lack of memory. It can easily be solved by running it with a bigger instance. This issue can be spotted retrospectively because the residual image will not have been generated by SourceXtractor++.

[13]This is performed by adding `sudo -u ec2-user env "PATH=$PATH"` before the commands run in "user data" where `$PATH` is the `PATH` environment variable from the user environment in the instance (obtained by `echo PATH`).

4 Results

After having detailed the technical aspect of the project I worked on, this section delves into the measurements and scientific analysis I realized. Thnaks to my work with tiling and automation, I was able to run SourceXtractor++ with Sérsic and Bulge+Disk models on the following fields: CEERS, GOODS-S, GOODS-N, PRIMER UDS, PRIMER-COSMOS. It is also relatively easy to run it on other fields available (now or in the future) on the DJA.

4.1 Validation

A first curcial step before doing any scientific analysis was to validate the results obtained with SourceXtractor++. To do this, the idea is to compare some values calculated by SourceXtractor++ to the ones already present in the DJA catalog. This was limited to photometry but I also compared my morphology results with the literature.

Magnitudes An easy validation, is to compare the magnitudes fitted with SourceXtractor++ models to the ones measured by aperture photometry using SExtractor and available in the DJA catalogs. To match sources from SourceXtractor++ to the DJA catalog, the same methods as for the merging of tiles has been used. I used the `match_to_catalog_sky` function with a threshold distance of 0.3″ to consider a match between two sources. This comparison can be seen on figure 19 on the CEERS field and with the Sérsic model.

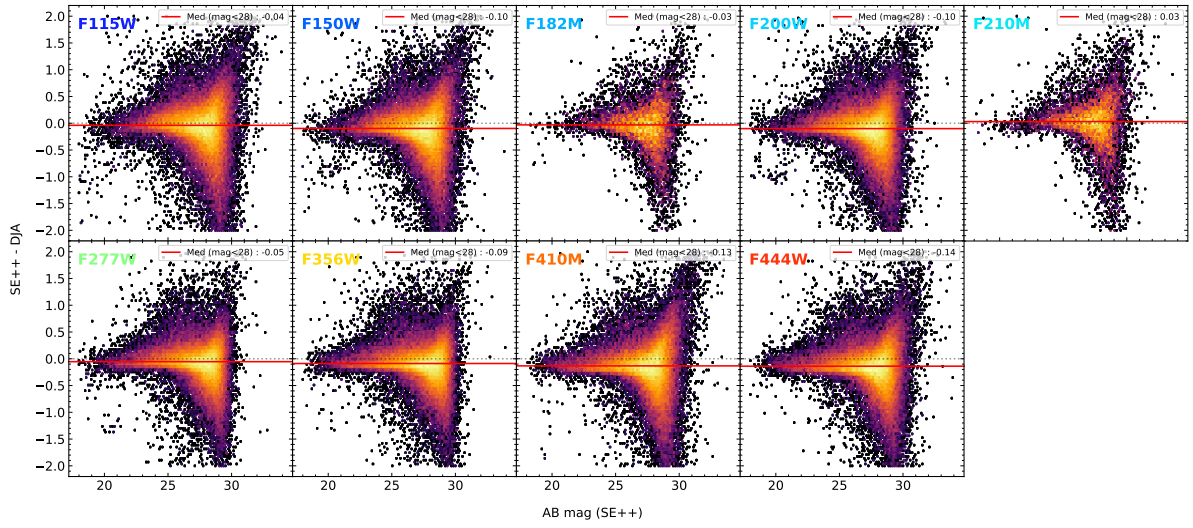


Figure 19: Magnitude comparison between the DJA catalog and the SourceXtractor++ Sérsic model fitting from this work. The color show the number of sources in each bin, with a logarithmic scale. The comparison is performed on all the available bands in the CEERS field. The red lines show the median difference for sources with mag<28.

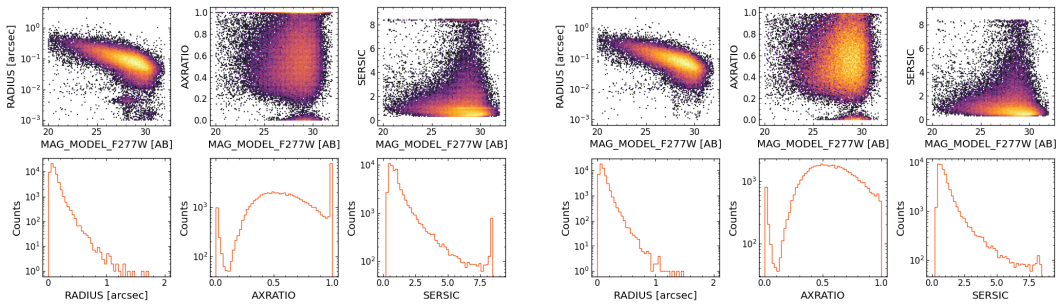
These plots were used a lot during the development of my programs and to set some parameters of SourceXtractor++. Some artifacts appeared in these comparisons which were resulting from non-fitted sources or bad modeling. After fixing these issues, one can see there is a great agreement between the magntiudes from the DJA and from SourceXtractor++. The median deviation for sources brigther than mag 28 is always smaller than 0.14 mag. The spreading of the distribution reaches ± 2 mag for the dimmest sources, which is in the order of the uncertainty given in the DJA catalog for them. Because

they are so dim, their SNR is low and one can see they are at the limit of detection. Overall, these plots show that the measurements performed by SourceXtractor++ agree with the ones present in the DJA catalogs. By taking the Bulge+Disk model, one gets similar results and agreement.

Morphology The objective of using SourceXtractor++ is to perform morphology measurements or galaxies. It is therefore also important to check and validate these values. For this, I studied the radius (for Sérsic, disks and bulges) of the detected sources, as well as their Sérsic index and their aspect ratio (represented by the `AXRATIO` value).

These morphology plots revealed some artifacts - both for the Sérsic and the Bulge+Disk models. They are symptoms of bad fitting (activation of the constraints for some parameters) or failure to fit. For better completeness (measured in 4.2), I did not remove them from the final catalogs but only associated them a flag value to make it easy to filter them, but also make it possible to keep them if wanted. The filtering is simply performed by thresholding: for constraints activations, I flagged the sources with values at constraint (with a small tolerance), and for fitting failures, I flagged the sources with values that stayed at their initial value (with a tolerance around). The effect of this selection is observed on figure 20.

Once this filtering has been performed, the resulting distributions match the literature.



(a) All sources (b) Clean sources

Figure 20: Morphology plots for the GOODS-S field modeled using Sérsic profiles. The color shows the number of sources in each bin using a logarithmic scale. (a) shows all the sources detected in the field (62844 sources), (b) shows the clean sources after removing bad model fittings (52519 sources). One can see on (a) horizontal lines resulting from these bad fittings.

4.2 Performance

As presented in 3.4.1, a part of my work has been to find how to optimize the performances of my programs. I measured this by looking at the CPU usage to ensure it reaches 100% for as long as possible and that the time per source is minimal when running SourceXtractor++. Another important aspect is to look at the completeness of the measurements performed. Indeed, SourceXtractor++ doesn't detect all the sources found in the DJA catalog, and fails to fit some of them.

CPU usage To look at the CPU usage and calculate the runtime, I exported the CPU usage data from AWS EC2. That way, for each tile, I have the CPU usage as a function of time, which can then be normalized by the number of sources found in the corresponding

tile. I did this study for the CEERS field, but the results are identical in other fields. The results are shown on figure 21.

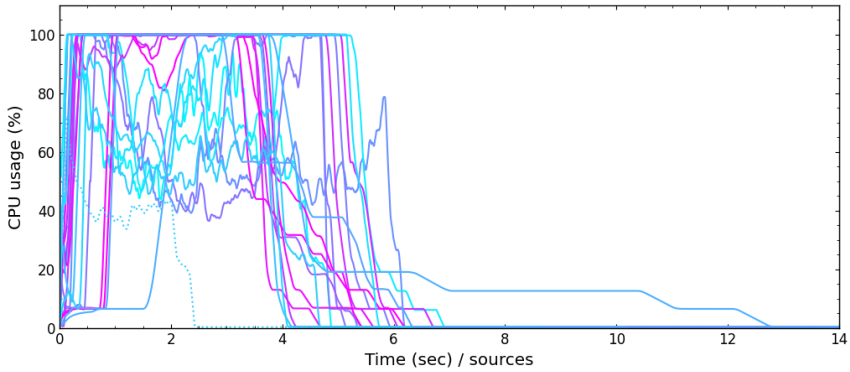


Figure 21: CPU usage normalized by number of sources for the 25 tiles used for SourceXtractor++ on the CEERS field, with a Bulge+Disk model (except the dash curve with the Sérsic model). The color gradient is the id of the tiles.

These curves reveal different effects, some explained, others not. First, it is interesting to see the consistency of the runtime of SourceXtractor++ for one source. No matter the tile, it takes around 4-6s per source with the Bulge+Disk model. Unsurprisingly, the Sérsic model is almost two times faster, at around 2-3s (with only one curve here, but by experience, it's the case on a larger scale).

In 3.4.1, the trailing effect is presented. This effect is clearly shown here with some curves staying below 10% for a long time (many hours for some) at the end of the run. This happens because only one large group of sources remain to fit and SourceXtractor++ runs it on only one CPU core.

One can also see that some tiles take a very long time to start: they remain below 10% before jumping to 100%. This effect, not really understood, is the collection step taking a very long time before transitioning to the measurement step. Because this step only runs on one CPU core, it keeps the total CPU usage low.

Finally, some tiles have been run on a *c6a.8xlarge* instance instead of a *c6a.4xlarge* because of a lack of memory. Although the *c6a.8xlarge* instance has twice as many CPU cores as the *c6a.4xlarge*, the CPU usage stays mostly around 40-50% (it peaks at 100% at the beginning but quickly goes down). This is not really understood, but might be an issue of SourceXtractor++ not being able to use all the available CPU cores available, for some mysterious reason.

Runtime Using the same data used for figure 21, it is possible to calculate the total runtime necessary for a field. This calculation will not be precise because I didn't save the runtime for all the tiles, especially because of the issues with instance types, and also because it is very dependant on the images.

On the CEERS field, with the Bulge+Disk model, I estimate to around 530±50h the total computation time (distributed on 25 instances). Because this field has 76637 sources according to the DJA catalog (see table 1), it gives a time of 25±3s/source for the Bulge+Disk model (the difference with the time presented before is due to the filtering of false sources presented on figure 18).

From experience, around 15% of the tiles require a *c6a.8xlarge* (16 vCPUs) instance rather than a *c6a.4xlarge* (8vCPUs). This gives a mean number of CPUs of 9.2. Finally, it can be calculated that the Bulge+Disk model takes 2.7±0.3s/source/CPU to

run. The Sérsic model runs approximately twice as less time, giving a run speed of 1.3±0.2s/source/CPU.

These values can be used to estimate roughly the runtime of SourceXtractor++ on a new field. However, since my program uses tiling, this total runtime is actually folded on multiple parallel instances. The number of sources to consider is therefore the number of sources in one tile. From my experience, 2′x2′ tiles take from 3 to 8 hours with the Sérsic model, and from 10 to 40h with the Bulge+Disk model.

Completeness As explained in 3.4.2, not every sources in the images are detected or fitted successfuly by SourceXtractor++. This creates a difference with the DJA catalog, taken as a reference here. This can be explained by multiple factors: different settings and algorithms for source detection, source profile too different than the models, issues with the algorithm, false detection in the DJA catalog...

It is therefore crucial to quantify the completeness of the measurements I performed with SourceXtractor++ compared to the DJA catalog, i.e. how many sources from the DJA have morphology measurements from SourceXtractor++. Because I used two models, this completeness is measured on the models separately, but also jointly to see how many sources have measurements with Sérsic and Bulge+Disk. The completeness is presented in table 1.

This completeness measure takes into account multiple selections:

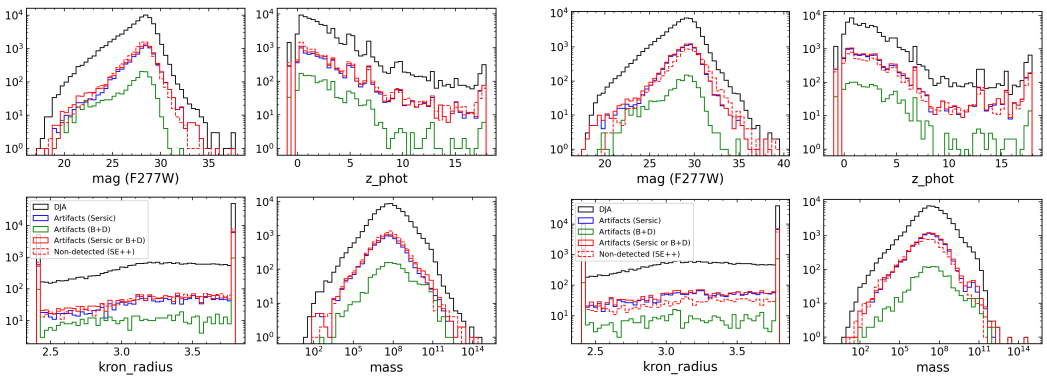
- Sources not detected by SourceXtractor++ and therefore not matched to the DJA catalog;
- Sources badly fitted by SourceXtractor++ resulting in the artifcats visible in figure 20;
- Quality selection by removing sources with SNR<3 or a magnitude above the limiting magnitude (5σ from the magnitude of the background) of the corresponding field, taken from Weibel et al. [2024]. This selection is also performed on the raw DJA catalogs.

Field	DJA	Sérsic	Bulge+Disk	Both
CEERS	67035	52604 (78.5%)	59046 (88.1%)	**51329 (76.6%)**
GOODS-S	57355	44931 (78.3%)	52754 (92.0%)	**44016 (76.7%)**
GOODS-N	65481	53291 (81.4%)	58852 (89.9%)	**51465 (78.6%)**
PRIMER-UDS (N)	68857	58947 (85.6%)	67134 (97.5%)	**57945 (84.2%)**
PRIMER-UDS (S)	65864	57397 (87.1%)	64537 (98.0%)	**56476 (85.7%)**
PRIMER-COSMOS (E)	50655	42359 (83.6%)	48496 (95.7%)	**41597 (82.1%)**
PRIMER-COSMOS (W)	51362	40493 (78.8%)	46964 (91.4%)	**39704 (77.3%)**
Total	426609	350022 (82.0%)	397783 (93.2%)	**342892 (80.4%)**

Table 1: Completeness table of the morphology measurements using SourceXtractor++ compared to the DJA catalog for different fields. Numbers show only sources with a F277W magnitude below the limiting magnitude of the field, and a SNR bigger than 3. For the SourceXtractor++ columns, they show the sources that are well fitted (see figure 20). The percentages are relative to the DJA catalog.

The completeness analysis shows that around 80% of the sources in the DJA catalog are matched to sources considered good from the SourceXtractor++ measurements, with morphology in Sérsic and Bulge+Disk models. It is noticeable that the filtering is mostly caused by the Sérsic modeling. It can be concluded that either the Sérsic model fits more poorly the sources in the images and therefore make the model fitting algorithm fail more often, or the selection done afterwards is too agressive (hence the choice to leave the sources considered as bad fits in the final catalog, so that anyone can do better filtering).

It is interesting to look at the distribution of some physical values for the non-detected / non-matched and bad sources. This could reveal some explanations as to why some sources fail to be measured by SourceXtractor++. For this, we decided to look at magnitude (in F277W), redshift (z_{phot} estimated by EAZY), radius[14] and mass (estimated by EAZY). These results are presented in figure 22



(a) CEERS (b) GOODS-S

Figure 22: Plots of major physical values to study completeness in different fields. The curves show the distribution of non-detected (dashed) or badly fitted (plain colored) sources.

It is interesting to see that the distribution for the complete catalog, the non-detected sources and the sources identified as badly fitted, are all the same. This shows that there is no immediate correlation between the type of source and the success of modelization by SourceXtractor++ and my program.

4.3 Astrophysical conclusions

The primary goal of my internship was to actively participate in the research around high-redshift galaxies by implementing tools to measure their morphology with SourceXtractor++. This technical work has also been expanded with a more scientific aspect by studying the results I obtained. This allows to further validate my measurements by checking that they give similar conclusions as the ones already known in literature.

4.3.1 Morphology and quiescent galaxies

One of the main research topics at the Cosmic DAWN Center is the study of quiescent galaxies. These are galaxies that no longer form any stars. Multiple reasons are brought forward to explain how a galaxy can "die". It could be because it simply used all of the

[14]Here, we use the Kron radius calculated by SExtractor and available in the DJA catalog. This radius is used for aperture photometry but is bounded and does not reflect very small or very large sources. This does not affect the results as it can be seen that its distribution is identical in shape for all the curves.

gas it had in stars. Another possibility is that it expelled its gas, leaving not enough to form new stars. This outflow could also be caused by nearby galaxies sucking the gas from the galaxy.

The most common way to find quiescent galaxies is to look at a so-called UVJ diagram, as presented by Patel et al. [2011]. This diagram is a member of the color-color diagrams. In astronomy, the color of a source is the difference of its magnitude measured in two different wavelength ranges (using two different filters). The color-color diagram therefore places a source in a 2D plot depending on two color measurements (usually from three magnitude measurements since the two colors generally share a filter in common).

To make measurements easier to compare, some standard filter sets exist. The historic one, and still used to give a lot of magnitudes and colors, is the UBVRIJHK Johnson-Cousins filter set, which can be seen on [Girardi et al., 2002, Figure 3]. To separate star-forming and quiescent galaxies, the U (ultraviolet, 320-400nm), V (visible, 500-600nm) and J (near-infrared, 1.1-1.4µm) filters are used.

Since the JWST isn't equipped with these filters, it is necessary to use SED fitting. The idea is to use the measured flux with multiple filters of JWST to fit a template SED (depending on the source type) and then calculate the integrated flux that would be seen of this SED using the wanted filters (here, U, V and J). This work is done by EAZY [Brammer et al., 2008] and is directly available on the DJA for the fields I studied.

Thanks to the modelization using the Sérsic and Bulge+Disk models, it is possible to study correlations between the morphology of galaxies and their position on the UVJ diagram (and therefore their classification between quiescent and star-forming). In figure 23, the distribution of Sérsic indices is displayed on the UVJ diagram for the CEERS field. Each bin shows the median index from all the galaxies falling in it. The distributions are showed for various redshift ranges and only for galaxies with a mass M such that $\log(M/M_\odot) > 10$ where $M_\odot$ is the mass of the Sun.

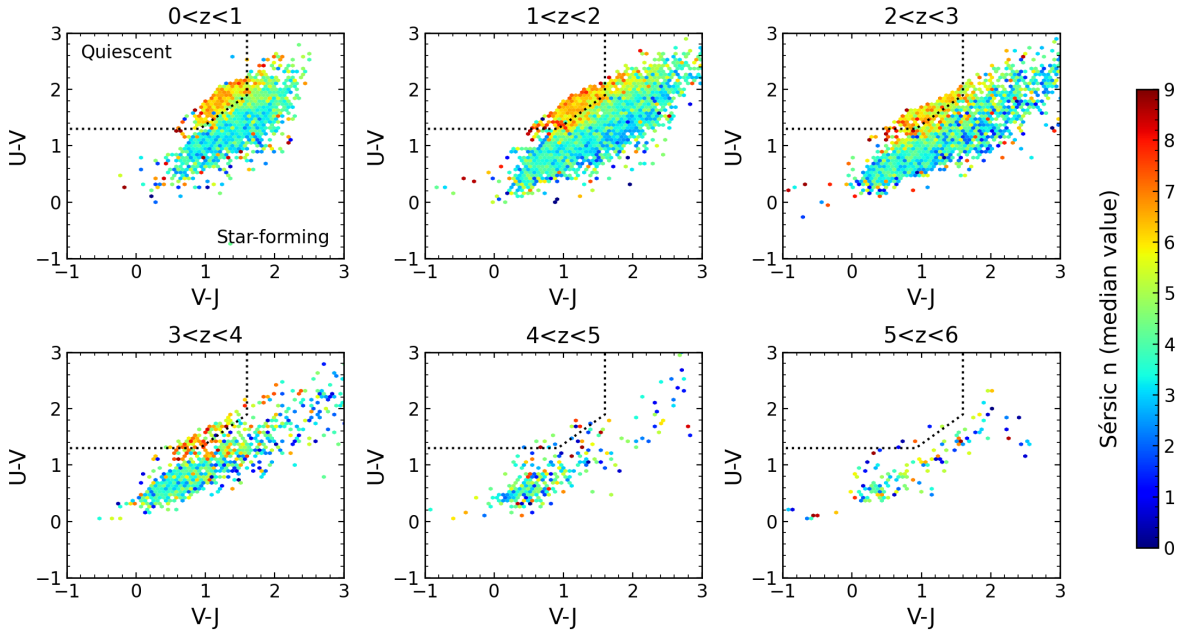


Figure 23: Distributions of Sérsic indices on UVJ diagrams for different redshift ranges. Galaxies are taken from all the fields covered in this work and such that $\log(M/M_\odot) > 10$. The color of each bin shows the median of the Sérsic index n of all galaxies falling in it, according to the color scale.

It can be seen on figure 23 that there is a gradient of Sérsic index orthogonal to the

border between quiescent and star-forming galaxies. It also shows that quiescent galaxies mostly have a Sérsic index n $>$ 5, indicating galaxies with a predominant bulbe. This is consistent with previous observations that quiescent galaxies are generally elliptical galaxies, following a de Vaucouleurs' profile (n=4). It is well known that disk galaxies produce new stars in their outskirts and in their arms, thus the bluer color there (symptom of young stars) and the redder color in the center (symptom of old stars). When a galaxy stops producing stars, these star-forming regions die down and the galaxy's Sérsic index increases.

An interesting point that shows here, although it's not a result from my work directly, is to see that there are almost no quiescent galaxies at redshift $z > 4$. This indicates either that galaxies before $z = 4$ didn't have time to use all their gas and die out, or that we don't detect these young quiescent galaxies (because of instrument limitations or treatment biases). Some observations tend to show that quiescent galaxies exist before $z = 4$, which contradicts simulations where galaxies take longer to form and die. This is not the only difference between current observations with the JWST and simulations. The biggest one in the high-redshift field are the Little Red Dots (LRDs), very bright and red objects at high redshift, that are way too massive to exist in the early universe according to our current knowledge [Matthee et al., 2024].

Another way to see the link between quiescent/star-forming and elliptical/disk galaxies is by classifying based on their morphology. For this classification, we used the Sérsic index n and the bulge-to-total mass ratio (B/T) calculated by the Bulge+Disk model in SourceXtractor++. In figure 24, we devised two classes:

- Bulge-dominated (elliptical) galaxies: n $>$ 1 and B/T $>$ 0.5;
- Disk-dominated (disk) galaxies: n $<$ 4 and B/T $<$ 0.4.

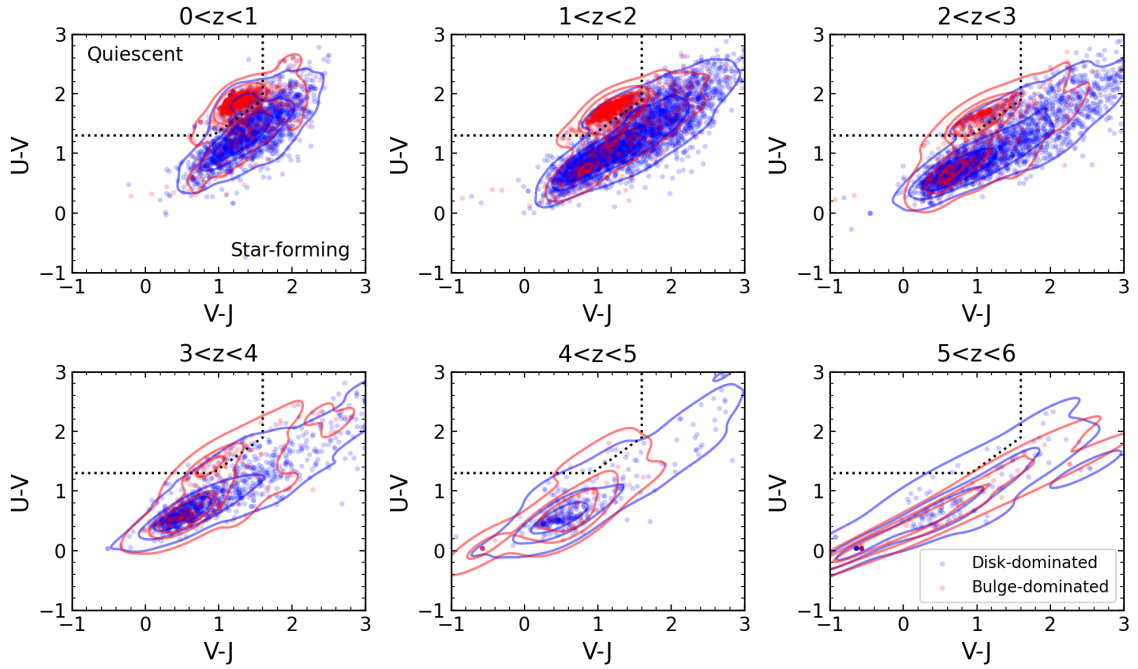


Figure 24: UVJ diagrams for different redshift ranges of galaxies classified as bulge- or disk-dominated based on Sérsic and B+D profiles. Galaxies are taken from all the fields covered in this work and such that $\log(M/M_{\odot}) > 10$.

The figure 24 shows the same conclusion as above. We see that quiescent galaxies are mostly bulge-dominated, elliptical, galaxies. This plot also shows how two independant methods can identify quiescent galaxies. The UVJ selection by Patel et al. [2011] uses photometric measurements. Because of redshift, these measurements must be obtained through SED fitting[15]. The other identification is through the morphology we calculated in this work.

Rather than using this binary classification between bulge- and dik-dominated, it is possible to use the bulge-to-total mass ratio (B/T) to have a continuous classification. Figure 25 shows the galaxies from all the fields used in this work on UVJ diagrams colored by their B/T value. They are also further separated in three redshift bins to show different epochs of galaxies, and in three B/T bins to better show effects of morphology.

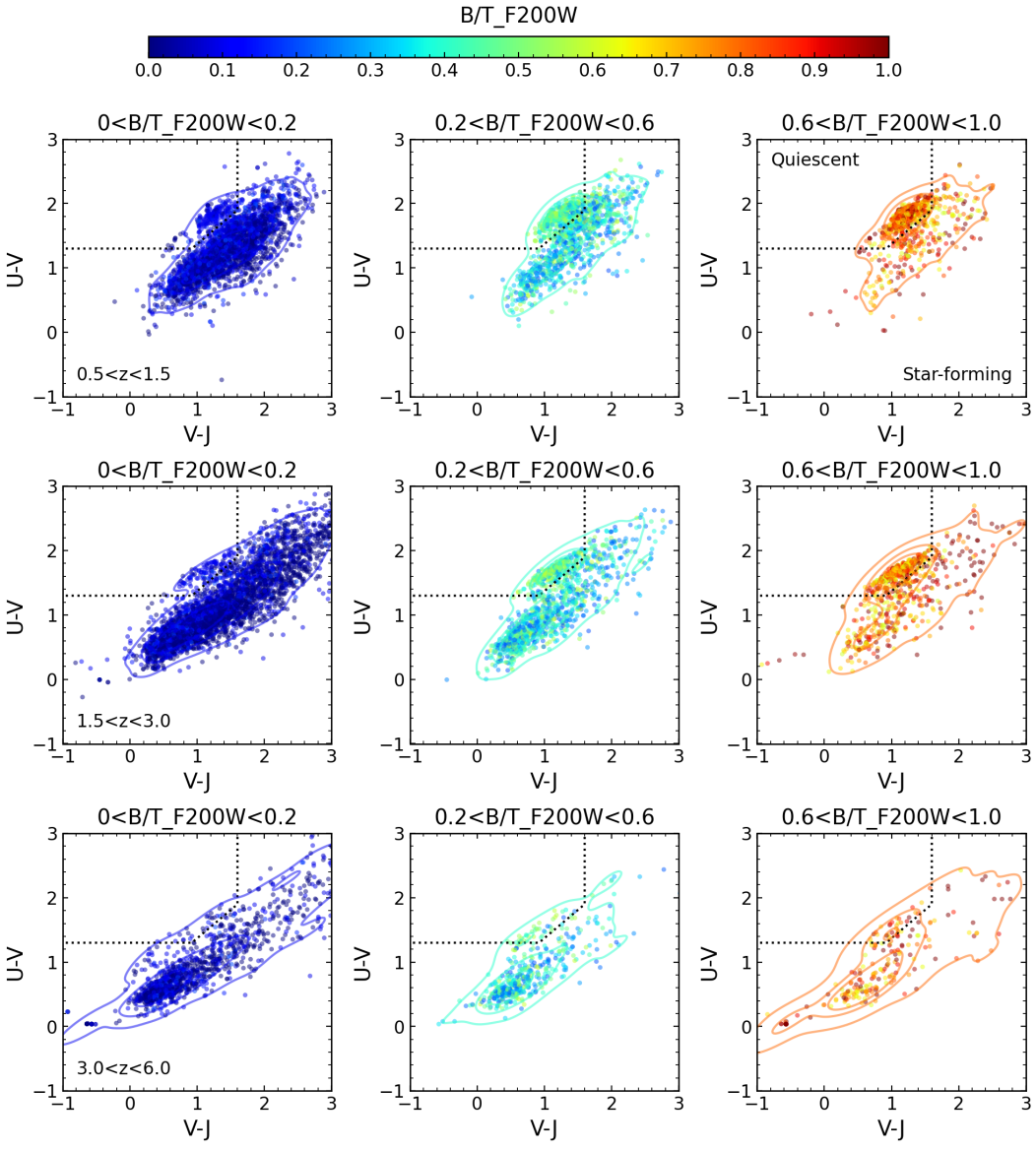


Figure 25: UVJ diagrams for different redshift and B/T ranges of galaxies. Galaxies are taken from all the fields covered in this work and such that $\log(M/M_{\odot}) > 10$. Their color shows their B/T value.

[15]The other option would be produce U, V and J filters for different redshift ranges, which is highly unpractical, and would still require SED fitting to know the redshift of the observed galaxies to take the correct set of filters...

We can better see in figure 25 that not all quiescent galaxies are bulge-dominated. Some are identified by the photometric selection as quiescent, but can have a B/T ratio smaller than 0.2. Furthermore, not all bulge-dominated galaxies are quiescent: some have a B/T ration higher than 0.6 but don't fall in the quiescent region.

This conclusion illustrates how galaxy evolution can be very different from as statistical point of view and from a ponctual point of view. Statistically, the different figures presented above show that quiescent galaxies are mostly bulge-dominated, and *vice versa*. However, this is not an absolute rule and some galaxies can no longer produce star, while still being disk-dominated. Although statistics is useful to understand the general galaxy evolution, some individual observations can challenge these models and lead to better, more complete, theories.

4.3.2 Size evolution through cosmic times

The previous message is especially true in the following paragraphs where we discuss the size evolution of galaxies through cosmic times. Because we have photometric redshifts for all the galaxies considered here, it is possible to study the size of these galaxies versus their redshift. This is very interesting to understand how galaxies may have evolved with time.

It is known that galaxies were more compact in the past (meaning they were smaller for the same mass) and their Sérsic index was generally smaller [Conselice, 2014]. However, no one knows precisely how galaxies have grown. In this work, thanks to the large number of galaxies fitted with Bulge+Disk model, I present plots that show the size evolution of the bulges and disks for $0.5 < z < 8$. Thanks to the Sérsic modeling I did, I was also able to compare my results with the literature and found compatible values, although my measurements of the effective radii were slightly smaller than the models presented by Ormerod et al. [2024].

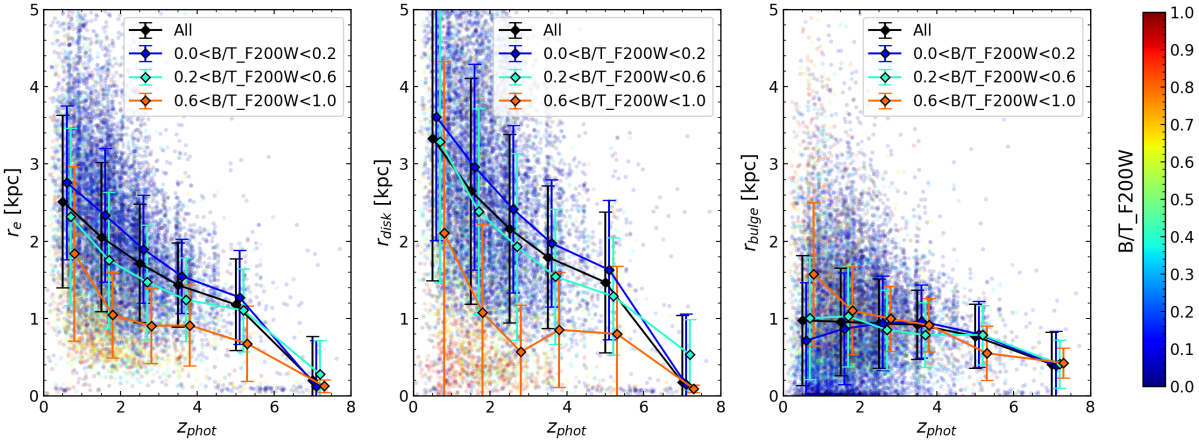


Figure 26: Size evolution of galaxies through cosmic times. These plots show the evolution of the effective radii, r_e (Sérsic model) and the disk and bulge radii, r_{disk} and r_{bulge} (B+D model). Galaxies are taken from all the fields covered in this work and such that $\log(M/M_\odot) > 10$. Their color shows their B/T value. The lines are calculated by taking the median value of the corresponding selected galaxies in redshift bins. The errorbars are the $\pm 1\sigma$ standard deviation in the redshift bin.

On figure 26, the sizes of galaxies (or their disk and bulge features) are given in comoving sizes. This means that they take the universe expansion into account. To calculate the comoving size, we use the following equation from Sahni and Starobinsky

[2000]. This equation is integrated in the `astropy` Python package, which we used through the `angular_diameter_distance` function.

$$D = \theta \frac{d_L(z)}{(1+z)^2}$$

where:

- D is the comoving size;
- θ is the angular size (measured on the images);
- z is the redshift;
- $d_L(z)$ is the luminosity distance, which depends on the redshift and the universe cosmological model.

The figure 26 shows how galaxies grow with time, which is consistent with the increase of galaxy compactness with an increase of redshift. From redshifts $z = 5$ to $z = 0.5$, the median effective radius of galaxies doubled. A similar conclusion comes from the evolution of the disk radii. However, it is interesting to see that bulges grow in the early universe, but stays relatively constant for redshifts $z < 5$. These conclusions should be taken with a grain of salt since it can be seen that the distributions of radii are quite large. The median size evolution doesn't automatically mean that individual galaxies grow in the same way.

I used the same B/T bins as on figure 25 to show the differences of size evolution between disk- and bulge-dominated galaxies. The effective radius diagram on figure 26 shows, to no surprise, that bulge-dominated (0.6<B/T<1.0) are smaller than the disk-dominated ones (0.0<B/T<0.2). This is a simple consequence of the effective radius which is the radius at which half the total light flux of the galaxy is emitted. For a bulge-dominated galaxy, this radius comes closer to the center than for a disk-dominated galaxy. It is however interesting to see that bulge-dominated galaxies seem to grow only from redshift $z = 3$, whereas disk-dominated galaxies grow continuously from $z = 8$.

By looking at the evolutions of r_{disk} and r_{bulge}, the differences between disk- and bulge-dominated galaxies emerge. For disk-dominated ones, we see that the bulges don't grow with time, and even seem to reduce in size from $z = 3$. Their effective radius growth seems therefore to be dominated by the growth of their disk.

For bulge-dominated galaxies, their disk don't appear to grow before $z = 3$. However, their bulge grow continuously by doubling in size from $z = 5$ to $z = 0.5$. Their effective radius growth is therefore really a combined effect of a disk and bulge growth. These galaxies could have grown from the bulge first, and this growth could have propagated to their disk.

It is important to note that galaxies can change of classification in their lifetime: a disk-dominated galaxy can become bulge-dominated, especially when it stops forming stars as explained earlier. The conclusions drawn before show the evolution of populations but not necessarily individual galaxies.

These results in the evolution of disks and bulges through cosmic times are unique and never seen with so many galaxies. My supervisor and I will look into publishing these results in a paper.

5 Outreach

In this section, I present how I shared my work to the researchers at DAWN, but also to anyone who can be interested in it. During my internship, I saw how the academic world is open (at least in astronomy), and how researchers can value humanity's scientific knowledge more than their individual carreer. Moreover, these two are not incompatible because research in astronomy always involve many researchers from all around the world, often grouped in big collaboration. I found this very motivating and much more exciting than keeping ones tools and data for themselves. Therefore, I wanted to participate in that, by publishing my work on the DJA, and by sharing my code on GitHub for instance.

5.1 DAWN Summit

Figure 27: Flash presentation of my work at the DAWN Summit.

Every year, DAWN organizes the DAWN Summit, three days where the researchers at DAWN present their work and on-going research. People attending are almost only other researchers from DAWN, but this is still very interesting, especially since the center is actually split on two campuses: one at KU (*Københavns Universitet*) and one at DTU (*Danmarks Tekniske Universitet*). This is the opportunity for researchers to learn more in-depth about what the others are doing, and to discuss about new projects to launch.

This year, a session was organized for interns and PhD students. This was a session of flash 5min presentation with 3min of questions. Because it happened during my first month of internship, I didn't have too much to show. I was still able to present my work on point-like sources detection (see 3.2.2) and the beginning of the PSF estimation. I also presented the work I was going to do during my stay at DAWN so that everybody could be aware of what I was working on.

At the very end of my internship, I also presented my work and especially the scientific results from 4.3 to all the researchers at DAWN. This allowed to show the morphology catalogs I created using SourceXtractor++ and they can now use. The plots I presented also showed preliminary results that can arise from this study.

5.2 `dja_sepp` Python package

The goal of my internship was to develop an easy-to-use tool that can be used with the DJA to run SourceXtractor++. To fully accomplish this objective, it is necessary to share my code in open-source. Many tools in astronomy, such as SExtractor, PSFEx, SourceXtractor++, EAZY, that I used more or less directly, are shared this way. This is the best way to reach the common goal of academic research: expanding humanity's scientific knowledge. By not sharing tools (or data), researchers have to make the same work twice, and lose precious time, or sometimes don't have the knowledge or capabilities to create similar tools.

It is very important to me to share my work and code for these reasons. I therefore published all my codes on GitHub, a web service used mainly by programmers because it allows to develop in a collaborative way and to deal with different versions or revert changes. However, its use has been expanding to many other fields because programming

is now used in any research, and also because GitHub can be used to easily publish data or projects online.

All my code is available on the DJA-SEpp repository[16]. I documented most of my code and notebooks to make it a bit easier for someone to use it in the future. I also wrote an introduction detailing the workflow to compute SourceXtractor++ modeling on DJA data and using an AWS EC2 instance. This documentation is far from perfect, but will hopefully simplify the work of future people.

This project has been published under the GNU GPL v3.0 licence. It is a permission licence that allows to use my code for any use (commercial or not), to modify it and even to use it in patents. However, it must credit the origin of the code, and be only published with the same licence and conditions. Furthermore, it doesn't come with any warranty. This is one of the most used licences to ensure sharing and use with no limitations, whilst also caring about improvements and keeping it open source and free-to-use.

To make my code even easier to use, I also published it as a Python package: `dja_sepp`. It is therefore available on PyPi, the standard collection of Python packages. This allows anyone to install all of my code using the classic `pip install` command. It was also very useful for my project because I used it abdondantly with AWS EC2 in order to install or update the code whenever a new instance is started to run SourceXtractor++.

Finally, I also wrote and publish a small tutorial to use jointly AWS EC2, VS Code (code interpreter) and Jupyter notebooks. It comes with scripts that help start a Jupyter server automatically on an AWS EC2 instance, and connect it to VS Code for seamless use. A step-by-step workflow is also given to install and use the scripts.

I sincerely hope my code will be used, entirely or even just some pieces. I strongly believe that academic research works best through cooperations and open and public data and tools. Keeping secrets does not make humanity understand the universe better.

[16] `https://github.com/AstroAure/DJA-SEpp`

6 Conclusion

During my 4.5 months at the Cosmic DAWN Center, I have embarked in a wonderful journey to further understand the very beginning of our universe. There I met researchers and PhD students from many different countries who helped me discover the world of academia. Thanks to them, I also learned a lot about astronomy and cosmology, about the evolution of galaxies during their lifetime, about the Epoch of Reionization and the cosmic dawn. I had the chance to work direclty with images taken by the incredible James Webb Space Telescope. This was for me an opportunity to discover how scientists work with data coming from scientific spacecrafts, valuable knowledge and experience for someone like me who wants to work as an engineer on the development of such space missions. I truly think that every space engineer should also experience the research world to better understand the needs and constraints of scientists and how to integrate them in the development of space missions.

Through my work at DAWN, I also participated in expanding catalogs with valuable morphology measurements. This data will be useful for future research to study the size and shape evolution of galaxies from the cosmic dawn to the present days. With my supervisor, we plotted diagrams showing first results of these evolutions. We find a bimodality at redshifts $z < 4$ between bulge-dominated quiescent galaxies and disk-dominated star-forming ones. We also show that bulge-dominated and disk-dominated population don't grow in the same way, with the first one seemingly growing from the bulge and disk at the same time, while the second one grows from the disk only. Further studies are enabled by this addition to the DAWN JWST Archive with morphological datas. Our first results will maybe be published as a paper in the coming months.

Acknowledgments

I want to thank a few people that made my stay in Copenhagen and at DAWN a real treat. They all teached me so much and allowed me to have a good time in the office and outside, and for that, they deserve that I thank them here.

Of course, these acknowledgments have to start with Marko Shuntov, my supervisor for this internship. Thank you first for having taken me as an intern at DAWN. Thank you for having explained me so many things on astrophysics, astronomy and with SourceXtractor++, especially by giving me some insights on the configuration to use and on the scientific plots to make. Thank you for being so available on Slack.

I thank also Sune Toft, director of DAWN, for hosting me there and for his welcome. Thank you to Helena Baungaard, secretary and HR, for welcoming me, for your fight for my access in my first office and for the social events at Bakken and Tivoli you organized. You really made my stay at DAWN and in Denmark an awesome time !

During this internship, I shared my office with three others internes or PhD students : Negin, Matthieu and Olivia. Thank you for the nice atmosphere, for our discussions and for all our foosball games on lunch time !

Thank you to Gabe Brammer for your precious help with AWS and the DJA. Thank you to all the PhD students for the very nice and friendly mood at DAWN and during our social days. And thank you to all the researchers at DAWN for the journal club, cake talks and meetings where I learned so much about astrophysics, especially high-redshift and EoR.

For the financing, I have to thank very much the ESDS chaire at Ecole polytechnique for the internship grant, especially Pascal Chabert and Sylvie Pottier. I must also thank the Erasmus+ program for the study grant I received. They really helped me live nicely in Copenhagen and enabled me to visit Denmark and Sweden as well, while I was there and since they're pretty far from France (since I refuse to take the airplane for tourism).

References

Astropy Collaboration, Adrian M. Price-Whelan, Pey Lian Lim, Nicholas Earl, Nathaniel Starkman, Larry Bradley, David L. Shupe, Aarya A. Patil, Lia Corrales, C. E. Brasseur, Maximilian Nöthe, Axel Donath, Erik Tollerud, Brett M. Morris, Adam Ginsburg, Eero Vaher, Benjamin A. Weaver, James Tocknell, William Jamieson, Marten H. van Kerkwijk, Thomas P. Robitaille, Bruce Merry, Matteo Bachetti, H. Moritz Günther, Thomas L. Aldcroft, Jaime A. Alvarado-Montes, Anne M. Archibald, Attila Bódi, Shreyas Bapat, Geert Barentsen, Juanjo Bazán, Manish Biswas, Médéric Boquien, D. J. Burke, Daria Cara, Mihai Cara, Kyle E. Conroy, Simon Conseil, Matthew W. Craig, Robert M. Cross, Kelle L. Cruz, Francesco D'Eugenio, Nadia Dencheva, Hadrien A. R. Devillepoix, Jörg P. Dietrich, Arthur Davis Eigenbrot, Thomas Erben, Leonardo Ferreira, Daniel Foreman-Mackey, Ryan Fox, Nabil Freij, Suyog Garg, Robel Geda, Lauren Glattly, Yash Gondhalekar, Karl D. Gordon, David Grant, Perry Greenfield, Austen M. Groener, Steve Guest, Sebastian Gurovich, Rasmus Handberg, Akeem Hart, Zac Hatfield-Dodds, Derek Homeier, Griffin Hosseinzadeh, Tim Jenness, Craig K. Jones, Prajwel Joseph, J. Bryce Kalmbach, Emir Karamehmetoglu, Mikołaj Kałuszyński, Michael S. P. Kelley, Nicholas Kern, Wolfgang E. Kerzendorf, Eric W. Koch, Shankar Kulumani, Antony Lee, Chun Ly, Zhiyuan Ma, Conor MacBride, Jakob M. Maljaars, Demitri Muna, N. A. Murphy, Henrik Norman, Richard O'Steen, Kyle A. Oman, Camilla Pacifici, Sergio Pascual, J. Pascual-Granado, Rohit R. Patil, Gabriel I. Perren, Timothy E. Pickering, Tanuj Rastogi, Benjamin R. Roulston, Daniel F. Ryan, Eli S. Rykoff, Jose Sabater, Parikshit Sakurikar, Jesús Salgado, Aniket Sanghi, Nicholas Saunders, Volodymyr Savchenko, Ludwig Schwardt, Michael Seifert-Eckert, Albert Y. Shih, Anany Shrey Jain, Gyanendra Shukla, Jonathan Sick, Chris Simpson, Sudheesh Singanamalla, Leo P. Singer, Jaladh Singhal, Manodeep Sinha, Brigitta M. Sipőcz, Lee R. Spitler, David Stansby, Ole Streicher, Jani Šumak, John D. Swinbank, Dan S. Taranu, Nikita Tewary, Grant R. Tremblay, Miguel de Val-Borro, Samuel J. Van Kooten, Zlatan Vasović, Shresth Verma, José Vinícius de Miranda Cardoso, Peter K. G. Williams, Tom J. Wilson, Benjamin Winkel, W. M. Wood-Vasey, Rui Xue, Peter Yoachim, Chen Zhang, Andrea Zonca, and Astropy Project Contributors. The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package. *The Astrophysical Journal*, 935(2):167, August 2022. doi: 10.3847/1538-4357/ac7c74.

H. Atek, M. Shuntov, L. Furtak, J. Richard, J. Kneib, G. Mahler, A. Zitrin, H. McCracken, S. Charlot, J. Chevallard, and Iryna Chemerynska. Revealing galaxy candidates out to $z \sim 16$ with jwst observations of the lensing cluster smacs0723. 2022.

E. Bertin. Automated Morphometry with SExtractor and PSFEx. In I. N. Evans, A. Accomazzi, D. J. Mink, and A. H. Rots, editors, *Astronomical Data Analysis Software and Systems XX*, volume 442 of *Astronomical Society of the Pacific Conference Series*, page 435, July 2011.

E. Bertin, M. Schefer, N. Apostolakos, A. Álvarez-Ayllón, P. Dubath, and M. Kümmel. The SourceXtractor++ Software. In R. Pizzo, E. R. Deul, J. D. Mol, J. de Plaa, and H. Verkouter, editors, *Astronomical Data Analysis Software and Systems XXIX*, volume 527 of *Astronomical Society of the Pacific Conference Series*, page 461, January 2020.

É. Bertin and S. Arnouts. Sextractor: Software for source extraction. *Astronomy & Astrophysics Supplement Series*, 117:393–404, 1996.

G. Brammer, P. V. van Dokkum, and P. Coppi. Eazy: A fast, public photometric redshift code. *The Astrophysical Journal*, 686:1503 – 1513, 2008.

Gabriel Brammer. grizli, September 2023. URL `https://doi.org/10.5281/zenodo.8370018`.

Howard Bushouse, Jonathan Eisenhamer, Nadia Dencheva, James Davies, Perry Greenfield, Jane Morrison, Phil Hodge, Bernie Simon, David Grumm, Michael Droettboom, Edward Slavich, Megan Sosey, Tyler Pauly, Todd Miller, Robert Jedrzejewski, Warren Hack, David Davis, Steven Crawford, David Law, Karl Gordon, Michael Regan, Mihai Cara, Ken MacDonald, Larry Bradley, Clare Shanahan, William Jamieson, Mairan Teodoro, Thomas Williams, and Maria Pena-Guerrero. Jwst calibration pipeline, March 2024. URL `https://doi.org/10.5281/zenodo.10870758`.

S. Carniani, K. Hainline, F. d'Eugenio, D. Eisenstein, P. Jakobsen, J. Witstok, B. Johnson, J. Chevallard, R. Maiolino, J. M. Helton, C. Willott, B. Robertson, S. Alberts, S. Arribas, W. Baker, R. Bhatawdekar, K. Boyett, A. Bunker, A. Cameron, P. Cargile, S. Charlot, M. Curti, E. Curtis-Lake, E. Egami, G. Giardino, Kate Isaak, Z. Ji, G. Jones, M. Maseda, E. Parlanti, T. Rawle, G. Rieke, Marcia J. Rieke, B. R. D. Pino, A. Saxena, J. Scholtz, R. Smit, F. Sun, S. Tacchella, H. Ubler, G. Venturi, C. Williams, and C. Willmer. A shining cosmic dawn: spectroscopic confirmation of two luminous galaxies at $z \sim 14$. 2024.

Adam Ginsburg Christoph Deil. reproject, April 2024. URL `https://reproject.readthedocs.io/en/stable/index.html#`.

L. Ciotti and G. Bertin. Analytical properties of the $R^{1/m}$ law. *Astronomy and Astrophysics*, 352:447–451, December 1999. doi: 10.48550/arXiv.astro-ph/9911078.

Christopher J. Conselice. The Evolution of Galaxy Structure Over Cosmic Time. *Annual Review of Astronomy and Astrophysics*, 52:291–337, August 2014. doi: 10.1146/annurev-astro-081913-040037.

R. Doyon, J. Hutchings, M. Beaulieu, L. Albert, D. Lafreniére, C. Willott, D. Touahri, N. Rowlands, M. Maszkiewicz, A. Fullerton, K. Volk, A. Martel, P. Chayer, A. Sivaramakrishnan, R. Abraham, L. Ferrarese, R. Jayawardhana, D. Johnstone, M. Meyer, J. Pipher, and M. Sawicki. The jwst fine guidance sensor (fgs) and near-infrared imager and slitless spectrograph (niriss). In *Other Conferences*, volume 8442, 2012.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Second International Conference on Knowledge Discovery and Data Mining (KDD'96). Proceedings of a conference held August 2-4*, pages 226–331, January 1996.

M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981. URL `https://dl.acm.org/doi/10.1145/358669.358692`.

A. S. Fruchter and R. N. Hook. Drizzle: A Method for the Linear Reconstruction of Undersampled Images. *Publications of the Astronomical Society of the Pacific*, 114(792):144–152, February 2002. doi: 10.1086/338393.

L. Girardi, G. Bertelli, A. Bressan, C. Chiosi, M. A. T. Groenewegen, P. Marigo, B. Salasnich, and A. Weiss. Theoretical isochrones in several photometric systems: I. johnson-cousins-glass, hst/wfpc2, hst/nicmos, washington, and eso imaging survey filter sets. *Astronomy and Astrophysics*, 391(1):195 – 212, July 2002. ISSN 1432-0746. doi: 10.1051/0004-6361:20020612. URL `http://dx.doi.org/10.1051/0004-6361:20020612`.

E. Hubble. A relation between distance and radial velocity among extra-galactic nebulae. *Proceedings of the National Academy of Sciences of the United States of America*, 15 3:168–73, 1929.

P. Jakobsen, P. Ferruit, C. Oliveira, S. Arribas, G. Bagnasco, R. Barho, T. Beck, S. Birkmann, T. Böker, A. Bunker, S. Charlot, P. Jong, G. Marchi, R. Ehrenwinkler, M. Falcolini, R. Fels, M. Franx, D. Franz, M. Funke, G. Giardino, X. Gnata, W. Holota, K. Honnen, P. Jensen, M. Jentsch, T. Johnson, D. Jollet, H. Karl, G. Kling, J. Köhler, M. Kolm, N. Kumari, M. Lander, R. Lemke, M. López-Caniego, N. Lützgendorf, R. Maiolino, E. Manjavacas, A. Marston, M. Maschmann, R. Maurer, B. Messerschmidt, S. Moseley, P. Mosner, D. B. Mott, J. Muzerolle, N. Pirzkal, Jacques Pittet, A. Plitzke, W. Posselt, B. Rapp, B. Rauscher, T. Rawle, H. Rix, A. Rödel, P. Rumler, E. Sabbi, J. Salvignol, T. Schmid, M. Sirianni, C. Smith, P. Strada, M. Plate, J. Valenti, T. Wettemann, T. Wiehe, M. Wiesmayer, C. Willott, R. Wright, P. Zeidler, and C. Zincke. The near-infrared spectrograph (nirspec) on the james webb space telescope. i. overview of the instrument and its capabilities. *Astronomy & Astrophysics*, 2022.

Alexie Leauthaud, Richard Massey, Jean-Paul Kneib, Jason Rhodes, David E. Johnston, Peter Capak, Catherine Heymans, Richard S. Ellis, Anton M. Koekemoer, Oliver Le Fèvre, Yannick Mellier, Alexandre Réfrégier, Annie C. Robin, Nick Scoville, Lidia Tasca, James E. Taylor, and Ludovic Van Waerbeke. Weak Gravitational Lensing with COSMOS: Galaxy Selection and Shape Measurements. *The Astrophysical Journal Supplement Series*, 172(1):219–238, September 2007. doi: 10.1086/516598.

Jorryt Matthee, Rohan P. Naidu, Gabriel Brammer, John Chisholm, Anna-Christina Eilers, Andy Goulding, Jenny Greene, Daichi Kashino, Ivo Labbe, Simon J. Lilly, Ruari Mackenzie, Pascal A. Oesch, Andrea Weibel, Stijn Wuyts, Mengyuan Xiao, Rongmon Bordoloi, Rychard Bouwens, Pieter van Dokkum, Garth Illingworth, Ivan Kramarenko, Michael V. Maseda, Charlotte Mason, Romain A. Meyer, Erica J. Nelson, Naveen A. Reddy, Irene Shivaei, Robert A. Simcoe, and Minghao Yue. Little red dots: an abundant population of faint agn at z 5 revealed by the eiger and fresco jwst surveys, 2024. URL `https://arxiv.org/abs/2306.05448`.

P. Oesch, G. Brammer, P. V. Dokkum, G. Illingworth, R. Bouwens, I. Labbé, M. Franx, I. Momcheva, I. Momcheva, M. Ashby, G. Fazio, Vanessa L. González, B. Holden, D. Magee, R. Skelton, R. Smit, L. Spitler, L. Spitler, M. Trenti, and S. Willner. A remarkably luminous galaxy at z = 11.1 measured with hubble space telescope grism spectroscopy. *The Astrophysical Journal*, 819, 2016.

P. Oesch, G. Brammer, R. Naidu, R. Bouwens, J. Chisholm, G. Illingworth, J. Matthee, E. Nelson, Y. Qin, N. Reddy, A. Shapley, I. Shivaei, P. V. van Dokkum, A. Weibel, K. Whitaker, S. Wuyts, A. Covelo-Paz, R. Endsley, Y. Fudamoto, E. Giovinazzo, T. Herard-Demanche, J. Kerutt, I. Kramarenko, I. Labbé, E. Leonova, J. Lin, D. Magee, D. Marchesini, M. Maseda, C. Mason, J. Matharu, R. Meyer, C. Neufeld, G. P. Lyon, D. Schaerer, R. Sharma, M. Shuntov, R. Smit, M. Stefanon, J. Wyithe, and M. Xiao. The jwst fresco survey: Legacy nircam/grism spectroscopy and imaging in the two goods fields. *Monthly Notices of the Royal Astronomical Society*, 2023.

K. Ormerod, C. J. Conselice, N. J. Adams, T. Harvey, D. Austin, J. Trussler, L. Ferreira, J. Caruana, G. Lucatelli, Q. Li, and W. J. Roper. EPOCHS VI: the size and shape evolution of galaxies since z 8 with JWST Observations. *MNRAS*, 527(3):6110–6125, January 2024. doi: 10.1093/mnras/stad3597.

S. Patel, B. Holden, D. Kelson, M. Franx, A. van der Wel, and G. Illingworth. The uvj selection of quiescent and star-forming galaxies: Separating early- and late-type galaxies and isolating edge-on spirals„. *The Astrophysical Journal Letters*, 748, 2011.

W. D. Pence, L. Chiappetti, C. G. Page, R. A. Shaw, and E. Stobie. Definition of the Flexible Image Transport System (FITS), version 3.0. *Astronomy and Astrophysics*, 524:A42, December 2010. doi: 10.1051/0004-6361/201015362.

Chien Y. Peng, Luis C. Ho, Chris D. Impey, and Hans-Walter Rix. Detailed Structural Decomposition of Galaxy Images. *The Astronomical Journal*, 124(1):266–293, July 2002. doi: 10.1086/340952.

Marshall D. Perrin, Rémi Soummer, Erin M. Elliott, Matthew D. Lallo, and Anand Sivaramakrishnan. Simulating point spread functions for the James Webb Space Telescope with WebbPSF. In Mark C. Clampin, Giovanni G. Fazio, Howard A. MacEwen, and Jr. Oschmann, Jacobus M., editors, *Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave*, volume 8442 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 84423D, September 2012. doi: 10.1117/12.925230.

G. Rieke, G. Wright, T. Böker, J. Bouwman, L. Colina, A. Glasse, K. Gordon, T. Greene, M. Güdel, T. Henning, K. Justtanont, P. Lagage, M. Meixner, H. Nørgaard-Nielsen, T. Ray, M. Ressler, E. V. van Dishoeck, and C. Waelkens. The mid-infrared instrument for the james webb space telescope, i: Introduction. *Publications of the Astronomical Society of the Pacific*, 127:584 – 594, 2015.

M. Rieke, D. Kelly, and S. Horner. Overview of james webb space telescope and nircam's role. In *SPIE Optics + Photonics*, volume 5904, 2005.

Marcia J. Rieke, Brant Robertson, Sandro Tacchella, Kevin Hainline, Benjamin D. Johnson, Ryan Hausen, Zhiyuan Ji, Christopher N.A. Willmer, Daniel J. Eisenstein, Dávid Puskás, Stacey Alberts, Santiago Arribas, William M. Baker, Stefi Baum, Rachana Bhatawdekar, Nina Bonaventura, Kristan Boyett, Andrew J. Bunker, Alex J. Cameron, Stefano Carniani, Stephane Charlot, Jacopo Chevallard, Zuyi Chen, Mirko Curti, Emma Curtis-Lake, A. Lola Danhaive, Christa DeCoursey, Alan Dressler, Eiichi Egami, Ryan Endsley, Jakob M. Helton, Raphael E. Hviding, Nimisha Kumari, Tobias J. Looser, Jianwei Lyu, Roberto Maiolino, Michael V. Maseda, Erica J. Nelson, George Rieke, Hans-Walter Rix, Lester Sandles, Aayush Saxena, Katherine Sharpe, Irene Shivaei, Maya Skarbinski, Renske Smit, Daniel P. Stark, Meredith Stone, Katherine A. Suess, Fengwu Sun, Michael Topping, Hannah Übler, Natalia C. Villanueva, Imaan E.B. Wallace, Christina C. Williams, Chris Willott, Lily Whitler, Joris Witstok, and Charity Woodrum. JADES Initial Data Release for the Hubble Ultra Deep Field: Revealing the Faint Infrared Sky with Deep JWST NIRCam Imaging. *The Astrophysical Journal Supplement Series*, 269(1):16, November 2023. doi: 10.3847/1538-4365/acf44d.

Brant E. Robertson. Galaxy Formation and Reionization: Key Unknowns and Expected Breakthroughs by the James Webb Space Telescope. *Annual Review of Astronomy and Astrophysics*, 60:121–158, August 2022. doi: 10.1146/annurev-astro-120221-044656.

V. Sahni and A. Starobinsky. The case for a positive cosmological lambda term. *International Journal of Modern Physics*, 9:373–443, 2000. URL `https://ned.ipac.caltech.edu/level5/March02/Sahni/Sahni_contents.html`.

J. L. Sérsic. Influence of the atmospheric and instrumental dispersion on the brightness distribution in a galaxy. *Boletin de la Asociacion Argentina de Astronomia La Plata Argentina*, 6:41–43, February 1963.

J. R. Weaver, L. Zalesky, V. Kokorev, C. J. R. McPartland, N. Chartab, K. M. L. Gould, M. Shuntov, I. Davidzon, A. Faisst, N. Stickley, P. L. Capak, S. Toft, D. Masters, B. Mobasher, D. B. Sanders, O. B. Kauffmann, H. J. McCracken, O. Ilbert, G. Brammer, and A. Moneti. The Farmer: A Reproducible Profile-fitting Photometry Package for Deep Galaxy Surveys. *The Astrophysical Journal Supplement Series*, 269(1):20, November 2023. doi: 10.3847/1538-4365/acf850.

Andrea Weibel, Pascal A. Oesch, Laia Barrufet, Rashmi Gottumukkala, Richard S. Ellis, Paola Santini, John R. Weaver, Natalie Allen, Rychard Bouwens, Rebecca A. A. Bowler, Gabe Brammer, Adam C. Carnall, Fergus Cullen, Pratika Dayal, Callum T. Donnan, James S. Dunlop, Mauro Giavalisco, Norman A. Grogin, Garth D. Illingworth, Anton M. Koekemoer, Ivo Labbe, Danilo Marchesini, Derek J. McLeod, Ross J. McLure, Rohan P. Naidu, Marko Shuntov, Mauro Stefanon, Sune Toft, and Mengyuan Xiao. Galaxy Build-up in the first 1.5 Gyr of Cosmic History: Insights from the Stellar Mass Function at $z \sim 4-9$ from JWST NIRCam Observations. *arXiv e-prints*, art. arXiv:2403.08872, March 2024. doi: 10.48550/arXiv.2403.08872.