Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Licenciatura en Ciencias de la Computación
Tesina de Grado

Métodos alternativos de clustering para la descomposición dinámica de galaxias en simulaciones astrofísicas

Nicolás Uriel Navall

Director: Dr. Juan Cabral
Co-Director: Dr. Pablo Granitto
Colaboradora: Lic. Valeria Cristiani

Fecha: Septiembre 2024

El paso más importante que puede dar alguien. No es el primero, ¿verdad?

Es el próximo. Siempre el próximo paso.

BRANDON SANDERSON

Reconocimientos

Este trabajo utilizó recursos computacionales del CCAD de la Universidad Nacional de Córdoba, que forman parte del SNCAD del MinCyT de la República Argentina.

Además, también se dio uso de recursos computacionales del Instituto de Astronomía Teórica y Experimental, que forma parte del CONICET. Como así también del Centro Extremeño de Investigación, Innovación Tecnológica y Supercomputación, que forman parte de la Fundación Computación y Tecnologías Avanzadas de Extremadura de la Red Española de Supercomputación.

Agradecimientos

Antes que nada quisiera agradecer a la Universidad de Rosario y todos los integrantes del Departamento de Ciencias de la Computación. Cuya participación educativa fue imprescindible para mi desarrollo profesional.

A mi madre y padre, cuyo ejemplo y presencia (y muchas veces contención emocional) fue lo que me permitió perseverar durante tantos años de estudio.

A mí familia extendida, los que ya no están y los que siguen estando. Quienes me permitieron momentáneamente olvidar el estrés con cada comida compartida.

A mis amigos, los cuales se bancaron incontables despotricaciones sobre mi vida académica, y cuya existencia me recordaba que todo esto, al final del día, valía la pena.

A mis compañeros, quienes me siento afortunado de también incluir en el anterior grupo, pero merecen además una mención especial. No solo compartimos muchas penas y glorias juntos (sumado a una pandemia mundial), sino que además no me es posible exagerar como su apoyo y camaradería fueron los motores que me permitieron llegar hasta acá. A todos ustedes, gracias infinitas.

Al doctor Mario Abadi por su aporte en el capítulo introductorio.

A mí director Juan Cabral, mí co-director Pablo Granitto, y la licenciada Valeria Cristiani, cuyo aporte y guía fueron indispensables para completar este trabajo.

Y por última a Dios, quien me bendijo con todas las herramientas y oportunidades que me brindó, pero aún más importante, rodeó de tantas personas que amo y apoyan en todo lo que me propongo.

Resumen

La astronomía se ha beneficiado enormemente del crecimiento constante en la capacidad de cómputo y los avances en los campos de la minería de datos y la inteligencia artificial. En particular, las modernas simulaciones hidrodinámicas han proporcionado conjuntos de datos masivos de galaxias con niveles de exactitud sin precedentes.

Aprovechando estos avances, este trabajo presenta los resultados de aplicar técnicas de clustering en el área de descomposición dinámica de galaxias, un campo que busca entender cómo las diferentes partes de una galaxia contribuyen a su movimiento y evolución general. Se comparan los resultados obtenidos con los generados por técnicas ya establecidas en el área. Específicamente, se evaluaron los métodos de Hierarchical Clustering, Fuzzy C-Means y Ensemble Agglomerative Clustering, en conjunto con dos técnicas de eliminación de valores atípicos: una propia de la astronomía (Corte en la mitad del radio de masa) y otra de uso general en minería de datos (Isolation Forest).

Los resultados demuestran que Hierarchical Clustering y Fuzzy C-Means pueden obtener resultados cercanos a los métodos establecidos en el área, aunque con limitaciones significativas, especialmente en la identificación de qué componente corresponde a qué grupo. Por otro lado, se observó que las métricas internas de clustering convencionales no son útiles para analizar el desempeño en este problema debido a la naturaleza particular de los datos astronómicos. En cuanto al Evidence Accumulation Clustering, mostró resultados distantes de los esperados y un alto costo computacional, lo que desaconseja su uso en la forma explorada. Un hallazgo importante fue la incapacidad de determinar si la eliminación de valores atípicos beneficia la tarea de búsqueda de grupos en este contexto específico.

Este trabajo abre camino a futuras investigaciones en el campo, incluyendo la extensión del conjunto de datos para incluir galaxias espirales, la exploración de técnicas de normalización y preprocesamiento adicionales, y la evaluación de otras técnicas de eliminación de valores atípicos y características del conjunto de datos que puedan ser relevantes para la descomposición dinámica de galaxias.

Palabras Claves: Minería de datos - Aprendizaje no supervisado - Clustering - Eliminación de atípicos - Simulaciones numéricas - Descomposición dinámica - Parámetro de circularidad

Abstract

Astronomy has benefited enormously from the constant improvements in computing power and the advances in fields like data mining and artificial intelligence. Specifically, modern hydrodynamic simulations have provided massive data sets of galaxies with unprecedented accuracy.

Leveraging these advancements, the work presented here presents the results of applying clustering techniques in the dynamic decomposition of galaxies, a field that seeks an understanding of how different parts of a galaxy contribute to its movements and general evolution. We compare the results obtained with the ones generated by already established methods in the area. Specifically, we will evaluate Hierarchical Clustering, Fuzzy C-Means, and Ensemble Agglomerative Clustering, together with two outliers removal techniques: one from coming from astronomy (Radial distance cut) and another one coming from data mining (Isolation Forest).

The results show that both Hierarchical Clustering and Fuzzy C-Means can get similar results to the already established methods used in the field, although with significant limitations, especially in identifying which component corresponds to which group. On the other hand, we found that internal validation metrics used on conventional clustering aren't useful for analyzing the performance of these techniques due to the nature of astronomy data. As to Evidence Accumulation Clustering, it displayed results that are far from the expected ones and at a very high computational cost, which discourages its usage in the way we explored it. An important discovery was the inability to determine if removing outliers benefits the clustering process in this specific context.

This thesis opens a path for future research in the field, including the extension of the dataset to include spiral galaxies, the exploration of normalization techniques and additional preprocess work, and the evaluation of other outliers removal methods and features of the data set that can be relevant for the dynamical decomposition of galaxies.

Keywords: Data Minning - Unsupervised learning - Clustering - Outlier removal - Numerical simulations - Dynamic decomposition - Circularity parameter

Índice general

Capítulo 1

Introducción

Hoy en día disponemos de una enorme cantidad de datos, cuya calidad y cantidad sigue creciendo constantemente. Es por esto que la minería de datos y el aprendizaje automatizado son más relevantes que nunca. El volumen de dicha información es tal que no tiene sentido que su extracción y análisis sea una tarea netamente humana (Hey et al., 2009). El campo de la astronomía no se ve ajena a esto, y de entre todos los tipos de datos relevantes para la disciplina, se encuentran los asociados a las galaxias y simulaciones cosmológicas.

El proceso de formación y evolución de galaxias es extremadamente complejo (White and Rees, 1978; White and Frenk, 1991; Mo et al., 1998), por lo cual las simulaciones numéricas son una herramienta fundamental para estudiar la formación de estos objetos, sus componentes estelares, y los halos de materia oscura donde se forman. Mientras que la evolución dinámica de la materia oscura es calculada en detalle a través de la interacción gravitacional de N-cuerpos, la interacción de los bariones (estrellas y gas) se calcula a través de las interacciones hidrodinámicas que tienen en cuenta gradientes de presión, enfriamiento radiativo, formación estelar, procesos de realimentación, enriquecimiento químico, etc.

En este sentido, entre las simulaciones hidrodinámicas cosmológicas de vanguardia existentes están las simulaciones EAGLE (Schaye et al., 2015), y The Next Generation Illustris Simulations (TNG) Pillepich et al., 2018. Estas simulaciones representan el estado del arte en cuanto a combinar una alta resolución numérica con volúmenes computacionales cosmológicos suficientemente grandes para que incluyan decenas de miles de galaxias individuales. Típicamente cada una de estas galaxias está representada por decenas de miles de partículas que representan un elemento de masa del orden de $\sim 10^5$ masas solares. Como curiosidad agregamos en la Fig 1.1 una imagen del final (tiempo actual) de una simulación de TNG para demostrar el realismo que logra este proyecto.

Estas galaxias son sistemas estelares complejos formados por varias componentes estelares que coexisten simultáneamente y que definen su morfología y cinemática. Las más prominentes son el núcleo, disco fino y grueso, halo estelar, barra y brazos espirales. Para estudiar la formación y evolución de las galaxias es indispensable entender el proceso de formación y evolución de cada una de estas componentes. En este sentido es fundamental contar con un método de descomposición dinámica que permita asignar cada partícula a una componente estelar dada. Un ejemplo de un método de asignación fue implementado por

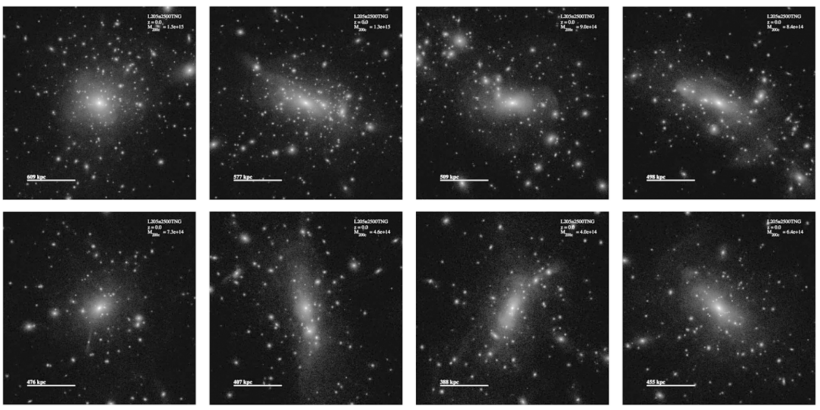

Figura 1.1: Conjunto de la luz intra cúmulo extendida (ICL) de los 8 cúmulos galácticos más masivos en TNG300-1 en tiempo actual.

Cristiani (2020) como una mejora del método original desarrollado por Abadi et al. (2003).

Esta situación provee la base y motivación para el desarrollo del presente trabajo ya que existen formas computacionales basadas en Minería de datos (DM) y Aprendizaje automático (ML) para la asignación de observaciones (partículas) a grupos (componentes) de manera automática.

1.1. Alcances y Objetivos

El objetivo principal es avanzar en la comprensión de las galaxias simuladas por TNG desde el punto de vista de sus subestructuras o componentes, utilizando un enfoque multidisciplinario que combine técnicas de ML y astronomía.

Dado que las partículas de las galaxias simuladas no están etiquetadas como pertenecientes a una componente dada, para agruparlas se debe recurrir a técnicas de clustering del aprendizaje no-supervisado.

Esta ausencia de etiquetas extiende el problema a que no es posible de manera directa determinar la calidad de una descomposición, por lo que las técnicas de validación interna nos van a permitir cuantificar los resultados respecto a la estructura interna de las componentes identificadas, sin necesidad de información ajena al conjunto de datos.

Igualmente, podemos evaluar las componentes encontradas, utilizando métricas clásicas de aprendizaje supervisado, comparando los resultados con los obtenidos por métodos de descomposición existentes que están bien probados en la comunidad y poseen un sólido fundamento físico (Abadi et al., 2003; Du et al., 2019) que nos sirven de Ground Truth (GT) para evaluar los métodos que implementemos. Finalmente es factible aprovechar análisis físicos del área de dinámica galáctica para una evaluación más rigurosa de los resultados obtenidos dentro del dominio particular del problema.

Como último detalle, no pretendemos estudiar cómo se comportan los métodos en ambientes que han sido limpiados de valores atípicos, ya sea por técnicas

utilizados en la ciencia de datos, o con métodos utilizados dentro del área de la astronomía.

1.1.1. Aplicaciones

El trabajo desarrollado posee diversas aplicaciones en el campo de la astronomía y el aprendizaje automático. Entre ellos, pueden mencionarse:

Avance en la comprensión de galaxias: Dado que con el trabajo planteado se logra profundizar la comprensión de las galaxias simuladas por TNG al combinar ML y astronomía.

Evaluación de nuevos métodos de clustering: En el misma línea, el aplicar técnicas de clustering no utilizadas en el área de formación y evolución de galaxias permitirá a los astrónomos reducir incertidumbres sobre las características físicas que resalten diferentes estrategias de clustering.

Metodología de evaluación de resultados: Un subproducto interesante de este trabajo es que aporta una metodología de evaluación de los métodos de descomposición dinámica de galaxias, pudiendo utilizarse en trabajos futuros del área.

1.2. Resultados Originales Presentados

Las contribuciones más preponderantes se centran principalmente alrededor de la evaluación de tres métodos de clustering pertenecientes a familias que nunca fueron utilizados en el área de formación y evolución de galaxias:

- Evaluación de métodos de Clustering Jerárquico (HC) teniendo en cuenta que la estructura de las componentes galácticas están jerárquicamente organizadas.
- Evaluación de métodos de Fuzzy Clustering (FCM), teniendo en cuenta que las partículas al ser de varias masas solares pueden pertenecer a más de una componente al mismo tiempo.
- Evaluación de métodos de Evidence Accumulation Clustering (EAC), para intentar capturar la compleja estructura física de las componentes que diferentes métodos de clustering pueden llegar a encontrar individualmente.

1.3. Organización de la tesis

Los capítulos de este trabajo se organizan de la siguiente manera: en el Capítulo 2 se describe el estado del arte en técnicas de ML en el área de descomposición dinámica así como las técnicas de clustering y evaluación que utilizamos a lo largo de todo el trabajo. El Capítulo 3 aborda los experimentos realizados con métodos de clustering jerárquicos en dos y más componentes y evaluando la presencia o no de atípicos. Los Capítulos 4 y 5 hacen lo propio utilizando métodos difusos y basados en densidad respectivamente. Finalmente el Capítulo 6 explica las conclusiones y los trabajos a futuro.

Capítulo 2

Antecedentes y estado del arte

2.1. De la astronomía a la astroinformática

En la actualidad nos enfrentamos a una avalancha constante de datos de creciente calidad y cantidad. Esta abundancia de información ha elevado la importancia de la DM y el ML a niveles sin precedentes. El volumen de datos disponible es tan abrumador que su extracción y análisis ya no pueden depender exclusivamente de la capacidad humana (Hey et al., 2009). En el ámbito de la astronomía, este fenómeno no es una excepción, y entre los tipos de datos más relevantes para esta disciplina se incluyen los provenientes de extensos relevamientos (Shen et al., 2003) y simulaciones cosmológicas de vanguardia (Pillepich et al., 2018; Schaye et al., 2015). En otras palabras, los astrónomos interesados en explotar las fuentes de datos deben adentrarse en conceptos como el aprendizaje automático, la identificación de patrones, la minería de datos y la reducción de la dimensionalidad; todas estas técnicas han sido agrupadas bajo la disciplina conocida como Astro-informática (Borne, 2010).

Es importante destacar que la Astro-informática no opera en aislamiento, ya que la masividad de la información actual, ha dado origen a diversas subdisciplinas informáticas que se dedican a abordar la tarea de organizar, acceder, integrar y extraer información de un creciente conjunto de datos, con el propósito de proporcionar apoyo en la toma de decisiones (Fox, 2011).

2.2. Galaxias

Como se menciona en la introducción, en este trabajo estamos interesados en las subestructuras de las galaxias, así que hemos decidido utilizar unos párrafos para explicar por qué es necesario realizar estudio en simulaciones.

Observacionalmente las galaxias se clasifican según su morfología en dos tipos principales: galaxias elípticas, las cuales presentan una apariencia elipsoidal y galaxias espirales, que exhiben una forma discoidal con brazos espirales y en la región central presentan una componente esferoidal Hubble (1936). A modo anecdótico puede mencionarse que esta clasificación es conocida como “Secuen-

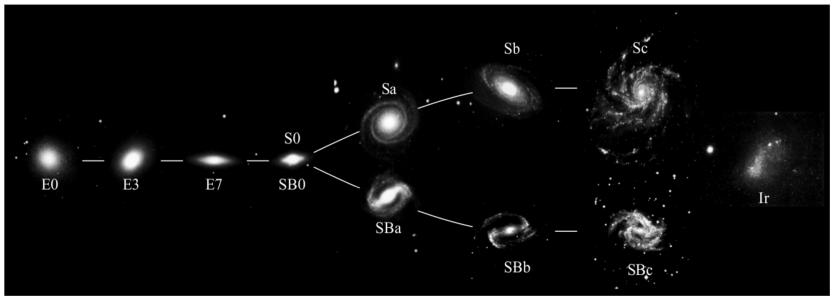

Figura 2.1: Clasificación morfológica o secuencia de Hubble (Carroll and Ostlie, 2006). En la rama principal izquierda pueden observarse arquetipos de galaxias elípticas (E0-E7), S0 es una lenticular, las ramas superior e inferior corresponde a galaxias espirales sin y con barra respectivamente. A la derecha una de tipo irregular.

cia de Hubble", y fue interpretada originalmente como una secuencia evolutiva. Puede observarse la secuencia completa en la Fig 2.1.

Respecto a la estructura podemos destacar dos componentes estelares principales en las galaxias [1]:

- El **esferoide** está compuesto por una población estelar roja y vieja, y, dependiendo el tipo de galaxia, éste puede ser más o menos prominente como el caso de las galaxias elípticas y espirales respectivamente. Además, se caracteriza por contener una baja cantidad de gas y polvo, resultando en una tasa de formación de estrellas relativamente baja (Mo et al., 2010).

  Desde el punto de vista dinámico, las estrellas del esferoide se encuentran soportadas por dispersión de velocidades, aunque puede presentar una rotación mínima.

  Adicionalmente, se puede subdividir dicha componente en las subcomponentes llamadas Bulge y Halo.

- En el caso del **disco**, está compuesto principalmente por estrellas jóvenes, gas y polvo, los cuales se encuentran en rotación en un plano preferencial en torno al centro de la galaxia. Este es la componente principal de las galaxias espirales y lenticulares en lo que respecta a su masa. Además, el disco presenta una mayor tasa de formación estelar respecto al esferoide, debido a la existencia de gas en forma de hidrógeno, resultando en un color más azul gracias a la presencia de estrellas jóvenes. En particular, las galaxias espirales cuentan, como su nombre lo indica, con una estructura en forma de brazos espirales (Mo et al., 2010).

  En galaxias de canto, se pueden identificar dos subcomponentes distintas: un disco delgado y otro grueso, con escala de radios similares, pero el primero es más delgado en altura y contiene generalmente poblaciones estelares más jóvenes, con un mayor contenido de gas frío y polvo que el segundo. Estos también son llamados "Cold Disk" y "Warm Disk" en

[1] Las galaxias además de estrellas poseen nubes de gas, polvo y materia oscura, elementos que están fuera del objeto de estudio de este trabajo.

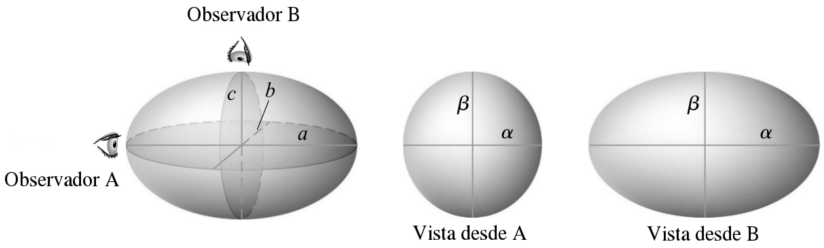

Figura 2.2: Proyecciones de una galaxia elíptica según la posición de dos observadores diferentes (Carroll and Ostlie, 2006)

la terminología en inglés, y utilizaremos ambas denominaciones de forma indistinta a lo largo del texto (en particular para referirnos a dichas subcomponentes en figuras, dado que estas siguen la terminología en inglés).

Es claro entonces que las galaxias son sistemas estelares complejos formados por varias componentes estelares que coexisten simultáneamente, las cuales definen su morfología y cinemática. Si bien queda claro que hay un estructura jerárquica, a lo largo de este trabajo dividiremos cada galaxia en cinco componentes: núcleo, disco fino y grueso, halo estelar y brazos espirales.

2.2.1. Descomposición dinámica de galaxias y simulaciones

Realizar una clasificación basada exclusivamente en la morfología de las galaxias tiene el problema de tener una influencia significativa de los efectos de proyección; ya que si consideramos una galaxia elíptica con semiejes $a > b = c$, observadores diferentes situados en diferentes posiciones percibirán diferentes formas proyectadas de este objeto, como se ilustra en la Fig 2.2, y consecuentemente la clasificación variará en cada caso. El otro problema obvio que se presenta es que en proyección una estrella podría parecer que está en el esferoide mientras que en realidad está en el disco.

Así es que el campo de estudio ha enfocado sus esfuerzos en las características dinámicas de estrellas en las componentes, habiendo en la actualidad una amplia variedad de métodos para llevar a cabo la tarea de descomponer dinámicamente las galaxias.

La naturaleza de este estudio requiere de disponer de las posiciones y velocidades de las estrellas individuales que la conforman, y esto excede a las capacidades observacionales de los telescopios actuales. En este sentido, las simulaciones hidrodinámicas cosmológicas de vanguardia existentes como EAGLE (Schaye et al., 2015) y TNG (Pillepich et al., 2018) nos brindan la posibilidad de acceder a esta información.

Estas simulaciones representan el estado del arte en cuanto a combinar una alta resolución numérica con volúmenes computacionales cosmológicos suficientemente grande para que incluyan decenas de miles de galaxias individuales. Típicamente cada una de estas galaxias está representada por decenas de miles de partículas que representan un elemento de masa del orden de $\sim 10^5 M_\odot$ (masas solares).

Abadi et al. (2003) fueron pioneros en esta área proponiendo el primer método para poder identificar las componentes estelares de una galaxia a partir de la distribución de lo que llamaron el parámetro de circularidad. Con el tiempo, surgieron diferentes variaciones del mismo, como los utilizados en los trabajos de Governato et al. (2009); Scannapieco et al. (2012); Tissera et al. (2012); Vogelsberger et al. (2014); Marinacci et al. (2014) y Park et al. (2019).

2.2.1.1. Parámetros dinámicos y métodos

Llegados a este punto nos resulta importante definir qué parámetros son los que definen la dinámica de una partícula estelar en una galaxia. Así, los tres valores que representan los movimientos de una partícula estelar y definen su ubicación en alguna de las componentes son:

Parámetro de circularidad (ϵ): Mide cuán parecida es la órbita de una partícula respecto a una órbita circular, para un dado valor de energía. Se calcula como el cociente entre la componente z del momento angular y el momento angular circular. Este último es el momento angular que tendría una partícula, si utilizase toda su energía cinética para moverse en una órbita circular.

$$\epsilon = \frac{J_z(E)}{J_{circ}(E)}$$

Parámetro de circularidad proyectada (ϵ_r): Representa la dispersión vertical del movimiento de una partícula con respecto al plano en el que se ubican las partículas que se mueven en órbitas circulares.

$$\epsilon_r = \frac{J_p(E)}{J_{circ}(E)}$$

Siendo $J_P = \sqrt{J_x^2 + J_y^2}$, donde J_x y J_y son las componentes restantes del vector momento angular de la partícula estelar.

Energía estelar normalizada ($E/|E|_{max}$): Es el cociente entre los valores de energía de las partículas estelares y el módulo de la energía de la partícula más ligada de la galaxia.

Representa qué tan ligada está la partícula a la galaxia. Las partículas más ligadas al sistema van a adoptar valores cercanos a -1, mientras que las menos ligadas van a tener valores cercanos a 0. Si el valor es cero la partícula se encuentra en una órbita parabólica mientras que si es mayor a cero la partícula se encuentra en una órbita hiperbólica, en ambos casos éstas no se encuentran ligadas a la galaxia.

El espacio que define esta dinámica puede observarse en la Fig 2.3. Es claro que si bien se sabe que la dinámica de las partículas ubica a las componentes en diferentes zonas de la figura, sigue siendo un espacio muy denso donde no hay divisiones claras entre las componentes.

En los siguientes capítulos nos referiremos al parámetro de circularidad como *eps*, al parámetro de circularidad proyectada como *eps_r* y a la energía estelar

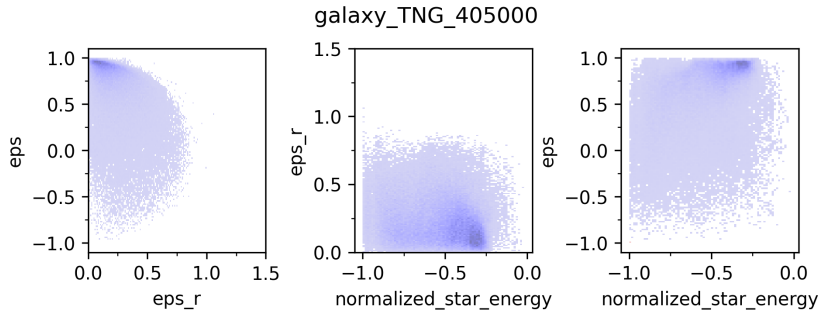

Figura 2.3: Espacio dinámico de la galaxia TNG_405000 en los tres parámetros de circularidad.

normalizada como *normalized_star_energy*. Para, de esta manera, unificar la terminología utilizada en el código y la encontrada en los datos con este trabajo.

Una vez que hemos definido los parámetros, es el momento de establecer los métodos de descomposición dinámica que los utilizarán. En particular, nos centraremos en dos métodos que hacen uso de diferentes sub-conjuntos de los parámetros previamente definidos e identifican diferentes cantidades de componentes.

Abadi et al. (2003): Utiliza el parámetro de circularidad ϵ definido anteriormente para separar las particulares estelares en dos componentes: el esferoide y el disco.

Este método asume que la componente esferoidal no presenta rotación y por lo tanto está representada por una distribución simétrica respecto de $\epsilon = 0$. Para ello selecciona el mismo número de partículas en órbitas corrotantes que aquellas que se encuentran en órbitas contrarrotantes. Luego, la distribución de las partículas restantes, se asignará al disco. Esto se debe a que el método asume que el disco está formado por partículas cuyo movimiento predominante es el de rotación. Como reflejo de esto, las partículas asignadas al disco tendrán $\epsilon \sim 1$.

Du et al. (2019): Realiza distribuciones Gaussianas que representan las estructuras a ser identificadas, utilizando ϵ, ϵ_r y $E/|E|_{max}$ como las dimensiones de dichas distribuciones.

A su vez, genera modelos de dichas estructuras utilizando GMM (Gaussian mixture models), variando el número de componentes a buscar de 2 a 9. En cada iteración, se crean 10 modelos con diferentes inicializaciones para alcanzar resultados más estables.

El algoritmo usa una función a minimizar y un valor de disparo previamente establecido, de forma tal de terminar cuándo el resultado obtenido de esta función sea menor al valor de disparo. Como la iteración comienza con 2 grupos y crece desde ahí, se tiene preferencia en buscar pocas componentes.

El método no asigna ninguna significancia física a cada una de las componentes encontradas, derivando dicha tarea de interpretación al usuario.

Estos dos métodos nos servirán como puntos de referencia para comparar con los algoritmos que implementemos.

2.3. Aprendizaje automático

La actualidad nos ha brindado las computadoras como nuevas fuentes de conocimiento, al punto que como dice el director de investigación en inteligencia artificial de Facebook[2], *Yann LeCun*:

> *En el futuro, la mayor parte del conocimiento del mundo será extraído y almacenado dentro de máquinas.*

Ese futuro predicho por *LeCun*, es en realidad el presente y es el contexto en el cual la Inteligencia Artificial (AI) se vuelve útil para aprovechar el poder de cómputo para que las maquinas auto–analicen su información. La AI tiene diferentes ramas siendo la de éxito actual aquella que postula que se puede imitar la inteligencia humana por medio del aprendizaje. En otras palabras, podemos brindar datos de ejemplo a un programa para que entienda su "forma" y generalice en una tarea dada.

El ML es una de las técnicas mas recurridas y exitosas en tiempos modernos dentro de la Astroestadística (Fluke and Jacobs, 2020). Puede encontrarse una de sus definiciones formales en el libro clásico del área "*Machine Learning*" (Mitchell et al., 1997):

> Se dice que un programa de computadora aprende de la experiencia E respecto a una tarea T y una medida de desempeño P, si el desempeño medido con P en una tarea T, mejora con la experiencia E.

El ML es un proceso inductivo para la exploración de conocimiento, donde su enfoque reside en la búsqueda de información general a partir de observaciones específicas que sesgan las suposiciones de conocimiento concreto, con el propósito de establecer una base sólida para realizar generalizaciones.

El éxito de un método de ML está dado por qué tan correctas resultan estas hipótesis y sesgos. Hay una preferencia particular por usar hipótesis simples, lo que suele llamarse "*Navaja de Ockham*"[3]

2.3.1. Tipos de aprendizaje

Todos los métodos de ML comparten un rasgo fundamental: requieren datos de entrenamiento para adquirir conocimiento. Sin embargo, diversos tipos de tareas (T), experiencias (E) y desempeños (P) conducen a múltiples especializaciones. La clasificación inicial más común categoriza a los modelos de ML en función de cómo el algoritmo utiliza o no valores objetivos a aproximar:

[2] `http://facebook.com`

[3] *Navaja de Ockham*: *En igualdad de condiciones, la explicación más sencilla suele ser la más probable.* Así si dos teorías en igualdad de condiciones tienen las mismas consecuencias, es más probable que la teoría más simple sea la correcta.

Aprendizaje Supervisado Busca lograr una aproximación adecuada de la función $F(X) = y$, en la que X representa los datos de entrada o matriz de *features* (características en inglés) y el vector y corresponde a sus respectivas etiquetas o clases objetivos. Esta aproximación puede tomar la forma de un conjunto finito de etiquetas discretas (clasificación) o intentar predecir un valor numérico real que represente alguna magnitud (regresión).

Aprendizaje No-Supervisado Adquiere conocimiento durante la ejecución de la tarea sin requerir etiquetas. Tareas de este tipo pueden abarcar la reducción de la dimensionalidad, la visualización de datos (manteniendo sus propiedades fundamentales) o la agrupación en conjuntos con características similares.

Sin importar el tipo de aprendizaje que utilicemos, la aplicación de estos algoritmos se separa en dos etapas: la de entrenamiento y la de predicción. En la primera se utilizan los datos de entrenamientos para ajustar los valores internos del modelo, luego de la cual el modelo está listo para realizar predicciones sobre nuevas datos. Además, hay varios modelos que poseen hiperparámetros que pueden ser configurados previamente a la fase de entrenamiento.

Esta distinción de dos etapas del aprendizaje automatizado puede verse reflejada, por ejemplo, en los polinomios. En estos, la etapa de entrenamiento se basa en establecer los puntos del conjunto de entrenamiento como los coeficientes del polinomio (valores internos), siendo la etapa de predicción evaluar el polinomio. Este tipo de método es llamado "Regresión Polinomial".

Dado que las partículas estelares de las galaxias simuladas no poseen etiquetas sobre la componente galáctica a la cual pertenecen, este trabajo centrará sus esfuerzos sobre el aprendizaje no supervisado en general y el *clustering*/agrupamiento en particular.

2.3.2. *Clustering*

Como se mencionó en la sección anterior, es de nuestro interés aplicar métodos de agrupamiento o *clustering*, para asignar las partículas estelares a las diferentes componentes galácticas. Así, los algoritmos de *clustering* persiguen la tarea de agrupar un conjunto de objetos de tal manera que los objetos de un mismo grupo (llamado *cluster*) sean más similares, en algún sentido, entre sí que los de otros grupos (*clusters*). Los algoritmos de esta familia tienen una taxonomía amplia y en este trabajo estamos interesados en tres tipos:

Métodos jerárquicos: Estos métodos tienen la particularidad de crear agrupamientos basándose en agrupamientos más pequeños de una manera jerárquica.

Es esperable que este tipo de algoritmos representen de alguna manera la estructura jerárquica que exista dentro de las componentes galácticas (Johnson, 1967).

Métodos difusos/*fuzzy*: Si bien técnicas de agrupamiento suave ya han sido aplicados al problema de descomposición dinámica como son los trabajos de Obreja et al. (2018) o Du et al. (2019), no existen referencias respecto al uso de agrupamiento difuso.

	Clase Predicha	
Clase Real	**Positiva**	**Negativa**
Positiva	Verdaderos positivos (TP)	Falsos negativos (FN)
Negativa	Falsos positivos (FP)	Verdaderos negativos (TN)

Tabla 2.1: Matriz de confusión para dos clases donde en las filas se muestras las clases reales predichas por nuestra GT, mientras que en las columnas las clases predichas por nuestro método. Donde TP son los verdaderos positivos, y TN los verdaderos negativos, mientras que FN y FP son falsos positivos y falsos negativos respectivamente (asignaciones erróneas por parte de nuestro método, tomando como ciertas las propuestas por el GT).

Dado que una partícula de varias masas solares puede pertenecer a más de una componente galáctica al mismo tiempo, queremos evaluar cómo se comporta los métodos de agrupamientos basados en lógica difusa para la asignación de la misma partícula en diferentes componentes.

Métodos basados en acumulación de evidencia: Este tipo de métodos permite combinar y consensuar resultados provenientes de otras técnicas de clustering.

Sabiendo la complejidad física encontrada en las galaxias, creemos que poder utilizar las mejores características obtenidas dentro de un conjunto de resultados puede resultar en una clusterización que no sería capaz de ser encontrada por ningún método individualmente.

Se expandirá sobre cada uno de los métodos mencionados en su capítulo correspondiente.

2.3.2.1. Métricas externas

Una característica que tienen los métodos de *clustering* es que estos solo son capaces de discernir grupos entre sí, no catalogarlos. Por lo cual, una vez etiquetados los datos de cada galaxia, es necesario compararlos con los obtenidos con algún método que tengan una fundamentación física sólida y además provean estas etiquetas. A los resultados de estos métodos de referencia los consideraremos verdad de referencia o GT.

Dado que nuestro interés radica en probar cómo los métodos antes presentados se pueden configurar para que encuentren diferentes números de grupos, hemos decidido utilizar como GT los métodos de Abadi et al. (2003) y el de Du et al. (2019), teniendo el cuidado de utilizar los mismos parámetros dinámicos para realizar el agrupamiento en cada caso. Así, por ejemplo, si el método de Abadi et al. (2003) sólo utiliza el parámetro de circularidad (ϵ) y busca dos *clusters* y deseamos compararlo con un método jerárquico, también utilizaremos solo ϵ y buscaremos como máximo dos *clusters*.

Comparando los resultados obtenidos por el GT y los del método que estemos evaluando, podemos crear la Matriz de Confusión (CM), como la que se presenta en la Tabla 2.1.

De la CM pueden derivase varias métricas, siendo las más populares:

Precision el cual determina cuánto de lo que seleccioné ($FP + TP$) era realmente relevante (TP):

$$Precision = \frac{TP}{TP + FP}$$

En nuestro contexto, identifica la proporción de partículas bien asignadas a una componente respecto al total que fue asignada a esa componente.

Recall Cuánto de lo relevante $(TP + FN)$ fue seleccionado (TP):

$$Recall = \frac{TP}{TP + FN}$$

Para nosotros la proporción de partículas bien asignadas a una componente respecto al total que debería haber sido asignado a esa componente.

Dado que las métricas externas definidas se aplican a una clase en particular, para obtener valores que representen a todas las clases de nuestro problema es necesario realizar un promedio pesado (teniendo en cuenta los TP de cada clase para lidiar con el desbalance de clases) entre todos los valores de *Precision* y *Recall* de cada clase.

Por ejemplo, supongamos que tenemos un conjunto de datos de flores, y tenemos que identificar cada una como tulipán, rosa y girasol. Una vez obtenido la clase a la cual pertenece cada flor utilizando un método predictivo, podemos comparar dicha etiqueta con la real y así calcular estas métricas para cada tipo de flor. Luego, utilizando dichos cálculos, podemos realizar un promedio pesado y así obtener *Precision* y *Recall* que englobe las 3 clases mencionadas.

2.3.2.2. Métricas internas

Este tipo de métricas utiliza todo el conjunto de datos que se usó para entrenar al modelo, y analiza la distancia intra y extra *cluster*, como así también la continuidad espacial que tienen los datos dentro de sus clusters. Esto da una idea de la estructura interna de los agrupamientos sin necesidad de recurrir a valores externos.

En particular, utilizaremos las métricas de validación internas clásicas en el área de *clustering*: el Puntaje de Silhouette (SSc) (Aranganayagi and Thangavel, 2007) y el Índice de Davies–Bouldin (DBI) (Davies and Bouldin, 1979).

Puntaje de Silhouette (SSc): El "Coeficiente de Silhouette" $s(i)$ para cada punto en un conjunto de datos con más de un grupo se define como

$$s(i) = \frac{b(i) - a(i)}{\text{máx}\{a(i), b(i)\}}$$

donde $a(i)$ es el promedio de la distancia entre el punto i con todos los puntos del mismo *cluster* y $b(i)$ es el promedio de la distancia entre el punto i con todos los puntos del *cluster* más cercano.

Por último, se calcula el promedio del coeficiente de Silhouette de cada punto del conjunto de datos para obtener así el SSc, el cual refleja la calidad de *clustering* sobre el conjunto dado (Rousseeuw, 1987).

Se interpreta el SSc de forma tal que, valores ~ 1 indican que los *clusters* están bien separados entre si y son a su vez bien compactos; mientras que los valores ~ -1 dan a entender que existen puntos mal clasificados; finalmente, si el puntaje es ~ 0 existe una superposición entre *clusters*.

Índice de Davies–Bouldin (DBI): Dadas las siguientes definiciones:

- s_i: La distancia promedio de cada punto en el *cluster* c_i al centroide de su mismo *cluster*.
- d_{ij}: Distancia entre los centros de los *clusters* c_i y c_j.
- R_{ij}: $\frac{s_i+s_j}{d_{ij}}$

DBI busca para todos los *clusters* c_i, el *cluster* c_j que maximiza el valor R_{ij}, para luego promediar todos los valores R_{ij} obtenidos.

La interpretación que se le da a dicha métrica es que valores ~ 0 indican un mejor agrupamiento, mientras que valores alejados de 0 (los cuales pueden crecer infinitamente) dan a entender un mal trabajo de *clustering*.

Vale aclarar que no necesariamente se debe entender a valores alejados de los ideales impuestos por cada métrica como un mal agrupamiento, ya que dependiendo de la naturaleza de los datos, si los *clusters* reales están mezclados entre si (como es nuestro caso), es natural que el puntaje final no se acerque demasiado a lo que se consideran son buenos valores para las métricas mencionadas. Por lo cual, en este trabajo calcularemos también dichas métricas sobre el agrupamiento realizado por los métodos utilizados como GT, para así utilizarlos como referencia.

2.3.2.3. Curva de Perfil de Velocidades (VPC)

La última métrica de evaluación a utilizar es propia de la rama de la descomposición dinámica de galaxias y utiliza propiedades físicas para evaluar el resultado de dicha descomposición.

La idea de utilizar Curva de Perfil de Velocidades (VPC) (Van de Hulst et al., 1957) consiste en, además de utilizar las métricas clásicas del aprendizaje no–supervisado, crear un puente hacia los astrónomos presentando los resultados en una forma que es habitual en el área. Cabe aclarar que existen otras formas de evaluar esta descomposición dentro del área, pero hemos preferido utilizar la más simple y sencilla de entender para el área de ciencias de la computación.

Los perfiles de velocidades muestran la velocidad de rotación de las componentes en una galaxia en función de su distancia radial al centro de las mismas. Por ejemplo en la Fig 2.4 podemos observar que las estrellas del disco se acumulan más cerca del centro de la galaxia, mientras que las partículas del halo predominan en la región más externa de la misma. Estos comportamientos son los esperables para dichas componentes.

Si superponemos nuestras curvas con las calculadas con nuestro GT, vamos a poder ver si en la dinámica de las componentes identificadas se reproducen los comportamientos típicos estudiados en la física; esto lo vuelve una métrica externa.

2.3.3. Atípicos/Outliers

Otro experimento de interés es la posibilidad de eliminar valores atípicos *outliers*. Estos valores son extremadamente raros o pueden suceder de algún tipo de error de medición. Además, en el contexto de simulaciones hidrodinámicas podemos considerar atípicos a valores que están muy alejados del centro galáctico y puede que estén muy débilmente ligados gravitacionalmente.

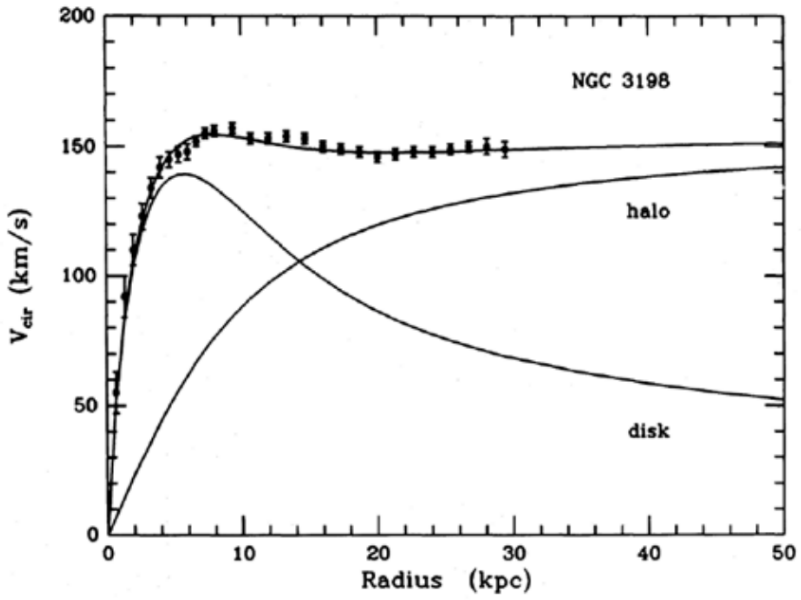

Figura 2.4: Curva de velocidad de rotación de una galaxia espiral barrada (Van Albada et al., 1985)

.

Así, hemos decidido utilizar dos métodos para la eliminación de atípicos para probar cómo se comportan los diferentes algoritmos de clustering: uno algorítmico y otro propio de la astronomía.

2.3.3.1. Isolation Forest (IF)

Liu et al. (2008) Detecta anomalías mediante árboles binarios. Divide el espacio de datos mediante líneas paralelas a la base canónica y asigna puntuaciones de anomalía a los datos, como se puede ver en la Fig 2.5. Otorgando de esta manera valores más altos a los puntos de datos que necesitan menos divisiones para ser aislados.

Para aislar un punto del resto de los datos, el algoritmo genera recursivamente particiones en la muestra seleccionando aleatoriamente un atributo, para luego seleccionar también de forma aleatoria un valor de partición entre los valores mínimo y máximo permitidos por ese atributo.

En el contexto del problema a tratar en este trabajo, dicho método será aplicado sobre las coordenadas en el espacio real de cada partícula estelar, en lugar de utilizar los parámetros de circularidad descritos anteriormente. Para, de esta manera, intentar identificar las partículas alejadas del centro de la galaxia como se mencionó.

2.3.3.2. Corte en distancia radial (RCut)

Desde el punto de vista de la astrofísica, es habitual considerar a las galaxias como las partículas estelares con radio $r <= X$kpc, donde kpc es una medida de

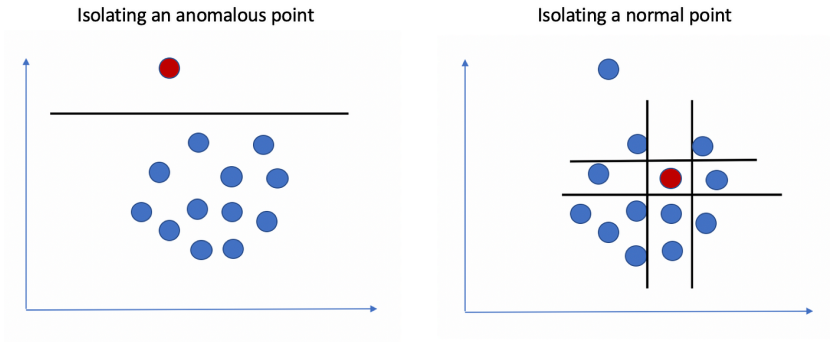

Figura 2.5: Aislando un punto anómalo en un conjunto de puntos utilizando IF. Gráfico proveído por Towards Data Science.

distancia igual a 3260 años luz y X es 3 veces el radio que encierra la mitad de la masa estelar. Esta heurística es usada en diversos trabajos como una forma rápida de descartar valores que es poco probable que pertenezcan a la galaxia, pero que el algoritmo que identifica las galaxias las haya asignado allí. Puede apreciarse el resultado de utilizar dicho método en la Fig 2.6.

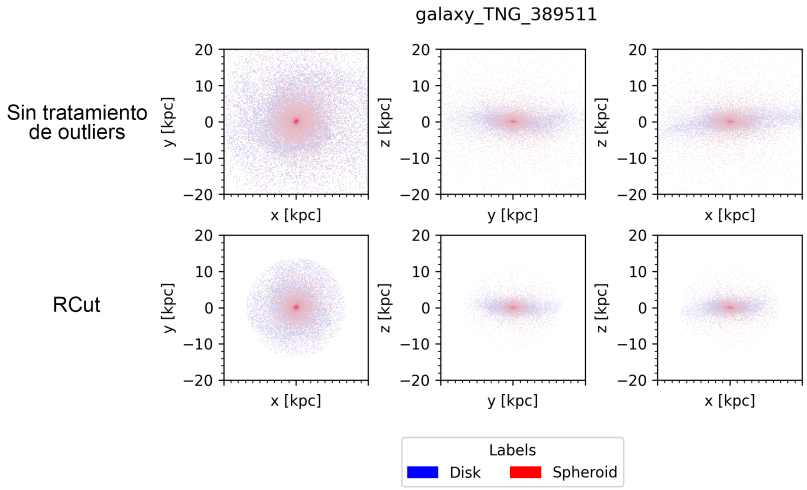

Figura 2.6: Demostración de RCut sobre la galaxia *galaxy_TNG_389511* en espacio real.

Capítulo 3

Clustering Jerárquico

Como se mencionó en el capítulo anterior las galaxias tienen sus componentes en una estructura jerárquicamente organizada, donde el núcleo está contenido dentro del Esferoide y el Disco Fino dentro del grueso. A modo de ejemplo puede observarse en la Fig 3.1 una vista esquemática de cómo se cree que es la forma de nuestra Vía láctea y se puede apreciar esta jerarquía.

En minería de datos, el HC se refiere a una familia de algoritmos que buscan en los datos grupos organizados jerárquicamente. Así, a medida que se aumenten la cantidad de clusters buscados, estos se irán dividiendo o fusionando hasta alcanzar el número deseado dependiendo del tipo del algoritmo utilizado, como se explicará a continuación.

Para usar este enfoque es necesario sospechar que los datos tienen algún tipo de estructura con diferentes niveles de granularidad o resolución, como es el caso de las componentes de una galaxia (Hubble, 1936).

En la práctica la forma de generar esta jerarquía puede realizarse de dos maneras distintas

- **Aglomerativas**: Este es un acercamiento ascendente. Cada observación comienza en su propio grupo, y los pares de grupos son mezclados mientras uno sube en la jerarquía.
- **Divisivas**: Este es un acercamiento descendente. Todas las observaciones comienzan en un grupo, y se realizan divisiones mientras uno baja en la jerarquía.

Para poder decidir qué grupos deberían ser combinados (para métodos aglomerativos), o cuando un grupo debería ser dividido (para métodos divisivos), se requiere una medida de disimilitud entre conjuntos de observaciones. En la mayoría de los métodos de HC, esto es logrado mediante el uso de una medida específica de distancia entre grupos de observaciones, también llamado "*linkage*", que especifica la disimilitud de conjuntos como una función de las distancias dos a dos entre ciertas observaciones en los conjuntos (Sharma et al., 2019).

En este trabajo utilizaremos la implementación de HC aglomerativo provista por scikit-learn (Pedregosa et al., 2011), la distancia euclidiana para medir la distancia entre puntos, y los linkages descritos a continuación:

- **Ward**: Minimiza la suma de las diferencias al cuadrado dentro de todos los clusters. Es un enfoque de minimización de la varianza.

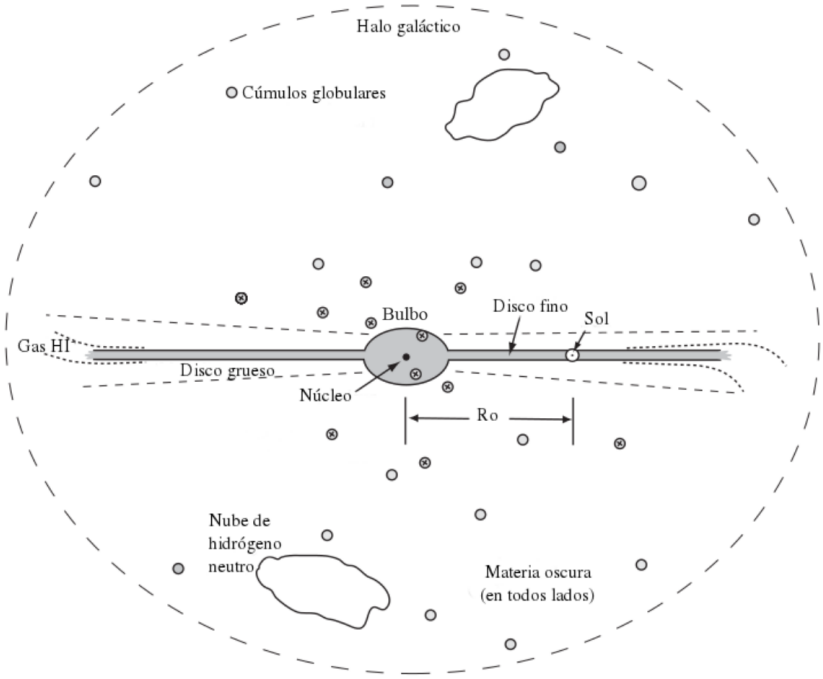

Figura 3.1: Vista esquemática de canto de la Vía Láctea (Sparke and Gallagher III, 2007)

- **Complete**: Minimiza la distancia máxima entre observaciones de pares de clusters.
- **Average**: Minimiza la media de las distancias entre todas las observaciones de pares de conglomerados.
- **Single**: Minimiza la distancia entre las observaciones más cercanas de pares de conglomerados.

Optamos por utilizar HC aglomerativo ya que parece ser el que mejores resultados da en el área de estudio (Yu and Hou, 2022). En cuanto a la distancia euclidiana, esta fue elegida dado que los dominios de los datos con los cuales trabajamos son continuos (Pillepich et al., 2018).

Por otro lado, para decidir sobre el linkage a utilizar realizamos los experimentos descriptos en la siguiente sección.

3.1. Selección de linkage

Para poder concluir en el linkage que nos da mejores resultados procedimos a aplicar HC con todos los linkages y compararlos con el método de Abadi, sobre un conjunto de 9 galaxias.

Como se puede observar en la Fig 3.2, utilizar los linkages Average y Single resultó en un cluster con casi todas las estrellas de la galaxia en una sola componente. Esto se debe a que las partículas estelares de diferentes componentes en

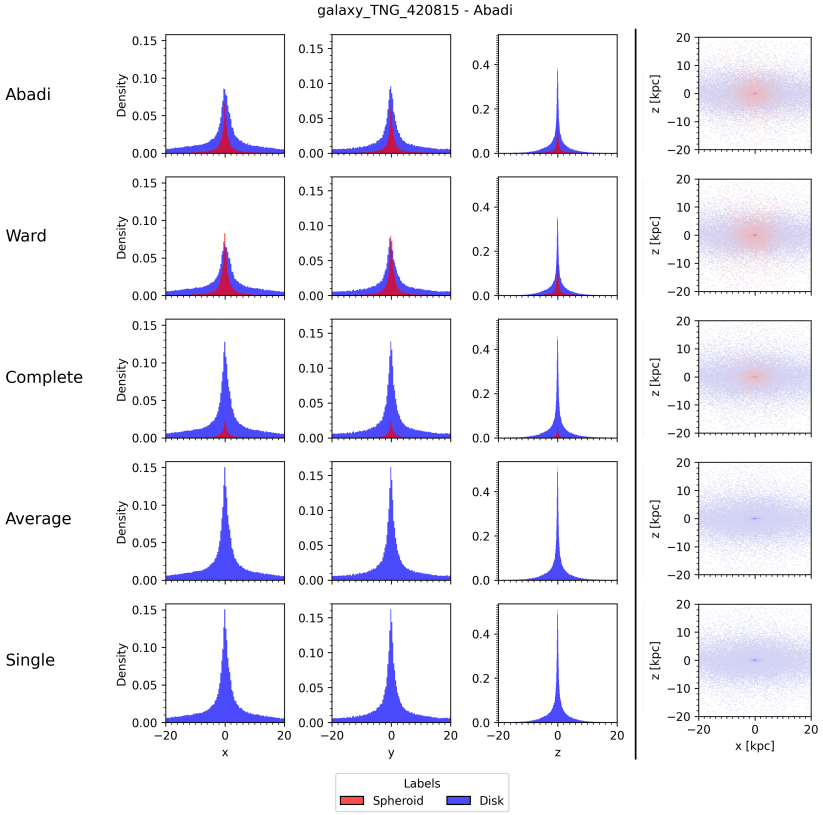

Figura 3.2: Histograma comparando los grupos obtenidos con HC con diferentes linkages contra Abadi sobre la galaxia *galaxy_TNG_420815* en el espacio real. Además se incluyen histogramas comparando eje a eje los grupos obtenidos sobre el espacio real.

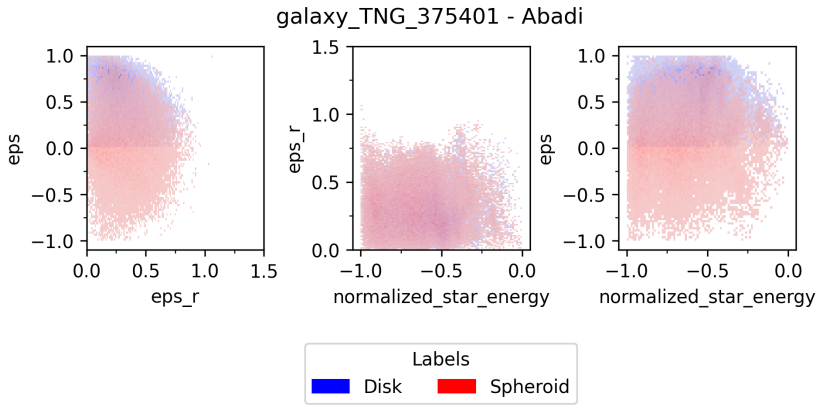

Figura 3.3: Scatterplot mostrando la cercanía de las componentes encontradas por Abadi en espacio circular sobre la galaxia *TNG_420815*.

el espacio circularidad están tan cercanas entre ellas (ver Fig 3.3), que la tarea de encontrar clusters en dicho espacio se ve dificultada. Por lo cual, al utilizar el linkage Single el cual hace uso de la distancia mínima inter-clusters, basta con tener un solo punto alejado del resto de nuestros datos para que éste sea identificado como un único cluster. Y en el caso de linkage Average, su tarea se ve perjudicada por el alto solapamiento de los clusters. Por otro lado con linkage Complete, si bien da algunos resultados similares a Abadi, es inestable y no logra resultados con la consistencia que logra Ward en el Apéndice A.

Finalmente, los buenos resultados de Ward pueden vincularse a como la varianza de las distancia entre clusters es una métrica mucho más consistente e inmune al ruido en comparación al resto.

Puede que en datasets sin galaxias con esferoides los linkages Complete, Average y Single den mejores resultados. Pero dado que tenemos galaxias con esferoides en el dataset utilizado en este trabajo procederemos a usar solamente el linkage Ward en lo que resta del capítulo.

A partir de este momento, nos referiremos a HC con Ward como HC o Ward indistintamente.

3.2. Detección de dos componentes: Disco y esferoide

Antes de comenzar, es importante diferenciar como utilizaremos el término *parámetro* y *feature* de aquí en adelante. Con *parámetro* nos referiremos a las "variables" que el algoritmo de aprendizaje automatizado aprende a partir de su entrenamiento, mientras que el término *feature* será utilizado para referirse a dicha variable en nuestro conjunto de datos (es decir, una "columna" de los datos).

En las siguientes pruebas se procedió a utilizar HC con los mismos features de los datos utilizados por Abadi (Abadi et al., 2003), *eps*, buscando 2 clusters. Los resultados obtenidos pueden observarse en la Fig 3.4. Si prestamos atención a la columna con el gráfico de *eps* vs *normalized_star_energy* se evidencia

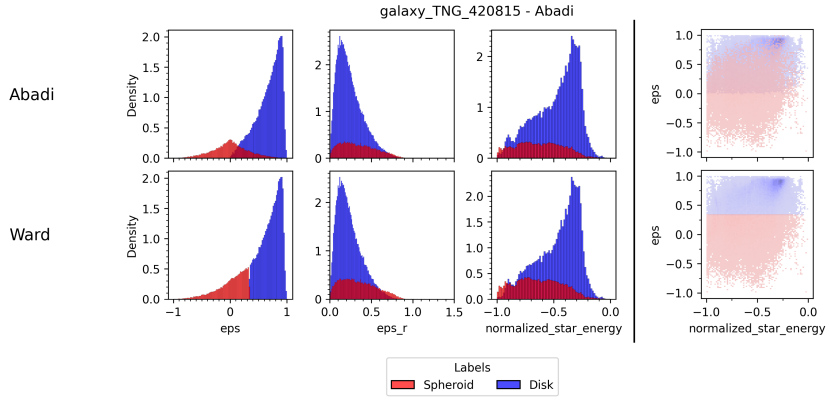

Figura 3.4: Histograma y gráfico de dispersión comparando los resultados obtenidos con Abadi contra los obtenidos con HC con linkage Ward sobre la galaxia *galaxy_TNG_420815* en espacio circular.

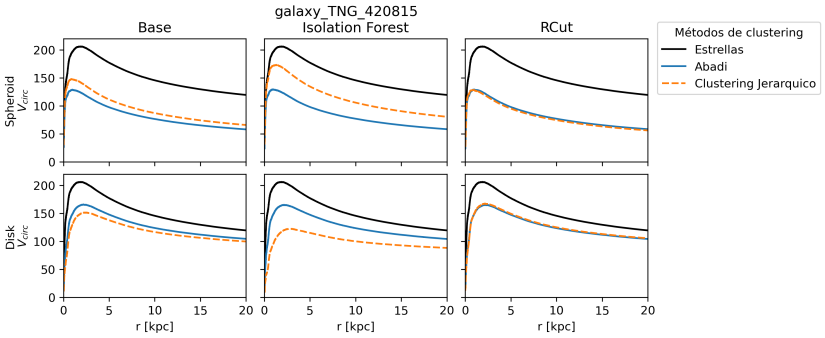

Figura 3.5: Curva de rotación sobre los resultados obtenidos con Abadi y HC sobre la galaxia *galaxy_TNG_420815*, utilizando diferentes métodos de eliminación de outliers.

como al carecer de la heurística de Abadi, Ward no puede cortar de manera adecuada el componente disco en *normalized_star_energy*; esta situación hace que el esferoide sea más grande como puede verse en la Tabla 3.1 y afecta los perfiles de velocidades (primera columna de la Fig 3.5).

	Spheroid	Disk	Total
Abadi	20322	91202	111524
Ward	26065	85459	111524

Tabla 3.1: Comparación partículas estelares etiquetadas en cada componente por Abadi y HC para la galaxia *galaxy_TNG_420815*.

Tomando como GT a Abadi, podemos ver que los valores obtenidos en las métricas de precision y recall (Fig 3.6, y Fig 3.7 columna 1) estan por arriba de 0,5 y son bastante altos.

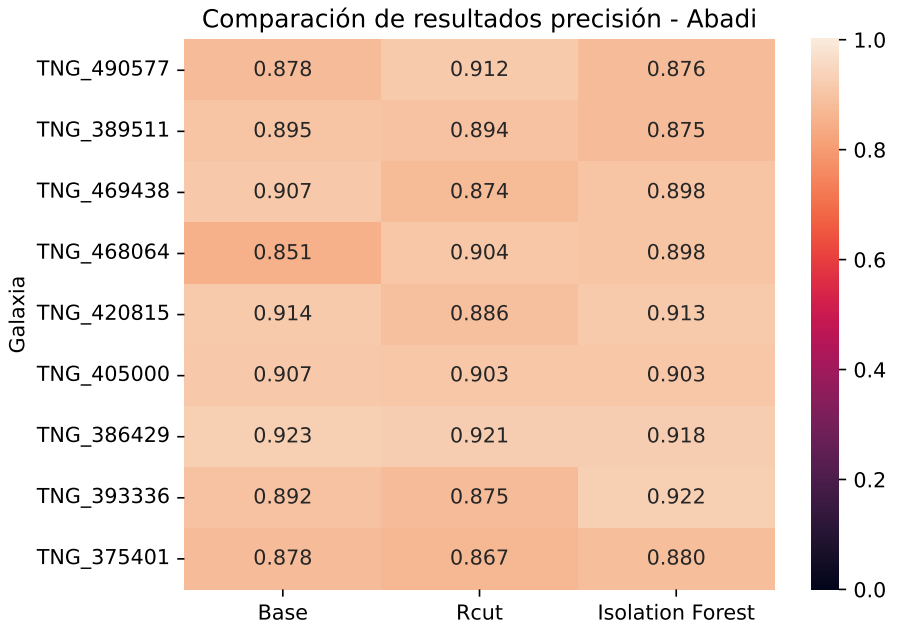

Figura 3.6: Comparación de precisión sobre los resultados obtenidos entre Abadi y HC con Ward.

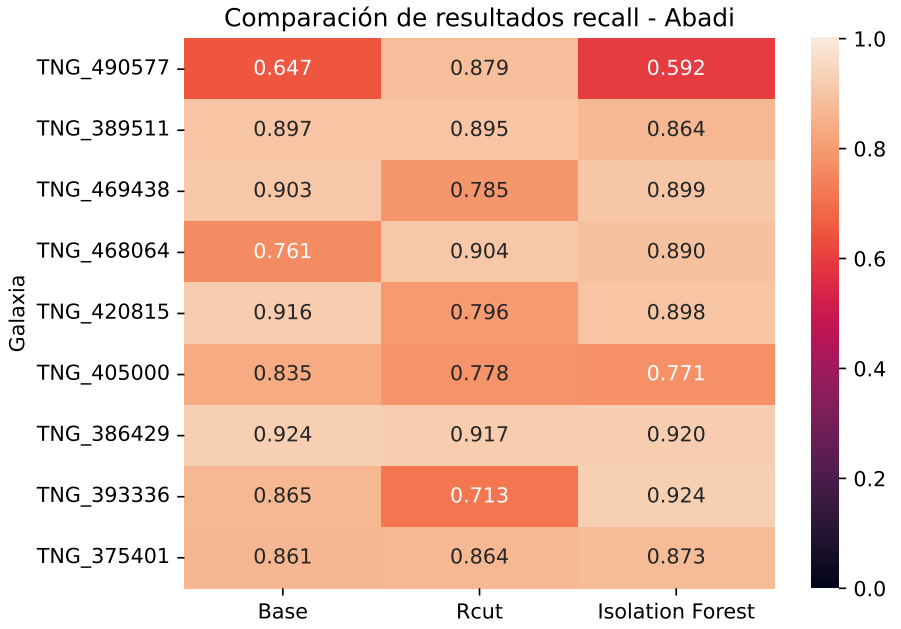

Figura 3.7: Comparación de recall sobre los resultados obtenidos entre Abadi y HC con Ward.

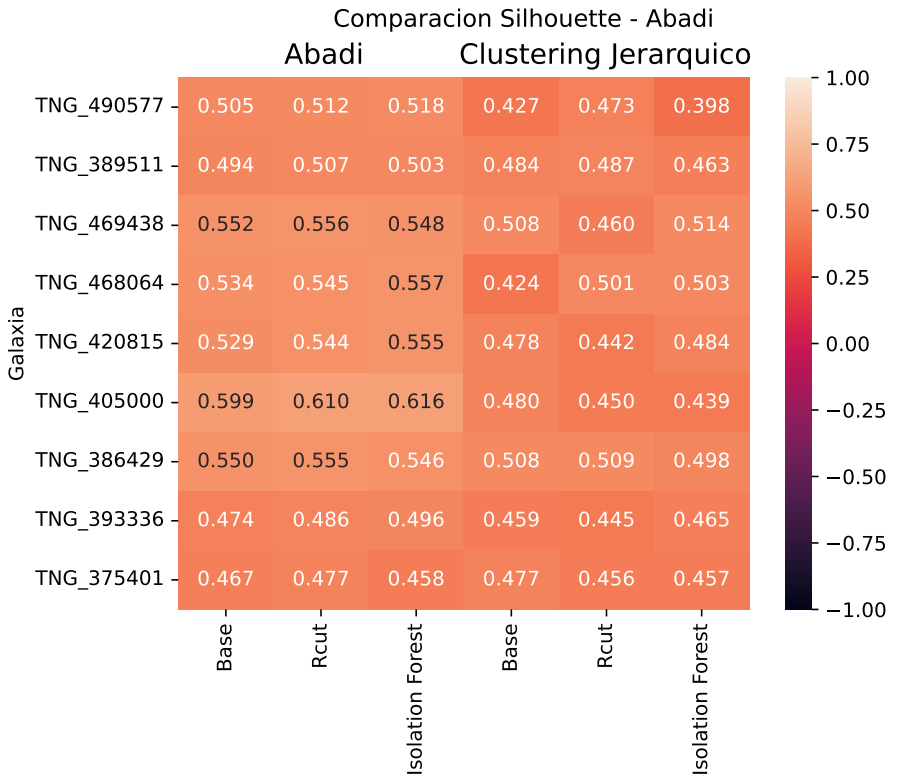

Figura 3.8: Comparación de métrica Silhouette sobre los resultados obtenidos entre Abadi y HC con Ward.

Si lo evaluamos con las métricas internas, podemos ver que Ward no presenta valores muy distintos a Abadi en la tabla de Silhouette (Fig 3.8), pero según Davis-Bouldin (primera y cuarta columna de la Fig 3.9) los clusters resultantes de Abadi son mucho más separados y con menos variación interna.

Los resultados mostrados hasta el momento parecen converger a los obtenidos por el algoritmo de Abadi, dado que HC logró encontrar una estructura similar a este último. Aunque Abadi tenga bases físicas, las cuales son más estables que un análisis meramente estadístico como el que realiza Ward, los artefactos que pueden observarse en los resultados obtenidos con HC nos muestran que hay una tendencia a converger a los obtenidos con Abadi, reforzando así las bases ya establecidas por éste en el campo de investigación actual.

Dicho esto, los resultados obtenidos con HC distan de ser tan buenos como los obtenidos por Abadi. HC parece priorizar las partículas que están rotando a la hora de clasificar el disco, ya que éstas están más separadas del esferoide.

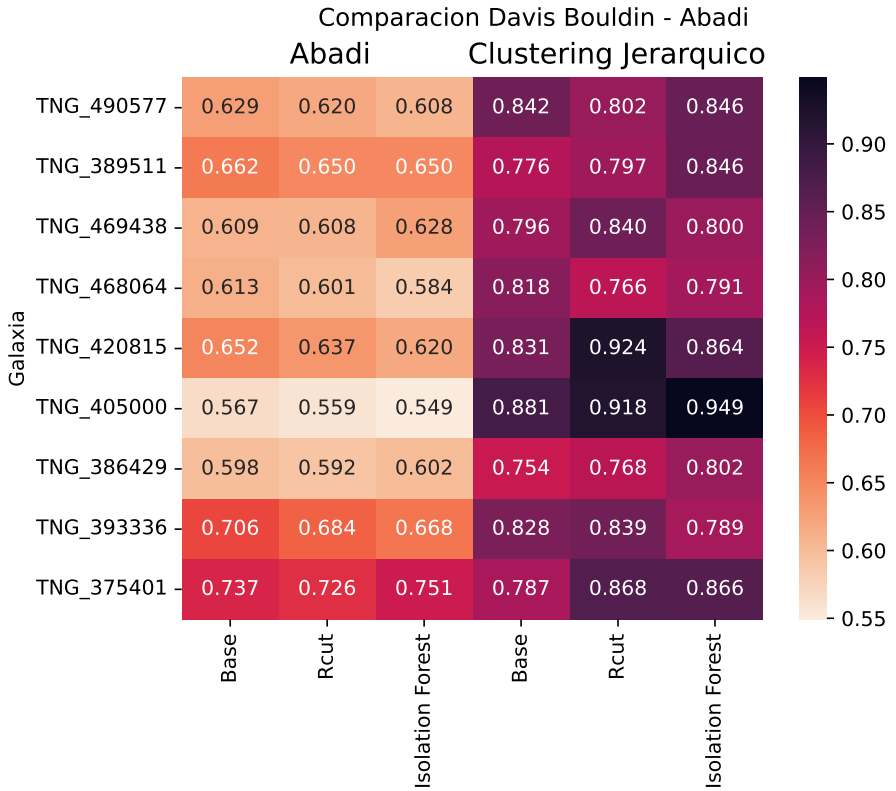

Figura 3.9: Comparación de métrica Davies Bouldin sobre los resultados obtenidos entre Abadi y HC con Ward.

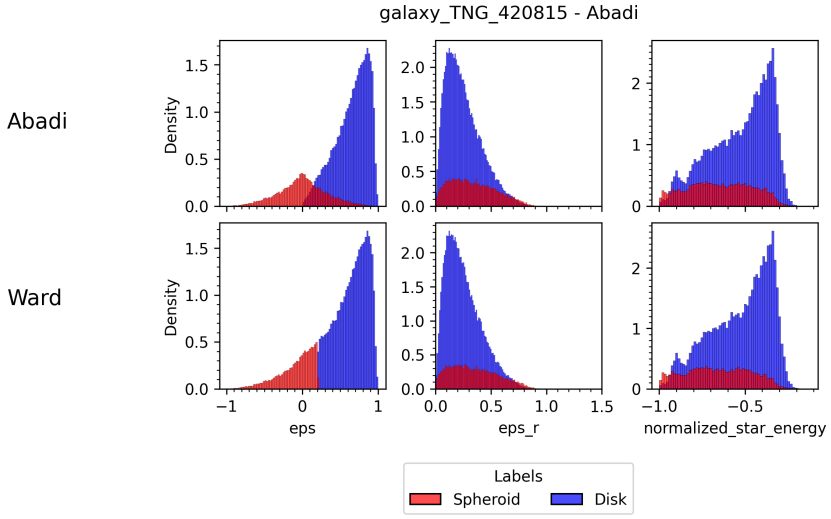

Figura 3.10: Histograma comparando los resultados obtenidos con Abadi contra los obtenidos con HC y linkage Ward, realizando eliminación de outliers con RCut sobre la galaxia *galaxy_TNG_420815* en el espacio circular.

3.2.1. Eliminación de outliers

Como se mencionó brevemente en el capítulo anterior, el método de RCut se basa en eliminar las estrellas cuya distancia desde el centro de la galaxia supera cierto valor.

En la sección anterior, la aplicación de HC sin trata de outliers sobre la galaxia 420815 parecía sobrestimar el esferoide. Podemos ver en la Fig 3.5 y la Fig 3.10 cómo, al aplicar RCut, las componentes encontradas parecen acercarse más a lo encontrado por Abadi (sobrestimando levemente el disco en el proceso). Interpretamos que la causa de dicho suceso fue que, dado que RCut remueve partículas estelares exteriores, y que el disco es la componente más externa de la galaxia, ésta será la componente que en mayor medida se verá afectada. Además, como es la componente con mayores valores de *eps* (como la Fig 3.10 lo muestra en comparación a la Fig 3.4), el pico de densidad de ésta disminuirá, aumentando así la varianza sobre la componente Disco.

Dado que en la Fig 3.11 se puede observar que al eliminar outliers con RCut el pico de densidad de *eps* desciende y la línea de corte en Ward se mueve hacia la derecha, es factible interpretar que dicho resultado proviene de la heurística de HC de disminuir la varianza en sus componentes, y como la reducción del pico de densidad del Disco aumentó la varianza de éste, para compensar el mayor nivel de varianza del Disco, HC mueve la línea de corte hacia la derecha.

Sin embargo, los resultados mostrados en la Fig 3.12 contradicen dicha hipótesis al mover la línea de corte hacia la izquierda al disminuir el pico de densidad de *eps* cuando eliminamos outliers con RCut en Ward.

Al observar los histogramas de espacio circular y gráficas de velocidad del resto de las galaxias en el Apéndice A, no parece haber ninguna clara correlación entre las características de las galaxias y la forma en la que HC encontrará sus componentes luego de remover outliers con RCut.

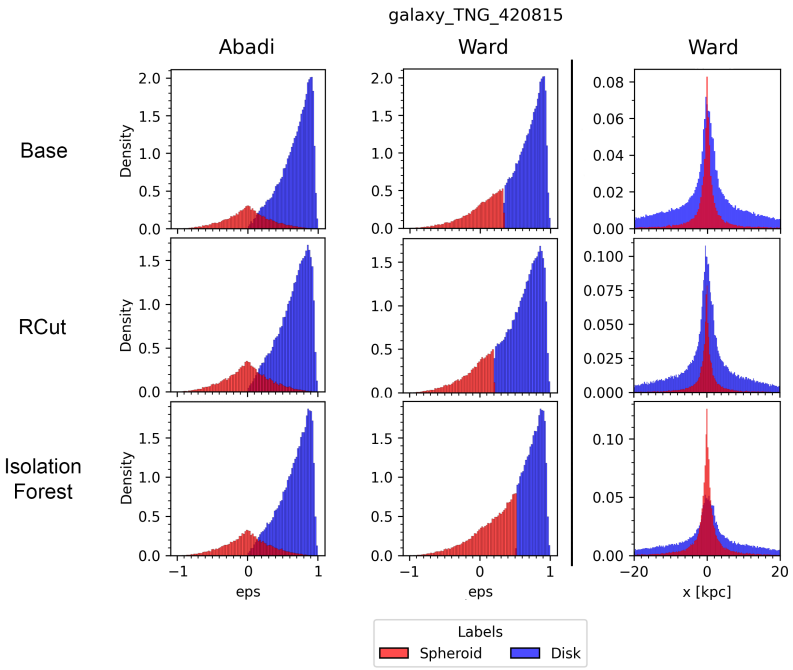

Figura 3.11: Comparación de histogramas en espacio circular y real entre Abadi y HC sin tratamiento de outliers, luego aplicando RCut, y luego aplicando IF sobre la galaxia *galaxy_TNG_420815*.

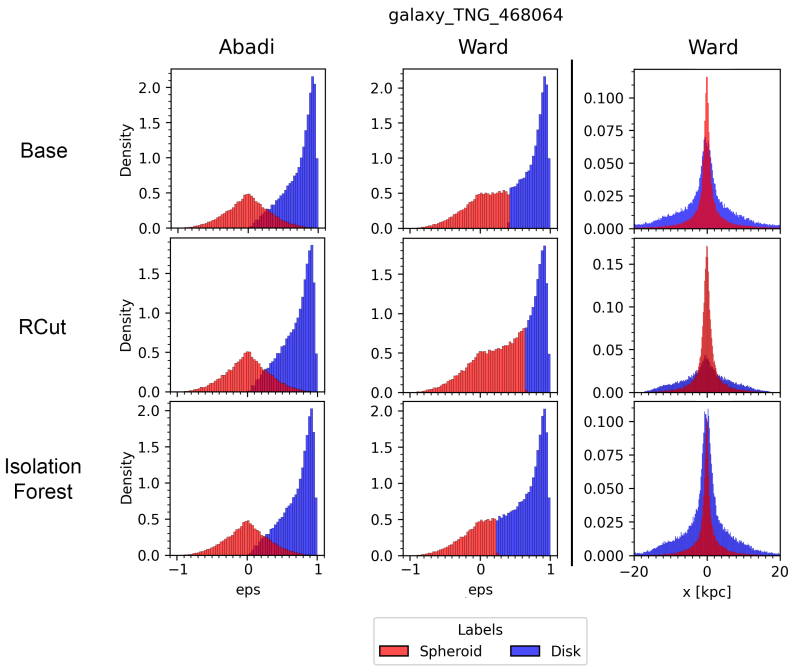

Figura 3.12: Comparación de histogramas en espacio circular y real entre Abadi y HC sin tratamiento de outliers, luego aplicando RCut, y luego aplicando IF sobre la galaxia *galaxy_TNG_468064*.

Por lo tanto, dado que remover outliers con RCut mueve el "corte" generado en *eps* para Ward para la izquierda o derecha de forma arbitraria (independientemente de si se sobrestimó el disco o el esferoide en el análisis previo antes de remover outliers), no podemos concluir que RCut sea beneficioso para encontrar las componentes del Disco y el Esferoide en galaxias utilizando HC.

En cuanto a IF, dado que éste se basa en encontrar puntos fácilmente aislables en el espacio real, se eliminarán outliers en los tres ejes x, y, z. En este caso, dependiendo de la galaxia, no solo se eliminarán partículas del disco sino también del esferoide. Como las partículas del Disco son las más externas en el espacio real, dicha componente será la más afectada a la hora de remover outliers con este método, lo cual resulta en HC moviendo la línea de "corte" en *eps* entre el Esferoide y el Disco hacia la derecha para reducir la varianza de este último, la cual se vio incrementada en comparación con los resultados obtenidos antes de eliminar outliers. Puede apreciarse en la Fig 3.11 un ejemplo de la tendencia de nuestro dataset a converger en este comportamiento, como así también en la Fig 3.12 cómo remover outliers con IF puede lograr el resultado opuesto en ciertas galaxias. Para más gráficas de circularidad y de velocidad sobre el resto de las galaxias referirse al Apéndice A.

Sin embargo, como sucedió con el método anterior, dado que dicha consecuencia solo nos es útil cuando aplicar Ward sin ningún tipo de tratamiento de outliers sobrestima el Disco (de forma tal que IF pueda corregirlo), IF no resulta de utilidad a la hora de buscar dos componentes con HC ya que depende de cómo éstos varían respecto a los resultados con Abadi.

Dado lo que podemos observar al comparar los resultados con Abadi en la Fig 3.6, con o sin remover outliers, se puede concluir que la aproximación de la densidad de cada cluster es bastante acertada. Mientras tanto, el heatmap de recall en la Fig 3.7 nos muestra lo mucho que varió la certeza del Clustering Jerárquico entre varias galaxias.

Como nota final, podemos concluir que el HC se comporta, como mucho, de forma similar a Abadi al buscar 2 clusters, pero como éste es más caro en memoria y computacionalmente (ya que calcula una matriz de distancia, por lo que estamos lidiando con magnitudes de $O(n^2)$ para ambos), el algoritmo de Abadi sigue siendo la mejor alternativa.

3.3. Comparación con Auto Gaussian Mixture - >2 clusters

Es necesario mencionar que, dado los malos resultados obtenidos por otros linkages al buscar dos componentes en la sección anterior, no sería de esperar que podamos obtener mejores resultados en esta sección, por lo cual no los revisaremos aquí.

Dado que ahora estamos buscando más de dos componentes, es natural pensar que dichas componentes surgirán de los clusters encontrados en la sección anterior. Es decir, si por ejemplo encontramos el Disco en una galaxia, es esperable que a la hora de buscar el Disco Fino y el Disco Grueso, las partículas estelares de ambos estén incluidos en la componente Disco original (o al menos una gran parte de ellas). Esto no es un problema para la componente utilizada

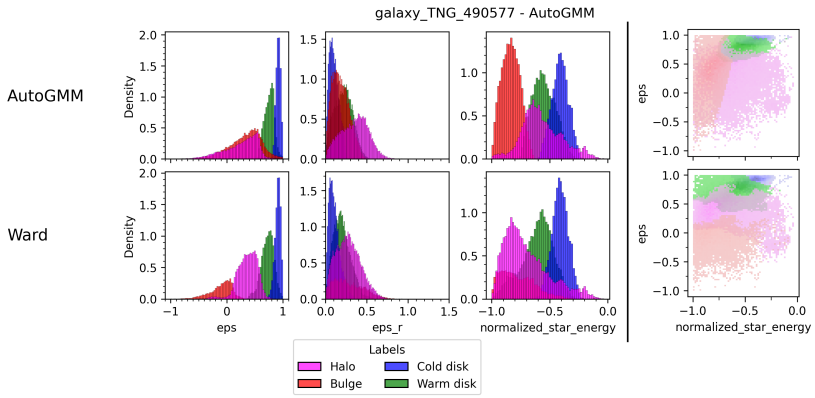

Figura 3.13: Histograma comparando los grupos obtenidos con HC contra AGMM sobre la galaxia *galaxy_TNG_490577* en el espacio circular.

como ejemplo, pero se vuelve un inconveniente a la hora de buscar el Halo y el Bulge dentro del Esferoide.

Dado que el Esferoide es una componente soportada por dispersión de velocidades, sus partículas estelares tiene una alta variación en *eps* (Mo et al., 2010). Como las componentes Halo y Bulge dependen del movimiento que predomina en el Esferoide, los valores de *eps* de sus partículas estelares tendrán una alta variación. Y, por consecuente, ambas componentes se verán "solapadas" en *eps*, como bien puede observarse en la grafica de Auto Gaussian Mixture (AGMM) en la Fig 3.13.

Como HC busca minimizar la varianza de cada cluster, esto se traduce en buscar que los clusters estén lo más separados entre ellos, lo cual va en directa contraposición con la naturaleza de las componentes mencionadas al observar el histograma de *eps*, incluso si dicha heurística tiene sentido dentro del marco teórico del agrupamiento de datos.

Esta distinción entre lo que es más correcto a la hora de buscar clusters abstractos contra buscar componentes de una galaxia se vuelve aún más evidente si revisamos las métricas de Silhouette y Davies-Bouldin en la Fig 3.14 y Fig 3.15 respectivamente. Estas métricas son calculadas a partir del concepto ideal abstracto de cluster, el cual se basa en que éstos sean compactos, separados espacialmente entre ellos, y que exista "continuidad" en cada uno. Los resultados de estas métricas apuntan a que las componentes encontradas por Ward son más correctas que las encontradas por AGMM, cuando esto no se corresponde en realidad con las estructuras de las galaxias.

Esto conlleva a que *eps* no sea una feature útil para Ward a la hora de diferenciar las dos componentes mencionadas. De las dos features restantes, *normalized_star_energy* sería la más prometedora según el orden de relevancia en el área. La importancia de dicha feature se ve reforzada al observar las componentes identificadas por AGMM en la Fig 3.16, en donde se puede ver una clara separación entre el Halo y el Bulge que podría resultar útil para HC. Sin embargo, la existencia de más clusters con partículas estelares "solapando" a ambos el Halo y el Bulge en los mismos valores de *normalized_star_ernergy* conlleva a que Ward se concentre en la clusterización que es posible derivar

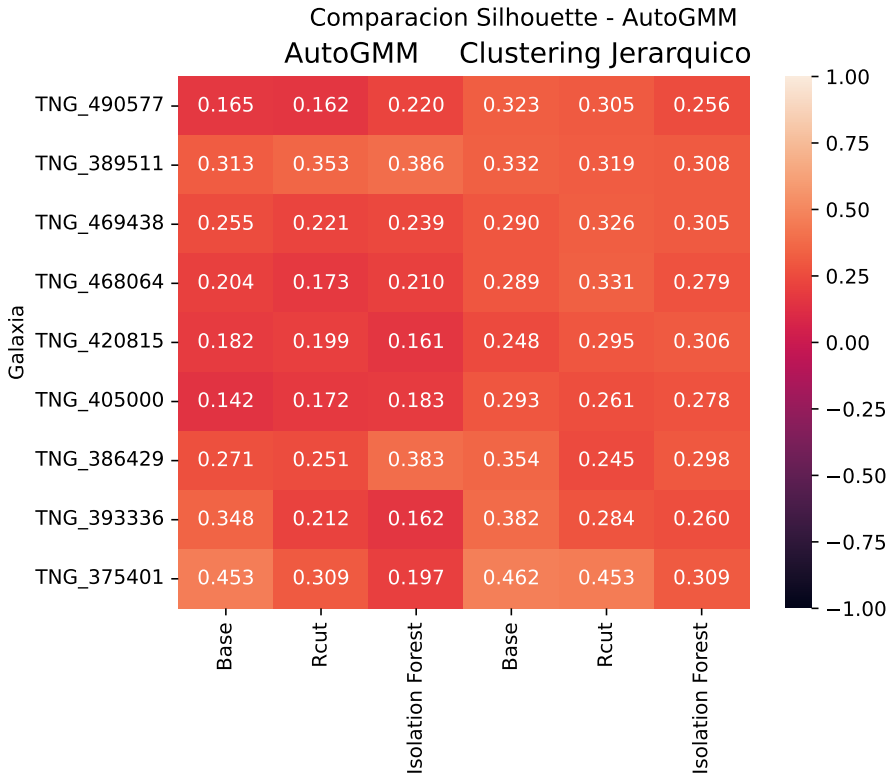

Figura 3.14: Comparación de métrica Silhouette sobre los resultados obtenidos entre AGMM y HC con Ward.

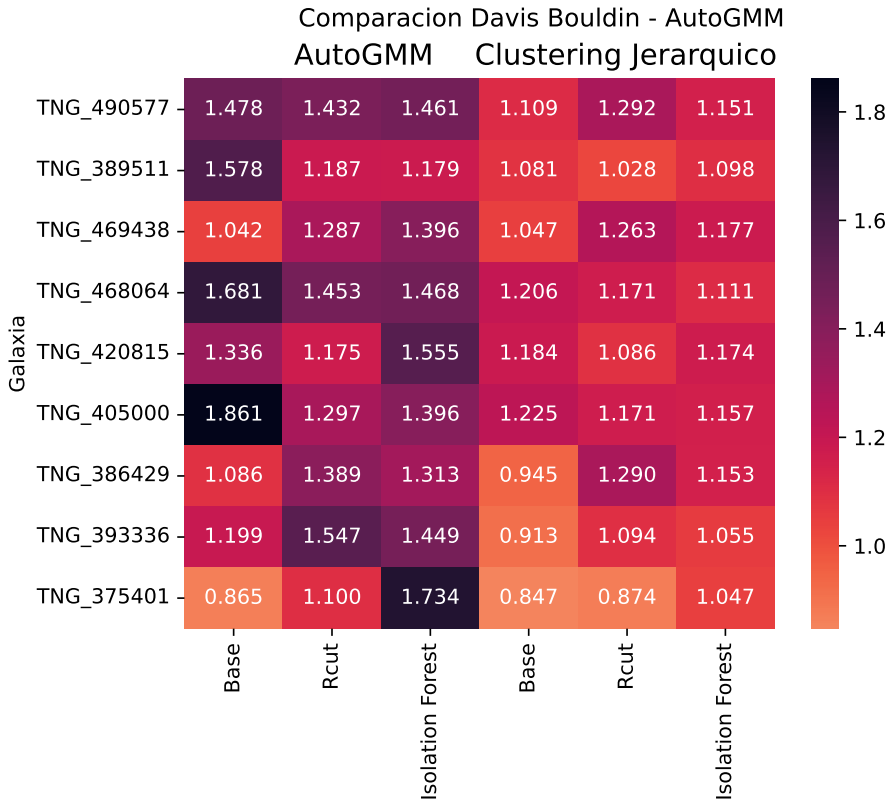

Figura 3.15: Comparación de métrica Davies Bouldin sobre los resultados obtenidos entre AGMM y HC con Ward.

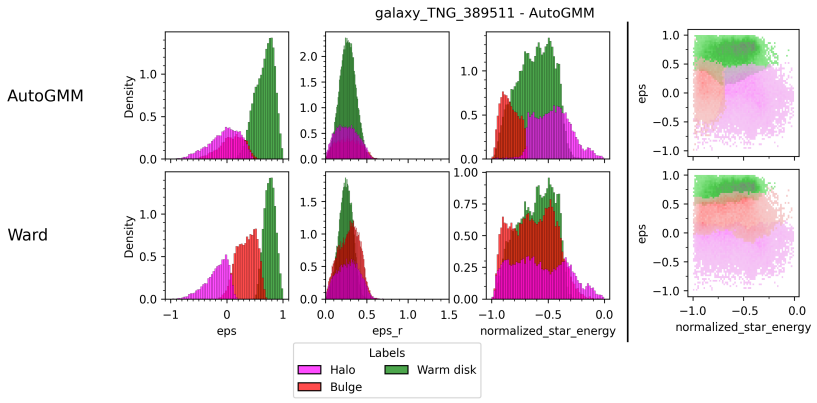

Figura 3.16: Histograma comparando los grupos obtenidos con HC contra AGMM sobre la galaxia *galaxy_TNG_389511* en el espacio circular.

de *eps*. Esto resulta en un agrupamiento casi exclusivamente unidimensional y claramente erróneo, como se puede ver en el scatterplot de la Fig 3.16.

Diferente es, por ejemplo, si se busca el Disco Fino y el Disco Grueso. Como las componentes del Disco son soportadas por rotación, esto significa que predomina el movimiento ordenado (Mo et al., 2010), y las dispersiones en *eps* de ambos grupos son más acotadas en comparación a las componentes del esferoide (con el Disco Grueso en particular teniendo más varianza que el Disco Fino). Esto implica que HC tenga una mayor facilidad a la hora de identificar dichas componentes, como bien puede observarse en la Fig 3.13.

Además, como el promedio de los valores de precisión obtenidos en las 9 galaxias de nuestro conjunto de prueba es de 0.65 (Fig 3.17), mientras que el de recall es de 0.58 (Fig 3.18), podemos afirmar con un alto nivel de certeza que HC no hace un buen trabajo al buscar más de dos componentes. Esto tiene sentido ya que las soluciones del método jerárquico con más clusters son particiones de las soluciones con menos clusters, por lo cual solo pueden aumentar el error al aumentar la cantidad de clusters a buscar.

Similar a como sucedió con Abadi, los resultados tienen una leve tendencia de converger a los obtenidos con AGMM. Aun así, nuevamente, la clusterización obtenida con HC dista de ser tan buena como la obtenida por AGMM, por lo cual este último método sigue siendo preferible.

3.3.1. Eliminación de outliers

Es necesario remarcar que para asignar labels a los clusters encontrados por HC (y los métodos de los siguientes capítulos) utilizamos una heurística que resuelve el "labels map". Es decir, una función que asocia un cluster a un componente de la galaxia. En este caso nuestra asignación automática maximiza el promedio pesado de los valores de Recall de cada conjunto encontrado de entre todas sus posibles permutaciones. Por lo cual, aunque pareciera que en la Fig 3.19 hemos etiquetado de forma errónea al Halo y el Bulge en IF, el método detallado nos provee una base sólida para razonar sobre los resultados.

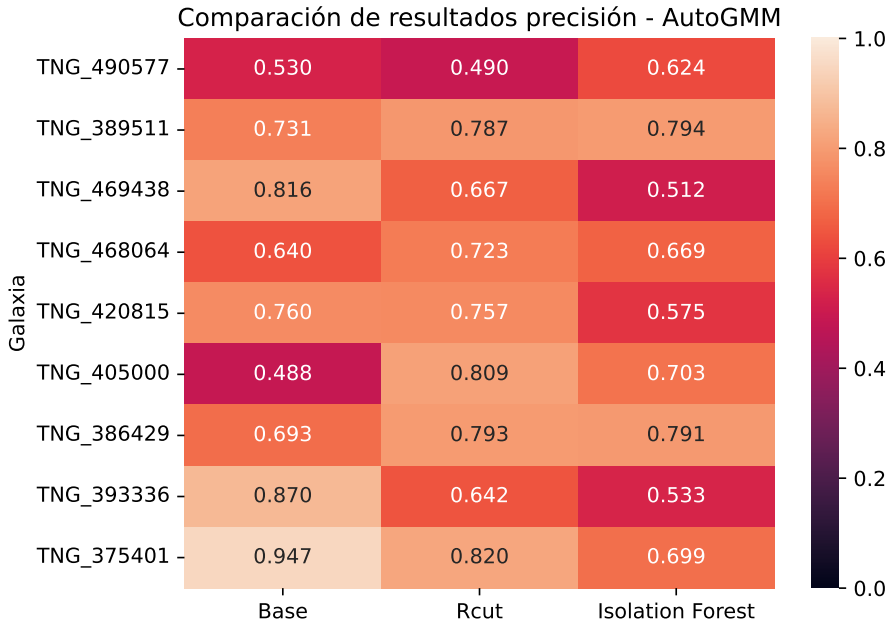

Figura 3.17: Comparación de precisión sobre los resultados obtenidos entre AGMM y HC con Ward.

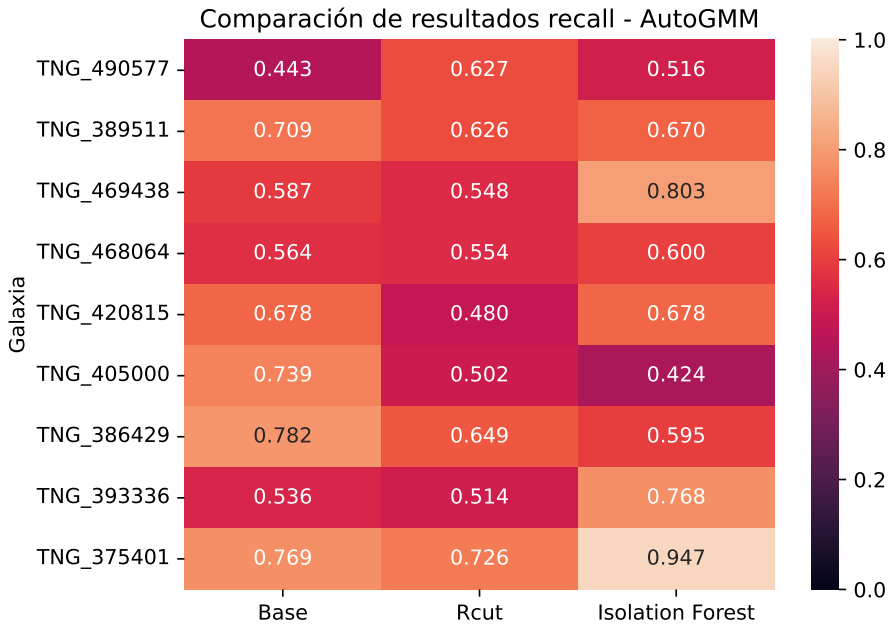

Figura 3.18: Comparación de recall sobre los resultados obtenidos entre AGMM y HC con Ward.

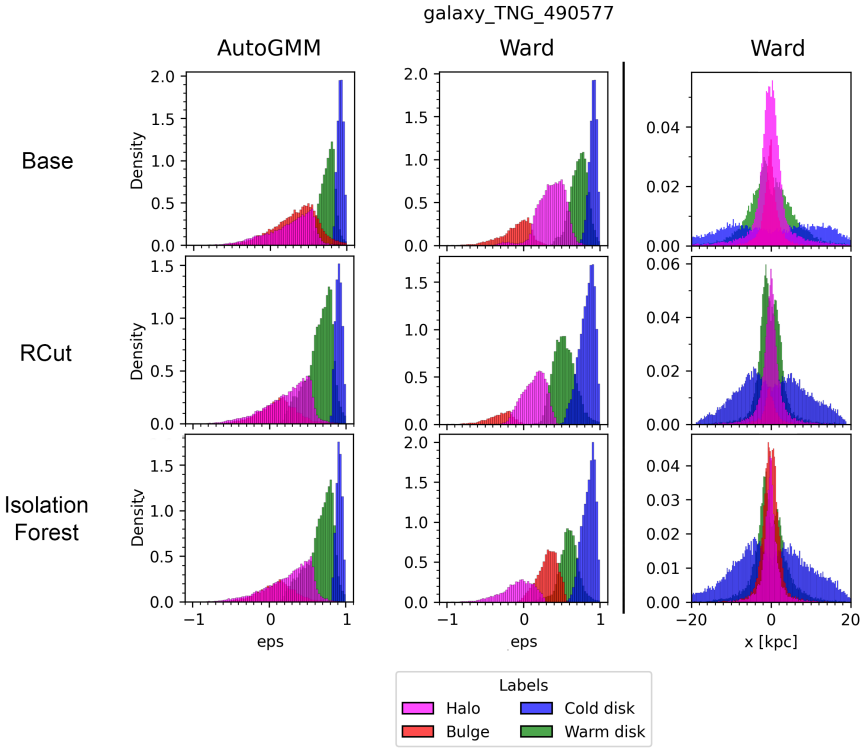

Figura 3.19: Comparación de histogramas en espacio circular y real entre AGMM y HC sin tratamiento de outliers, luego aplicando RCut, y luego aplicando IF sobre la galaxia *galaxy_TNG_490577*.

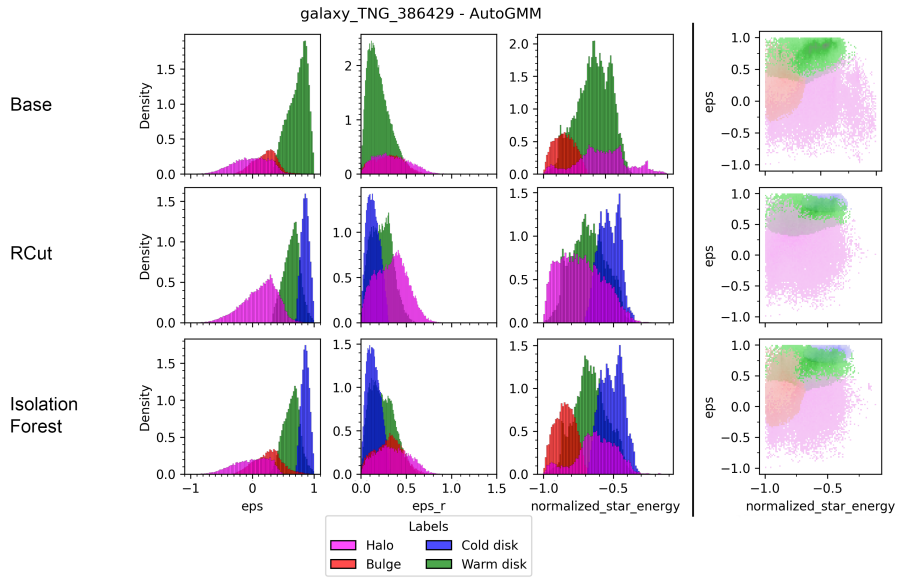

Figura 3.20: Comparación de como AGMM obtiene diferente cardinalidad de componentes al correr diferentes métodos para remover outliers sobre la galaxia *TNG_389511*.

Sin embargo, dado que la cantidad de clusters a buscar es establecida automáticamente por lo encontrado por AGMM, remover partículas estelares con métodos de eliminación de outliers resulta en la variabilidad de dicha cardinalidad, ya que el conjunto de datos con los cuales AGMM trabaja es diferente, resultando así en potencialmente un conjunto de componentes diferentes. Esto puede observarse en la Fig 3.20.

Pueden verse las galaxias cuya cardinalidad de componentes no fue alterada al remover outliers en el Apéndice A, pero dado que éstas son menos de la mitad de nuestro dataset, y que no hay una clara mejora en los resultados obtenidos al aplicar HC al remover outliers sobre éstas (como bien lo ejemplifica la Fig 3.19 y la Fig 3.21), no podemos concluir que remover outliers con RCut o IF sea beneficioso para la clusterización obtenida con HC al buscar más de dos componentes.

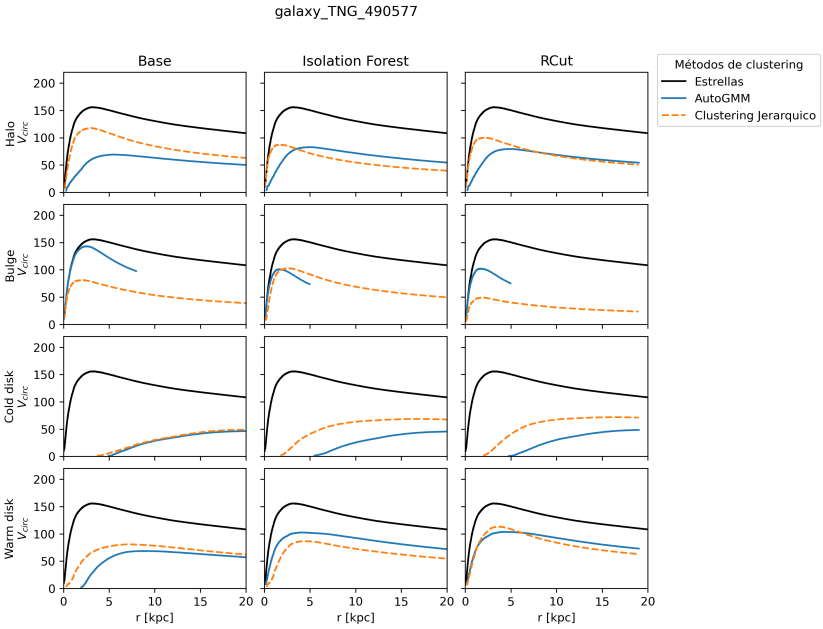

Figura 3.21: Curva de rotación sobre los resultados obtenidos con AGMM y HC sobre la galaxia *galaxy_TNG_490577*, utilizando diferentes métodos de eliminación de outliers.

Capítulo 4

Fuzzy Clustering

En este capítulo estaremos utilizando FCM para encontrar clusters en nuestros conjuntos de datos, usando en particular la implementación proveída por la librería *scikit-fuzzy* en Python (Virtanen et al., 2020).

FCM es un método de clustering en el cual cada punto de los datos puede pertenecer a más de un cluster, mediante la asignación de valores de pertenencia a cada punto de nuestros datos por cada cluster que estemos buscando. Así, puntos cerca del borde de un cluster obtienen un bajo valor de pertenencia.

Formalmente, dado un conjunto de datos $X = (x_1, x_2, ..., x_k)$, si tenemos en total k puntos de datos y c clusters, entonces se define la función de pertenencia como:

$$\mu_{ij} = \mu_i(x_j) \in [0,1] \quad \forall i = 1, ..., c \wedge \forall j = 1, ..., k \tag{4.1}$$

De forma tal que

$$\sum_{i=1}^{c} \mu_i(x_j) = 1 \quad \forall j = 1, ..., k. \tag{4.2}$$

Se define a una partición difusa como la caracterización de la participación de cada muestra en todos los clusters a partir de un valor de pertenencia que se encuentra en el rango $[0, 1]$. Estas particiones difusas se representan con una matriz, asociando cada fila a uno de los c clusters y cada columna a uno de los elementos de X, de forma tal que el valor en la fila i y la columna j indique la pertenencia del elemento j al cluster i. Puede definirse el conjunto de particiones difusas de la siguiente manera:

$$M_{fc} = \{U \in \Re^{c \times n} | U = [u_{ij}]; u_{ij} \in [0,1] \forall i, j; \sum_{i=1}^{c} u_{ij} = 1 \forall j; \sum_{j=1}^{k} u_{ij} > 0 \forall i\} \tag{4.3}$$

Por lo tanto, podemos traducir el problema de FCM a buscar una partición difusa (es decir, una matriz perteneciente al conjunto descrito arriba) óptima.

El algoritmo que utilizaremos para converger en una matriz pertenece al grupo de Fuzzy c-Means, los cuales a su vez forman parte de una clase de algoritmos basados en funciones objetivo. Por lo cual, pasaremos a definir la siguiente función de error mínimo cuadrático la cual el algoritmo minimizara de forma iterativa, para así encontrar la partición difusa óptima:

$$J_m(U, v) = \sum_{j=1}^{k} \sum_{i=1}^{c} (u_{ij})^m d_{ij}^2 \tag{4.4}$$

En donde d_{ij} es la distancia Euclidiana entre el elemento x_j y el centro del cluster i, mientras que $U \in M_{fc}$ contiene todos los pesos u_{ij} asociados a las distancias cuadradas y v representa el vector centro de cada cluster. El valor de m representa qué tan difusa queremos que sea la partición, con valores $m \to 1$ dando como resultado valores de pertenencia más distintivos (obteniendo resultados similares a k-means), mientras que con $m \to \infty$ la partición óptima se aproximará a la matriz en donde todos sus valores son $1/c$.

Una vez definida la función objetivo, el algoritmo se basa en seguir los siguientes pasos:

1. Fijar c, k y m. Elegir una matriz inicial $U^{(0)} \in M_{fc}$
2. Calcular los centros de los clusters como $v_i = \frac{\sum_{j=1}^{k} (u_{ij})^m x_j}{\sum_{j=1}^{k} (u_{ij})^m}; 1 \leq i \leq c$
3. Actualizar la matriz de partición difusa $U = [uij]$ con $u_{ij} = (\sum_{n=1}^{c} (\frac{d_{ij}}{d_{nj}})^{\frac{2}{m-1}})^{-1}; 1 \leq j \leq k; 1 \leq i \leq c$
4. Si se alcanzó la máxima cantidad de iteraciones, o la diferencia de la matriz U con respecto a la iteración anterior es menor al error mínimo dado, el algoritmo termina. Caso contrario, vuelve al paso 2.

Una vez terminado de correr el algoritmo, etiquetamos cada dato de nuestro conjunto con el cluster cuyo valor en la partición difusa sea mayor.

4.1. Detección de dos componentes: Disco y esferoide

Antes de comenzar es necesario entender que utilizar algoritmos de clasificación con una sola feature del dataset no podrá introducir un limite "difuso" entre clusters, ya que los números no tendrán ningún significado para el método, a diferencia de Abadi que les da una interpretación física. Por lo cual, si pedimos al algoritmo que busque clusters en una recta, no va a tener otras dimensiones de las cuales extraer información para entender donde los clusters se pueden solapar, haciendo que éstos siempre tengan una separación clara entre ellos, sin importar el valor de fuzziness utilizado. Esto implica que al mejor resultado que se puede aspirar al comparar este método con Abadi es un corte vertical sobre *eps* entre ambos clusters que se encuentra lo más cercano posible al valor donde la densidad del Disco pasa a ser menor que la del Esferoide en los obtenidos con Abadi.

Curiosamente, mientras buscábamos hiperparámetros, notamos que el valor de "corte" en *eps* que separa las dos clases encontradas se mueve hacia la derecha en la recta al aumentar el valor de fuzziness, como mostramos en la Fig 4.1.

Si observamos la ecuación del segundo paso del algoritmo descrito en la sección anterior, podemos observar cómo aumentar el valor de fuzziness implica

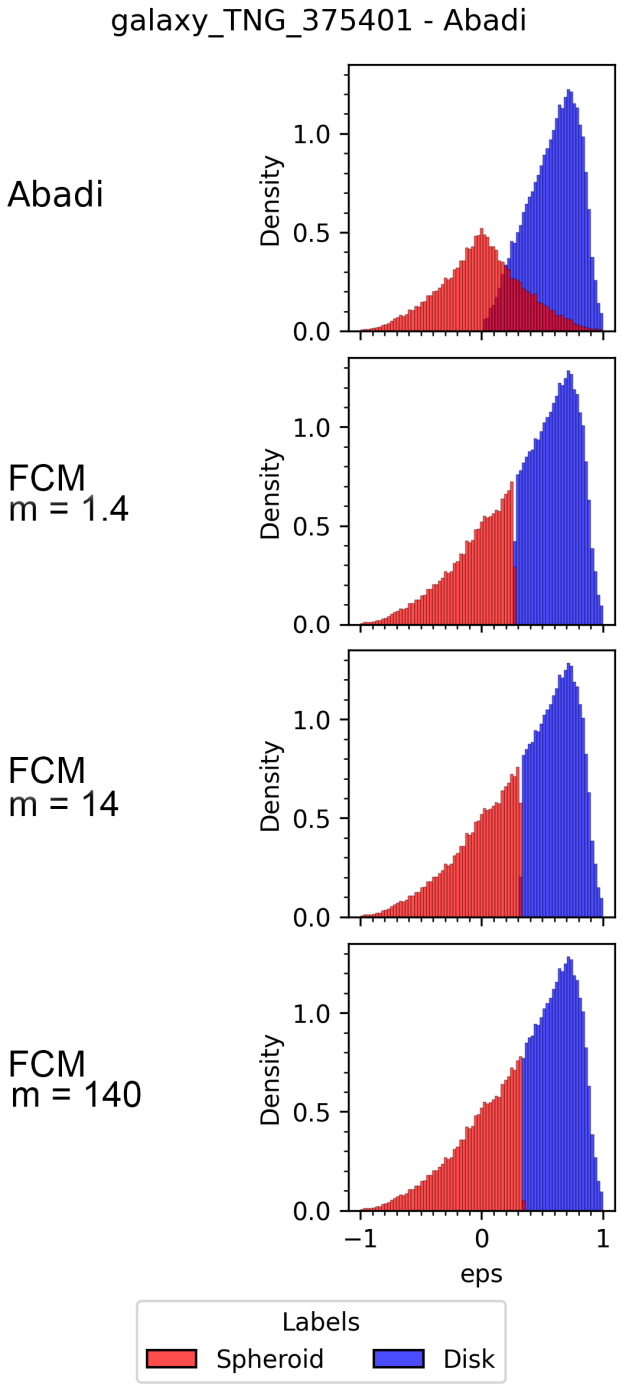

Figura 4.1: Histograma comparando los resultados obtenidos con FCM con diferentes valores de fuzziness sobre la galaxia *galaxy_TNG_375401* en espacio circular.

que el peso que tendrán los valores de pertenencia en la ecuación disminuirá, reduciendo las diferencias entre ellos. Esto asigna similar relevancia a cada punto para cada cluster, haciendo que éstos sean más "heterogéneos", pero este efecto no es justificación suficiente para explicar el comportamiento visto al aumentar el valor de fuzziness.

Al investigar el código de la librería usada para aplicar este método nos encontramos con que los valores de pertenencia de cada punto son representados por `float64`, cuyo mínimo valor representable es $2{,}2250738585072014 \times 10^{-308}$. Esto no parece ser un problema a simple vista, ya que los valores de pertenencia suelen ser mayores, pero si nos detenemos por ejemplo en el primer cuartil de los valores de pertenencia del Esferoide para la galaxia TNG_375401 vemos que éste es $7{,}56 \times 10^{-5}$. Con este valor, basta un valor de fuzziness de 75 para superar el mínimo valor representable por $float64$ en el segundo paso del algoritmo descrito, volviendo los pesos de un cuarto de los puntos del dataset nulos para el centro del Disco.

Por lo tanto, el peso que tendrán los puntos más alejados de cada cluster será nulos para el segundo paso del algoritmo, lo que provoca que las partículas que se encuentran más cercanas al extremo izquierdo de la recta no afectarán al cálculo del centro del cluster del Esferoide. Como la cantidad de partículas pertenecientes a dicha componente cuyo peso dejará de afectar al cálculo del centro del cluster se encuentran mayoritariamente en el extremo izquierdo del cluster (ya que el extremo derecho está cerca del centro, por lo cual sus valores de pertenencias siguen siendo relevantes), entonces se moverá el centro de la componente hacia la derecha, como se muestra en la Tabla 4.1.

Fuzziness	Centro de Esferoide	Centro de Disco
1,4	−0,078	0,616
14	0,008	0,647
140	0,027	0,643

Tabla 4.1: Centros de los clusters encontrados por FCM en *eps* sobre la galaxia *galaxy_TNG_375401* al aumentar el valor de fuzziness.

Sucede un efecto similar en el Disco. Las partículas de éste que se encuentren más a la izquierda en la recta pasaran a tener peso nulo en el cálculo de su centro, derivando en el desplazamiento a la derecha de éste (aunque en menor medida).

Ambas razones resultan en que partículas que se encuentren entre el centro del Esferoide y el Disco en la recta y las cuales antes tenían un valor de pertenencia al Disco mayor que el valor de pertenencia al Esferoide, pasen a formar parte del Esferoide, moviendo la línea de corte hacia la derecha en el proceso. Por lo tanto, entre más grande es el valor de fuzziness, la diferencia entre los puntos que pasaran a no formar parte del cálculo de centros de clusters entre el Esferoide y el Disco se incrementará aún más, moviendo el corte hacia la derecha en la misma medida. Una vez entendido eso, dado el dominio de resultados posibles, pasamos a elegir un valor de fuzziness que acerque la línea de corte al punto promedio en donde la densidad del Disco supera a la del Esferoide en todas las galaxias de nuestro dataset. Luego de recorrer un amplio rango de valores de fuzziness, terminamos optando por 1,4 como valor a utilizar para maximizar los valores de recall promedio de todas las galaxias, como podemos observar en Fig 4.2.

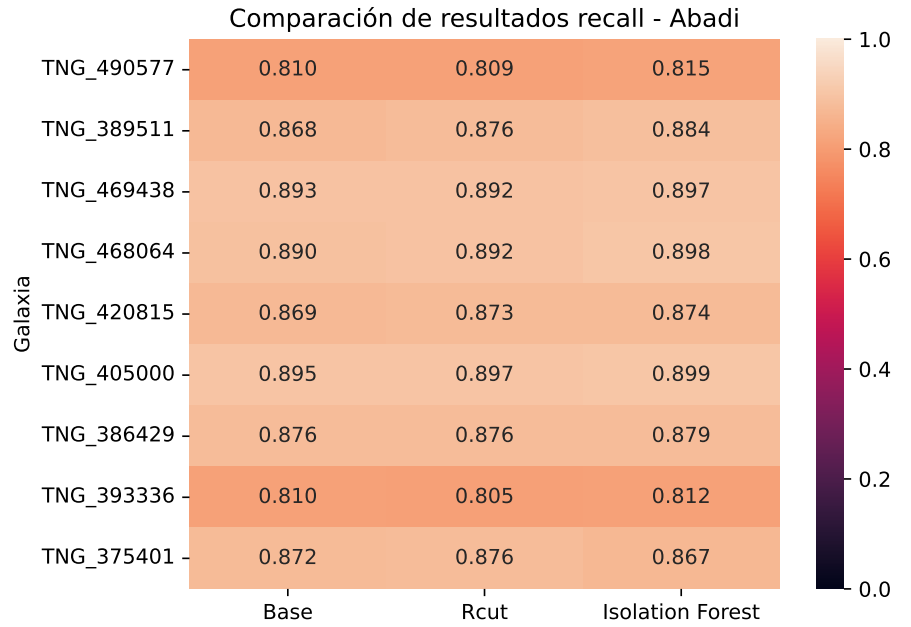

Figura 4.2: Comparación de recall sobre los resultados obtenidos por FCM, considerando a Abadi como ground truth

Además, en las pruebas que hicimos concluimos que fijar el error mínimo en 0,00005 era suficiente para obtener buenos resultados, y disminuirlo aún más no resultaba en una gran diferencia en cantidad de iteraciones. Por su parte, fijamos la cantidad máxima de iteraciones en 1000, aunque el algoritmo siempre terminó a partir de la condición de parada del error mínimo.

Sabiendo cual es el mejor resultado al cual podemos apuntar y como los valores de recall fueron tan positivos, consideramos el resultado del experimento como exitoso y a Fuzzy Clustering como una buena alternativa al algoritmo de Abadi, siempre y cuando se entiendan sus limitaciones.

4.1.1. Eliminación de outliers

Aplicamos métodos de eliminación de outliers con la intención de eliminar posibles partículas que no pertenezcan a la galaxia y así obtener mejores resultados a la hora de clusterizar. Sin embargo, nos encontramos que al aplicar IF y RCut apenas si modifican los conjuntos encontrados de forma perceptible, como parece indicar la Fig 4.2 y la Fig 4.3. Esto también es fácilmente visto en la Fig 4.4.

Esto se debe a que Fuzzy clustering es un método robusto dado que limita el impacto de los outliers al calcular los centros de los clusters (Dave and Krishnapuram, 1997). Por esto, la presencia de outliers no modificará de forma significativa los clusters finales obtenidos, y en consecuencia, removerlos no llevará a conjuntos diferentes.

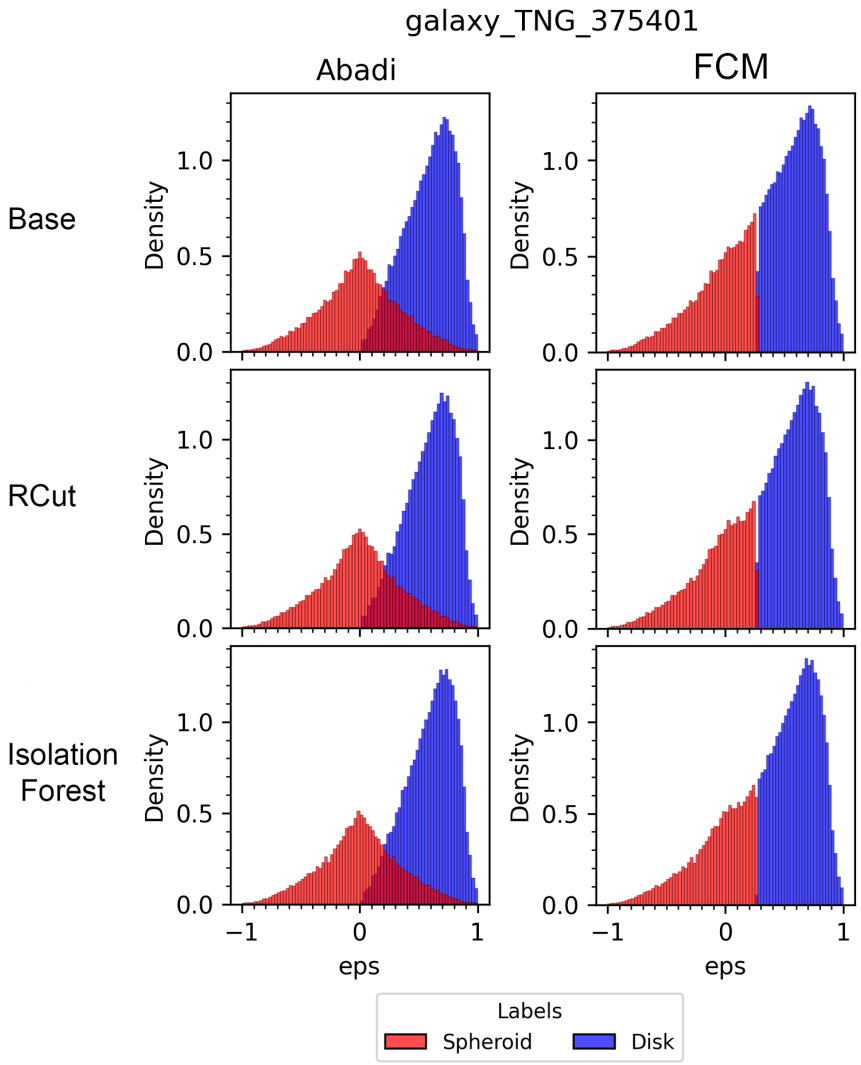

Figura 4.3: Curva de rotación sobre los resultados obtenidos con Abadi y FCM sobre la galaxia *galaxy_TNG_375401*, utilizando diferentes métodos de eliminación de outliers.

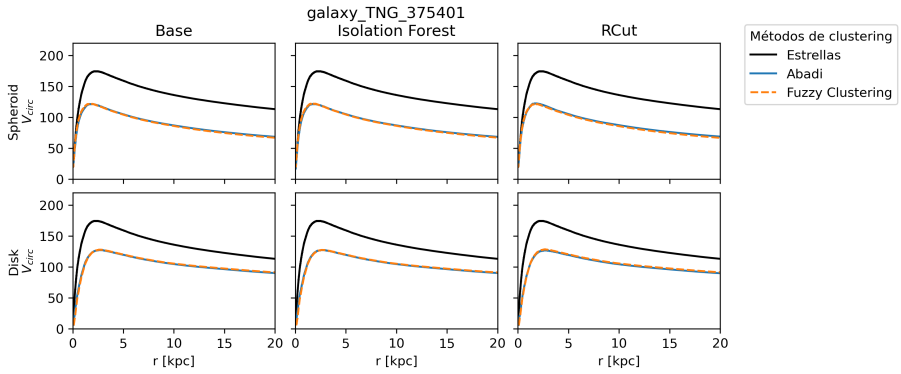

Figura 4.4: Curva de rotación sobre los resultados obtenidos con Abadi y FCM sobre la galaxia *galaxy_TNG_375401*, utilizando diferentes métodos de eliminación de outliers.

4.2. Comparación con Auto Gaussian Mixture - >2 clusters

Al igual que en la sección anterior, antes de comenzar con el experimento hicimos un análisis sobre los hiperparámetros para obtener el valor de fuzziness a utilizar. Pero a diferencia del análisis anterior, la búsqueda de dicho valor tuvo diferentes efectos sobre los resultados a partir de usar más de una feature a la hora de buscar clusters. Esto significa que la variación del valor de fuzziness tuvo el resultado esperado, y al variarlo y comparar los valores de recall obtenidos con los resultados provenientes de utilizar AGMM concluimos que un valor de fuzziness de 3 nos posiciona lo más cerca posible a los resultados obtenidos por este último.

Asimismo, mantendremos los valores de 0,00005 y 1000 utilizados para el mínimo error y la máxima cantidad de iteraciones que dispondrá el algoritmo, usados en la sección anterior.

Cuando comparamos HC contra AGMM en el capítulo anterior mencionamos como el clustering obtenido por el primero era prácticamente unidimensional siguiendo la feature de *eps*. Lamentablemente, los resultados vistos con FCM son similares en este aspecto, como podemos observar en la Fig 4.5.

Creemos que dicho resultado proviene de una "sobre-relevancia" de una feature sobre las otras. Si bien *eps* es sin lugar a dudas la feature que mejor nos permite identificar clusters en nuestro conjunto de datos, *normalized_star_energy* también contiene información importante, sobre todo para identificar el Halo y el Bulge, como puede observarse en el histograma de AGMM en la Fig 4.6.

Si nos detenemos sobre el tercer paso del algoritmo de FCM podemos ver que éste deriva el valor de pertenencia de cada partícula con cada cluster a partir de la distancia de ésta hacia el centro de cada cluster. Sin embargo, observando las distribuciones de las partículas estelares en la Fig 4.7 podemos apreciar los rangos de valores que una partícula puede tener en cada feature; los valores de *eps* tienen un rango $[-1, 1]$, mientras que el rango de los valores de *normalized_star_energy* es $[-1, 0]$ y el de *eps_r* $[0, 1{,}5]$ (aunque a fines prácticos suele ser $[0, 1]$, incluso menor dependiendo la galaxia). Esto implica que, dada

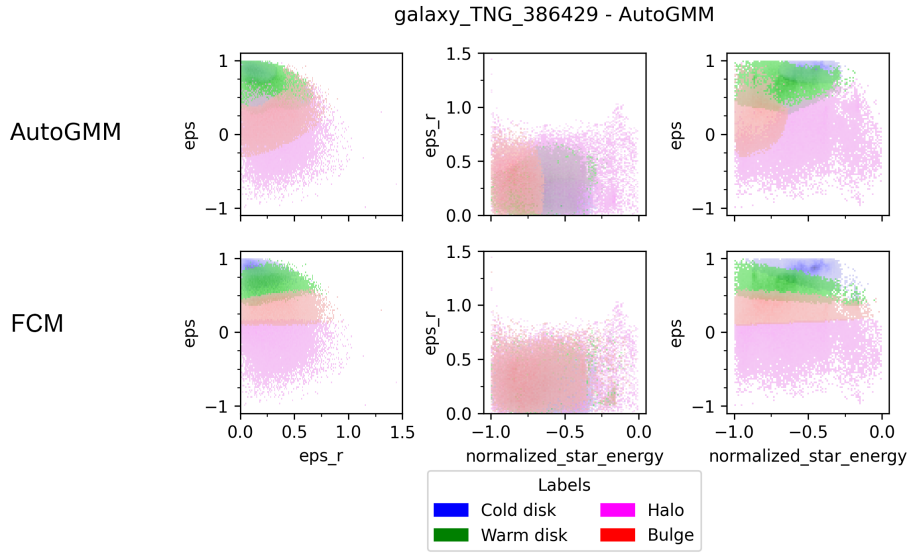

Figura 4.5: Histograma comparando los resultados obtenidos con AGMM contra los obtenidos con FCM sobre la galaxia *galaxy_TNG_386429* en espacio circular.

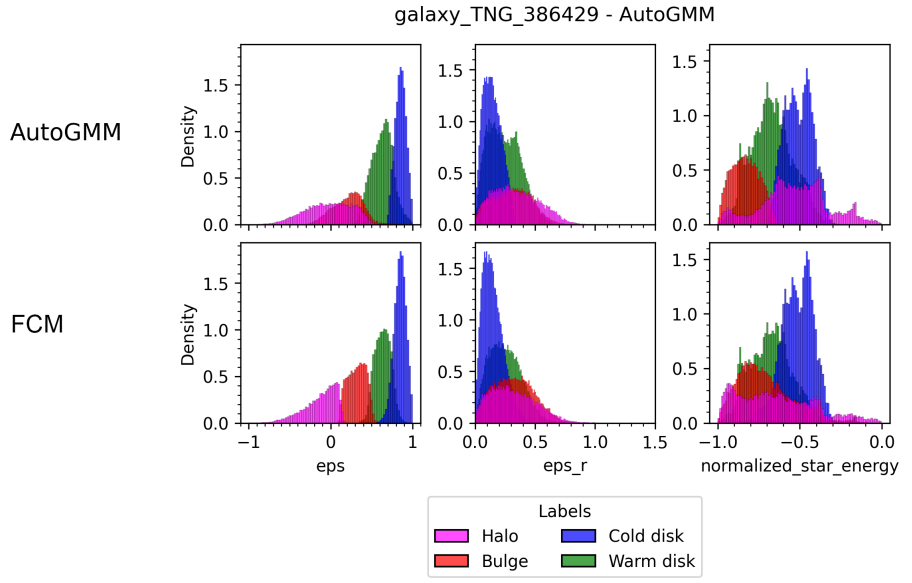

Figura 4.6: Histograma comparando los resultados obtenidos con AGMM contra los obtenidos con FCM sobre la galaxia *galaxy_TNG_386429* en espacio circular.

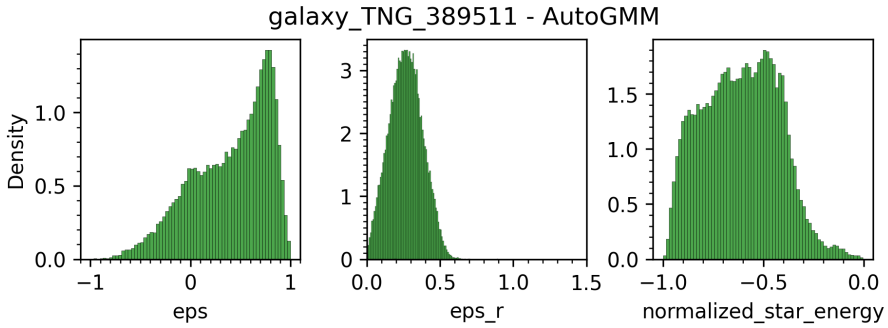

Figura 4.7: Histograma comparando los rangos de valores en las cuales se acumulan las partículas estelares en cada feature del espacio circular de la galaxia *galaxy_TNG_389511*.

una partícula $x = (x.eps, x.normalized\_star\_energy, x.eps\_r)$ y un centro de cluster $c = (c.eps, c.normalized\_star\_energy, c.eps\_r)$, hay altas probabilidades de que la distancia de x a c sea mayormente dominada por la distancia entre $x.eps$ y $c.eps$.

Curiosamente, podemos ver el efecto que tiene *normalized_star_energy* sobre los valores de pertenencia al observar como en la Fig 4.8, en *eps* vs. *normalized_star_energy* no tenemos líneas separadoras entre clusters totalmente rectas sino que son un tanto inclinadas. Dicha inclinación proviene del (leve) aporte que hace la feature de *normalized_star_energy* sobre el valor de pertenencia de las partículas.

Se puede ver de forma aún más clara el poco impacto que tiene *normalized_star_energy* en la clusterización realizada por FCM en la Fig 4.9. En ella observamos como el Bulge y el Halo están solapados en *eps* según el resultado obtenido por AGMM (dado que el Esferoide está soportada por dispersión de velocidades, como se explicó en el capítulo anterior), pero son fácilmente distinguibles en *normalized_star_energy*. Además, se puede ver el nulo aporte de *eps_r* al clustering al observar los resultados de FCM en el scatterplot de esta feature vs. *eps* en la Fig 4.8 por lo rectas que son las delimitaciones entre las componentes encontradas.

Vale la pena destacar que, aunque las componentes encontradas por FCM difieran de los obtenidos por AGMM, los centros de los clusters encontrados por el primero, mostrados en la Tabla 4.2, son bastante precisos y similares a lo observable con AGMM en la Fig 4.6.

	center.eps	center.eps_r	center.normalized_star_energy
Halo	−0,066	0,312	−0,667
Warm Disk	0,632	0,241	−0,652
Bulge	0,320	0,308	−0,703
Cold Disk	0,831	0,151	−0,519

Tabla 4.2: Features de los centros de los clusters encontrados por FCM sobre la galaxia *galaxy_TNG_386429*.

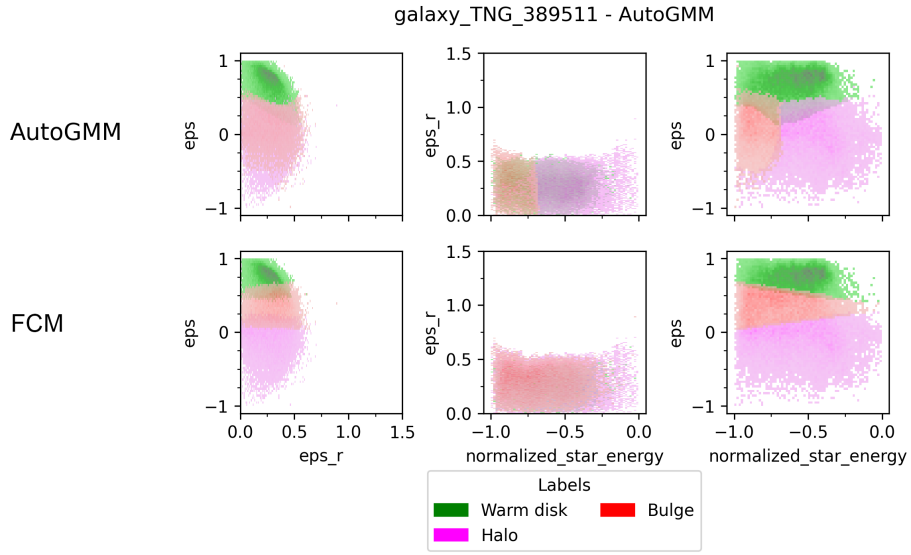

Figura 4.8: Histograma comparando los resultados obtenidos con AGMM contra los obtenidos con FCM sobre la galaxia *galaxy_TNG_389511* en espacio circular.

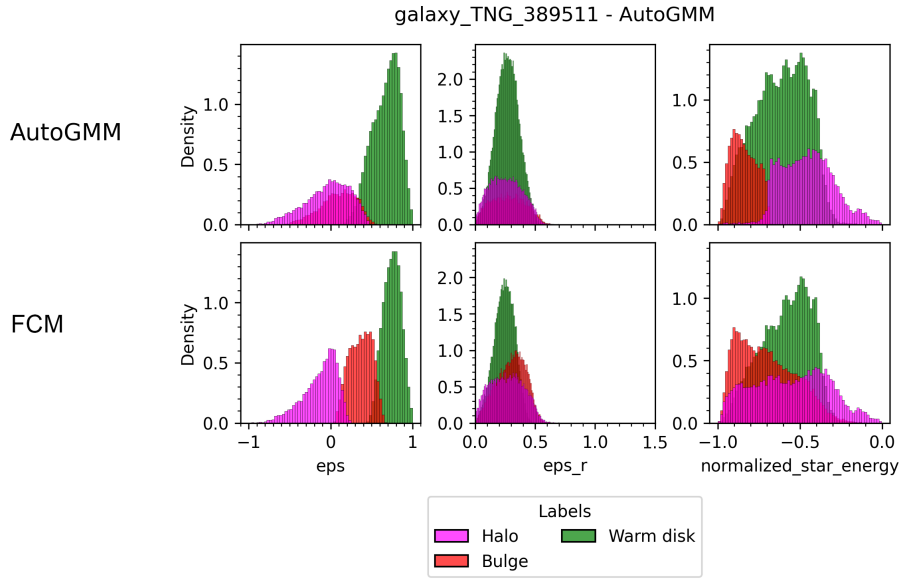

Figura 4.9: Histograma comparando los resultados obtenidos con AGMM contra los obtenidos con FCM sobre la galaxia *galaxy_TNG_389511* en espacio circular.

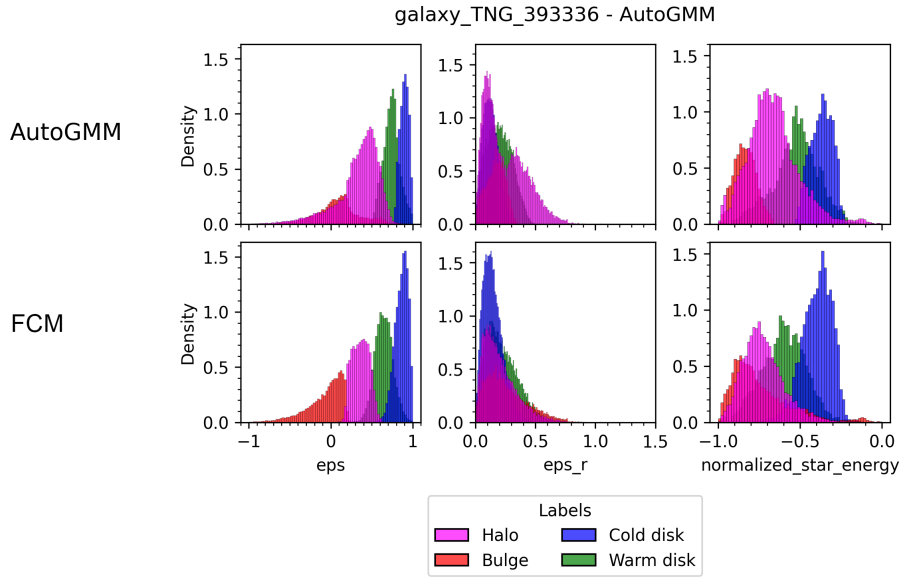

Figura 4.10: Histograma comparando los resultados obtenidos con AGMM contra los obtenidos con FCM sobre la galaxia *galaxy_TNG_393336* en espacio circular.

Algo que también notamos fue la rápida transición que realiza FCM entre el Bulge y el Halo. La razón detrás de esto es similar a lo explicado sobre el rango total que cada feature puede cubrir. Si una partícula está a una distancia similar entre dos centros de clusters en cierta feature (*normalized_star_energy* en este caso), entonces el peso que tendrán las distancias sobre esta feature a cada centro para calcular el valor de pertenencia de la partícula a cada cluster será similar entre ellas. Por esto las distancias en *eps* serán las que definan los valores de pertenencia a cada componente sobre la feature mencionada, resultando de esta manera en el corte limpio entre ambas componentes. Dicho concepto es una continuación de lo visto en la sección anterior en la cual solo utilizamos *eps* para clasificar.

Aunque esta idea está presente en todo nuestro dataset, se puede visualizar más claramente a partir de la Fig 4.10 y la Tabla 4.3, y lo cerca que los centros de las componentes Bulge y Halo están entre ellos en *normalized_star_energy*.

	center.eps	center.eps_r	center.normalized_star_energy
Bulge	0,005	0,229	−0,764
Halo	0,368	0,197	−0,713
Warm Disk	0,632	0,206	−0,586
Cold Disk	0,857	0,145	−0,399

Tabla 4.3: Features de los centros de los clusters encontrados por FCM sobre la galaxia *galaxy_TNG_393336*.

Además, al observar el histograma en *eps* en Fig 4.10 también descubrimos una leve tendencia de FCM a sobreestimar el Cold Disk (Disco Fino) por sobre el Warm Disk (Disco Grueso) con respecto a AGMM.

Como el caso descrito previamente, la transición entre el Warm Disk y el Cold Disk encontrada por FCM que podemos observar en dicho histograma tiene su origen en qué tan separados están los centros de ambas componentes en las features restantes, $normalized\_star\_energy$ y eps_r (aunque el aporte de esta última a la clasificación sea mínimo, como se explicó arriba). Dado que la distancia entre ambas componentes es significativa en $normalized\_star\_energy$, como se puede ver en la Tabla 4.3 (y a raíz del valor de fuzziness utilizado), existirá un solapamiento entre ambos clusters en el histograma de *eps* en la Fig 4.10.

Podemos separar dicho solapamiento en dos zonas. La primera es la que se encuentra del Warm Disk yendo hacia el Cold Disk, y la segunda la que proviene de Cold Disk hacia el Warm Disk. Si comparamos la primera con la clusterización obtenida por AGMM en el mismo sector, dado que la transición entre ambos es menos difusa, entonces habrá una sobre estimación del Cold Disk sobre el Warm Disk en este sector. En la segunda zona, aunque AGMM también tenga una transición fuzzy, la densidad de Warm Disk es mayor a la que encuentra FCM, obteniendo como resultado nuevamente una sobre estimación del Cold Disk sobre el Warm Disk. Dicho comportamiento puede ser visto en la mayoría de las galaxias de nuestro dataset en las cuales encontramos estas componentes.

4.2.1. Eliminación de outliers

Al igual que en el capítulo anterior, eliminar outliers antes de generar clusters con AGMM puede resultar en un diferente grupo de componentes encontradas. Por esto reduciremos nuestro análisis a los casos en los cuales dicho conjunto de componentes se mantenía constante.

Sin embargo, como explicamos en la sección al comparar con Abadi, dada la robustez de FCM, tampoco obtuvimos diferencia al remover outliers con RCut o IF como puede observarse en la Fig 4.11.

Para terminar, dado los resultados sub óptimos de las métricas de recall y precision (en promedio 0.626 y 0.698 respectivamente en nuestro dataset) concluimos que el método no es una buena alternativa a AGMM al buscar más de dos componentes.

Sin embargo, dados los problemas explicados en esta sección, sería interesante en un trabajo futuro realizar una normalización sobre *eps* y eps_r previo a la clusterizacion de FCM para promediar el peso que cada feature tiene sobre el conjunto de clusters finales.

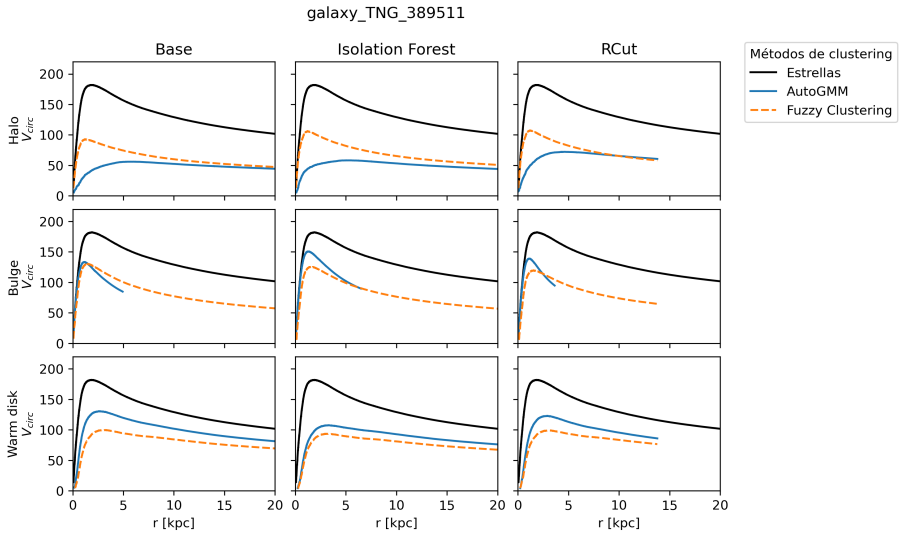

Figura 4.11: Curva de rotación sobre los resultados obtenidos con AGMM y FCM sobre la galaxia *galaxy_TNG_389511*, utilizando diferentes métodos de eliminación de outliers.

Capítulo 5

Evidence Accumulation Clustering

Los métodos de clustering ensambles se basan en tomar resultados generados por múltiples algoritmos de clustering y llegar a un consenso que sea más correcto que los resultados individuales de los algoritmos utilizados. Uno de estos métodos, que utilizaremos en este capítulo, es EAC (Fred and Jain, 2005).

Sea $X = x_1, x_2, ..., x_n$ nuestro conjunto de datos donde $x_i \in R^d$ y R^d es un espacio de d features. Luego, utilizando un método de clustering especifico se obtiene una partición P la cual encuentra k clusters en el conjunto mencionado. Dado N algoritmos de clustering podemos entonces derivar N particiones P las cuales definen los clusters encontrados por cada algoritmo. Por último, definimos al conjunto de estas particiones como:

$$\mathbb{P} = \{P^1, P^2, ..., P^N\} \tag{5.1}$$

$$P^1 = \{C_1^1, C_2^1, ..., C_{k_1}^1\} \qquad \dots \qquad P^N = \{C_1^N, C_2^N, ..., C_{k_N}^N\} \tag{5.2}$$

donde C_j^i es el cluster jth en la partición P^i, la cual tiene k_i clusters. Además, dado n_j^i como la cardinalidad de C_j^i, se cumple que $\sum_{j=1}^{k_i} n_j^i = n, \quad i = 1, ..., N$.

Utilizando la información disponible en las N particiones de nuestros datos en $\mathbb{P}$, el objetivo de EAC es obtener una partición de datos "óptima" llamada P^*. Adicionalmente, definimos k^* como el número de clusters en P^*.

Idealmente, P^* debe cumplir con las siguientes propiedades:

1. Consistencia con el ensamble de clusterings $\mathbb{P}$: Significa que P^* sea el resultado de un acuerdo entre las diferentes partes encontradas $P^i, i = 1, ..., N$.

2. Robustez antes pequeñas variaciones en $\mathbb{P}$: Los cantidad de clusters y los puntos pertenecientes a cada cluster en P^* deberían ser invariantes ante pequeñas perturbaciones de $\mathbb{P}$.

3. Ajuste con respecto al GT: Los clusters encontrados deberán coincidir fuertemente con las verdaderas etiquetas de nuestros datos.

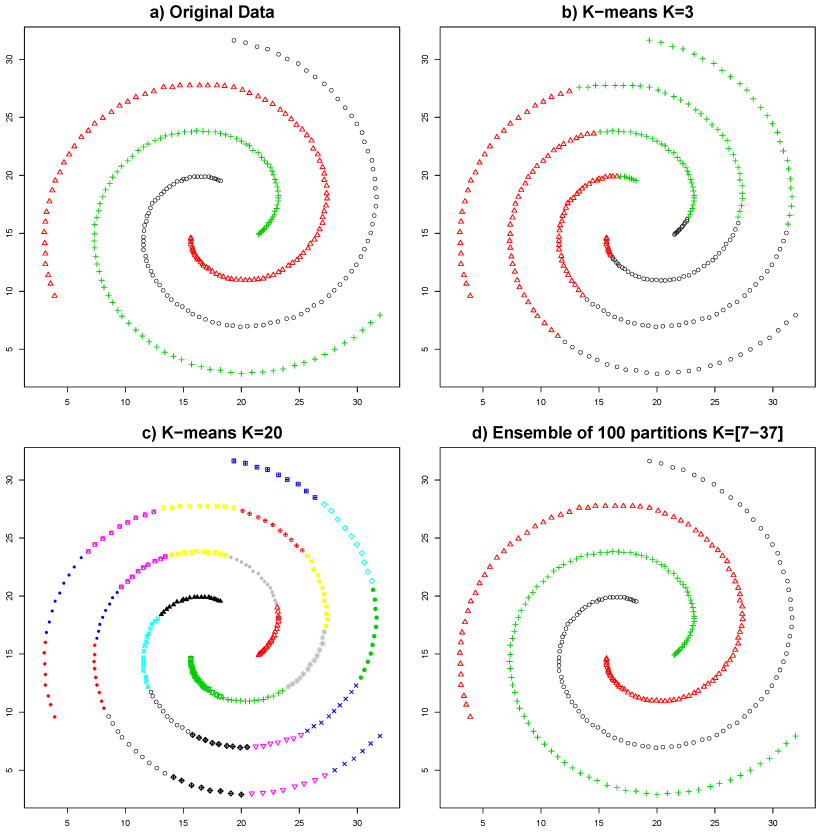

Figura 5.1: Comparación de clustering aplicado a un dataset entre diferentes corridas de k-means contra EAC con 100 instancias de k-means. Variando la cantidad de clusters buscados por estos entre 7 y 37 (Márquez et al., 2019).

La idea detrás de EAC es combinar los resultados de múltiples clusterings en una única partición de datos, interpretando cada uno de estos resultados como una "evidencia independiente" de la organización de los datos, como puede apreciarse en la Fig 5.1. Esto nos requiere abordar los siguientes tres problemas, que definen cada etapa del algoritmo de EAC:

1. Cómo generamos el ensamble de particiones.
2. Cómo combinamos la evidencia.
3. Cómo extraemos una partición de datos (conjunto de clusters) consistente de la evidencia combinada.

Hay diferentes formas de abordar el primer problema, las cuales pueden separarse de las siguientes categorías:

1. Elección de representación de datos:
   a) Empleando diferentes métodos de preprocesamiento y/o extracción de features, lo que resulta en diferentes representaciones de los datos o espacio de features.

b) Explorando subespacios de la misma representación de datos.

c) Perturbando los datos con métodos de bootstrapping o sampling.

2. Elección de algoritmos de clustering o parámetros de los algoritmos:

   a) Aplicando diferentes algoritmos de clustering.

   b) Usando el mismo algoritmo pero con diferentes parámetros o inicializaciones.

   c) Explorando diferentes medidas de disimilitud para evaluar relaciones en nuestros datos.

En este trabajo nos interesara aplicar lo mencionado en $2b$. Es decir, utilizaremos un algoritmo en particular, k-means, múltiples veces con semillas diferentes para generar el ensamble de clusters.

El método de clustering k-means (MacQueen et al., 1967) es un algoritmo clásico del área de minería de datos. Este propone, dado un conjunto de k clusters buscados, asignar de forma aleatoria los centros de dichos clusters e iterativamente optimizarlos, asignando los puntos al centro del cluster más cercano en cada iteración. Como el algoritmo es naturalmente inestable significa que podemos utilizar la variabilidad de múltiples corridas de k-means para llegar a un consenso, como se propone realizar con EAC.

Cabe aclarar que EAC no asume el número de clusters k_i para cada partición obtenida P^i, por lo que cada k_i pueden ser no sólo diferentes al número de clusters buscado por EAC, sino que también entre sí.

Una vez obtenido el ensamble de particiones, pasaremos a combinar la información proveída de estos con una matriz de co-asociación como puede observarse en la Fig 5.2, la cual nos permitirá relacionar la información obtenida con cada partición incluso si la cardinalidad varía entre ellas. Dicha matriz, también llamada matriz de similitud, es de dimensión $n \times n$ y representa qué tan similares son los objetos de un conjunto entre sí (en nuestro caso, los puntos de nuestros datos), asignándole un valor de cero a uno a cada una de sus celdas.

El algoritmo propone un mecanismo de votación que será utilizado como métrica de similitud entre los puntos de nuestros datos, bajo la suposición de que dos puntos pertenecen al mismo cluster si se encontraron en el mismo cluster repetidas veces en el ensamble de particiones. Dicha métrica es la siguiente:

$$C(i, j) = \frac{n_{ij}}{N},$$

donde n_{ij} es el número de veces que el par de puntos (i, j) son asignados al mismo cluster en las N particiones. Por lo cual, asignaremos el valor de $C(i, j)$ a la celda de la matriz de similitud de la fila i y la columna j.

Una vez generada la matriz de similitud podemos aplicar cualquier algoritmo de clustering sobre ésta. Fred and Jain (2005) optan por utilizar clustering jerárquico aglomerativo con single o average linkage.

Para correr dicho algoritmo en este trabajo utilizaremos la implementación realizada por la librería de Python "combo"[1] (Zhao et al., 2020). Aunque el

[1]Fue necesario crear un fork de la librería para poder solucionar un bug y realizar optimizaciones generales de la memoria utilizada por su implementación de EAC, como así optimizaciones extra específicas para nuestro campo de estudio. Se puede encontrar en https://github.com/originalnicodr/combo.

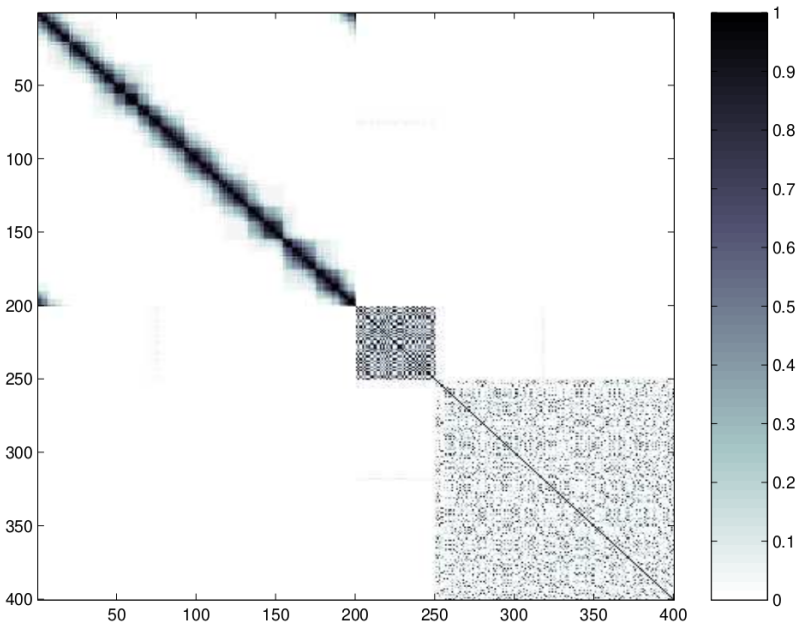

Figura 5.2: Matriz de asociación obtenida con EAC (Fred and Jain, 2005), en donde ambos ejes representan puntos en un conjunto de datos, y cada celda que tan similares son estos entre ellos. Con los valores más altos indicando mayor similitud.

paper original describe cómo la elección de donde cortar el dendrograma se realiza a partir de una medida de "tiempo" respecto a qué tanto se mantienen los clusters en el dendrograma antes de juntarse con otro, la librería mencionada nos da la opción de cortar el dendrograma en donde queramos para así obtener una partición de nuestros datos con la cantidad específica de clusters que buscamos. Esto nos es de especial utilidad para realizar los experimentos y comparaciones que venimos haciendo en los capítulos anteriores.

5.1. Detección de dos componentes: Disco y esferoide

En este trabajo utilizamos el algoritmo descrito de EAC con HC y single linkage. El algoritmo que decidimos utilizar fue k-means con $k_i = 100, i = 1, ..., N$.

Aunque comenzamos utilizando $N = 500$ k-means con inicializaciones al azar, el costo computacional de EAC es tan grande para nuestro espacio de datos que nos vimos obligados a utilizar $N = 100$ en su lugar, con los resultados variando poco entre ellos como se puede observar en la Fig 5.3

Lamentablemente, el costo computacional asociado a utilizar 100 corridas en lugar de 500 sigue siendo alto, lo cual nos impidió realizar todos los experimentos planeados como hicimos en los dos capítulos anteriores en un tiempo razonable. Sin embargo, los resultados que si alcanzamos a obtener con las corridas que hicimos están tan alejados del GT que dicha falta de experimentos no son de gran impacto en nuestro análisis.

Como ya se pudo observar en la Fig 5.3, las clasificaciones obtenidas distan mucho de las reales. En la figura mencionada también se puede apreciar cómo la gran mayoría de las partículas estelares se concentra en uno de los dos clusters. Para entender por qué sucede esto debemos pasar nuestra atención al comportamiento de cada k-means individual aplicado sobre la galaxia.

En la Fig 5.4 se puede ver cómo los clusters parecen estar separados en líneas siguiendo *eps*. La razón de esto parece ser que, dado que la columna *eps* es la única fuente de información desde la cual k-means puede derivar clusters, cada partícula estelar será etiquetada con el centro del cluster más cercano en el eje *eps*. Y, de forma similar a como sucedió en el capítulo anterior, esto culmina en clusters divididos por líneas perpendiculares al eje *eps*, ya que no tienen otras columnas o datos desde los cuales derivar clusters y aumentar la dimensionalidad de estas divisiones.

Luego de realizar las 100 corridas diferentes de k-means que aplica EAC, dado que la clusterización está siendo realizada sobre un solo eje, a medida que avanza la aglomeración del HC que se realiza en el último paso del algoritmo, se irán uniendo clusters adyacentes en el eje *eps* hasta obtener los últimos dos, separados por una línea recta en dicho eje.

Dado que estamos utilizando single-linkage en el segundo paso de EAC, como vimos en el capítulo de Clustering Jerárquico, la métrica de disimilitud entre dos clusters es la menor distancia posible entre dos puntos de ambos. En este caso, el mejor resultado al que el método podía aspirar (dividir verticalmente los datos en *eps*) hubiera requerido que las partículas alrededor del punto donde la densidad del Disco supera a la del Esferoide en Abadi no hayan compartido

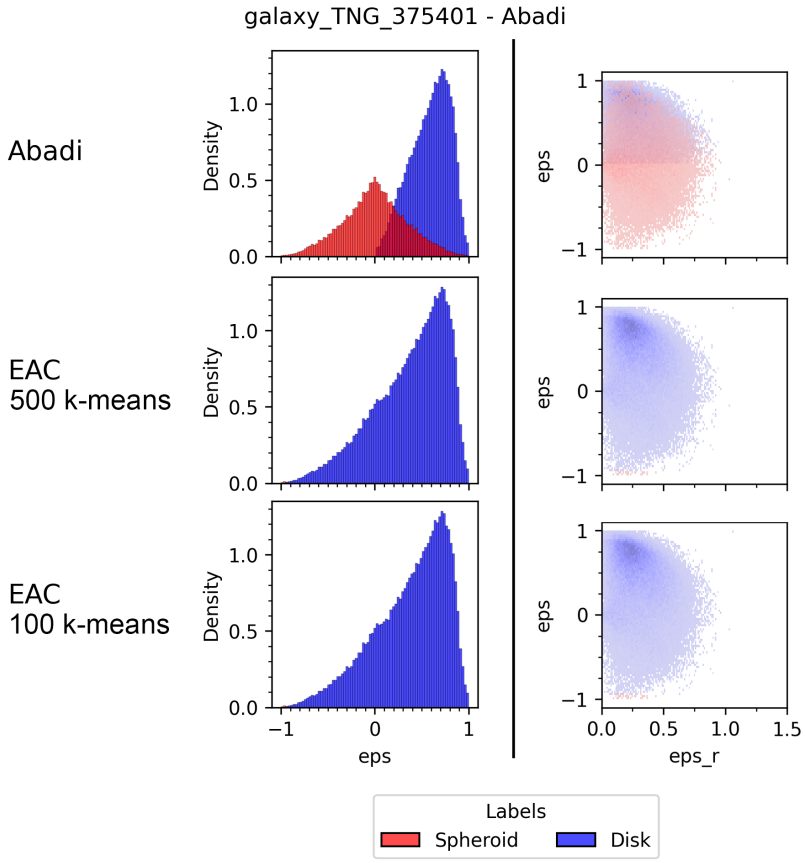

Figura 5.3: Comparación EAC con 500 k-means contra EAC con 100 k-means sobre la galaxia *galaxy_TNG_375401*.

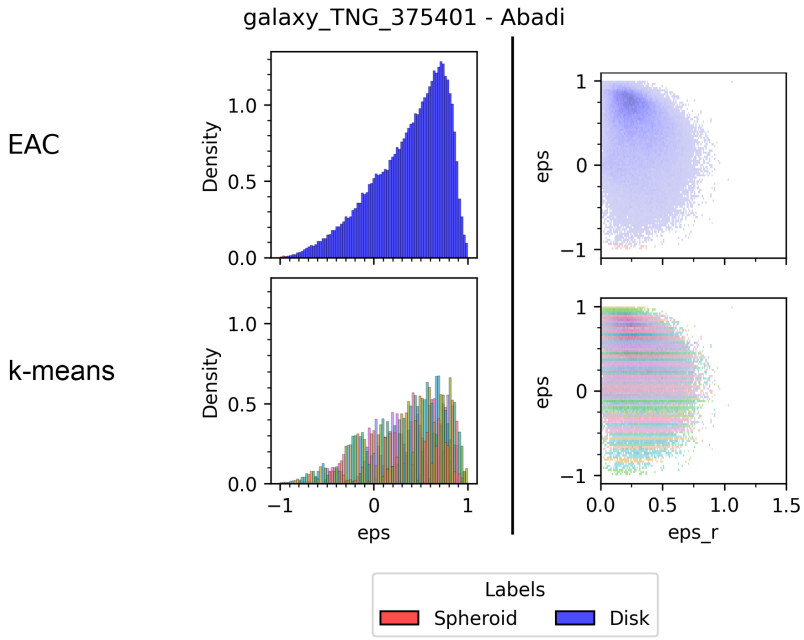

Figura 5.4: Comparación EAC contra una sola instancia de k-means sobre la galaxia *galaxy_TNG_375401*.

muchos clusters. Pero, dado lo cercanos que están todos los puntos en *eps*, EAC tenderá a priorizar fusionar los clusters que se encuentran en los lugares de mayor densidad del conjunto de datos, ya que al aumentar la densidad de los puntos las probabilidades de encontrar clusters con métricas de disimilitud pequeñas se incrementa al utilizar single-linkage. Es por esto que el Disco encontrado por EAC es tan pequeño en comparación al encontrado por Abadi y se encuentra en un extremo de la Fig 5.3. Alternativamente, podemos ver el mismo efecto en el otro extremo de *eps* en la Fig 5.5.

Nuevamente, lo visto en la Fig 5.5 es resultado de nuestra heurística de calcular la métrica de recall comparando las clasificaciones obtenidas entre EAC y Abadi para cada posible combinación de etiquetas en EAC y así quedarnos con el conjunto de etiquetas que maximicen el valor de recall. Dado que la diferencia de partículas entre los dos clusters encontrados por EAC es tan masiva, la heurística determinó que asignar el cluster encontrado más grande con el cluster de mayor densidad de Abadi era lo más acertado.

Por último, dada la vasta diferencia de los clusters obtenidos con EAC comparado con los obtenidos con Abadi y la explicación de por qué sucede esto detallada anteriormente, determinamos que no vale la pena dedicar el tiempo ni el costo computacional necesario para hacer los mismos experimentos eliminando outliers.

Concluimos que EAC utilizado con k-means no es una buena alternativa a Abadi, por el costo computacional y de memoria, como así también los resultados deficientes mostrados.

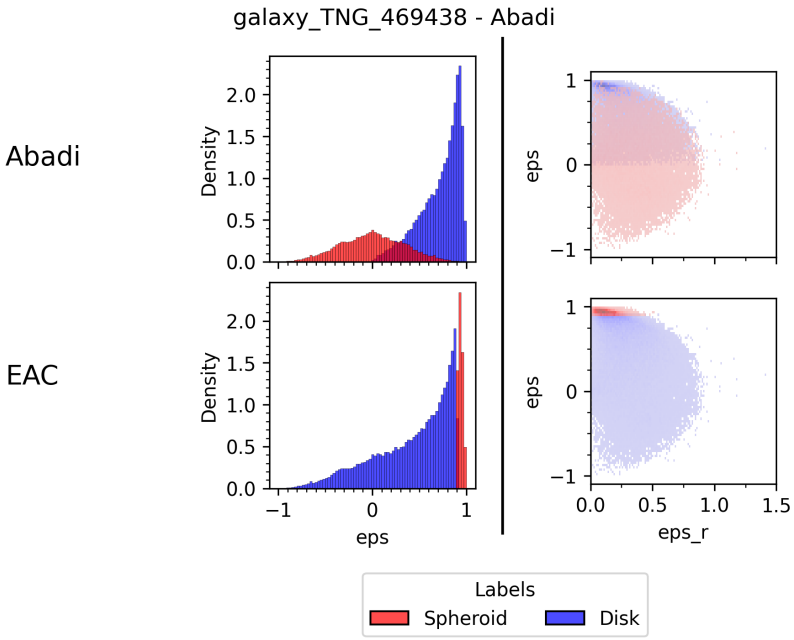

Figura 5.5: Comparación Abadi contra EAC sobre la galaxia *galaxy_TNG_469438*.

5.2. Comparación con Auto Gaussian Mixture - >2 clusters

Nuevamente, utilizamos $N = 100$ para realizar los experimentos con EAC y poder comparar sus resultados contra los obtenidos con AGMM. Además del alto costo computacional mencionado en la sección anterior, los resultados fueron aún más desalentadores, como puede observarse en la Fig 5.6.

Dado que la limitación sobre las features trabajadas no es un problema en este experimento (ya que utilizamos las tres presentadas al comienzo de este trabajo al tener que comparar contra los resultados obtenidos por AGMM), podemos asociar el mal desempeño de EAC por el valor de k elegido para los k-means utilizados dentro de éste. Utilizar valores tan altos de k (es decir $k_i = 100, i = 1, ..., N$ como ya habíamos mencionado) puede derivar en una sobrepartición de los k-means, lo que dificulta encontrar los patrones buscados en el dataset. El resultado es una matriz de consenso ruidosa y que no captura correctamente las co-ocurrencias de los puntos.

Potencialmente, como se sugiere en uno de los experimentos realizados en Fred and Jain (2005), variar los valores de k utilizados por cada k-means (dentro de las cotas adecuadas) en una misma corrida de EAC es más robusto que utilizar un k fijo. En dicho trabajo, se argumenta que usar un rango amplio de valores puede llegar a tener un efecto de promediado que nos lleve a la identificación de la estructura verdadera de nuestros datos. Lamentablemente, como bien se explicó en la sección anterior, dados los grandes costos computacionales de

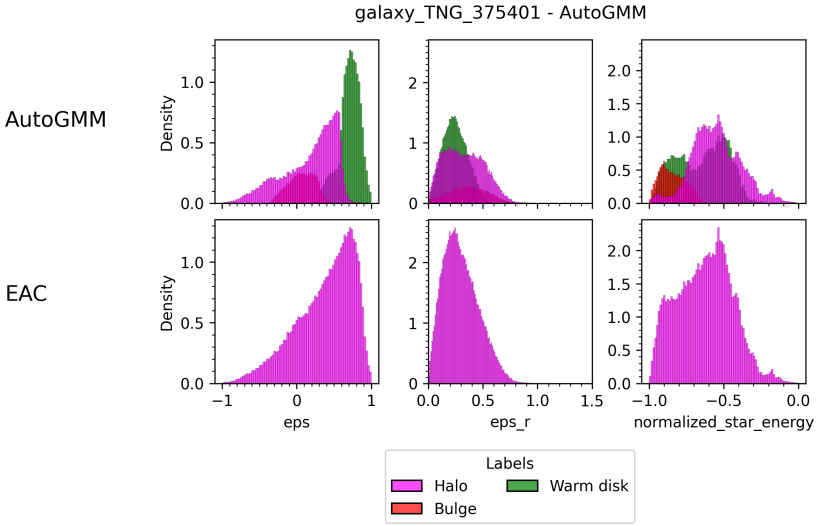

Figura 5.6: Comparación AGMM contra EAC sobre la galaxia *galaxy_TNG_375401* sobre el histograma del espacio circular.

aplicar EAC sobre nuestro conjunto de datos como así también el tiempo finito que se tuvo para realizar los experimentos, no nos es posible seguir investigando en la dirección sugerida variando el valor de k en cada corrida de k-means, y menos aún poder optimizar las cotas de dicha variación como hiperparámetros.

Finalmente, dados los pésimos resultados obtenidos al buscar más de dos componentes y los altos costos computacionales y de memoria, no nos es posible recomendar EAC como alternativa a AGMM con las pruebas encontradas hasta el momento.

Capítulo 6

Conclusiones

Para la realización de este trabajo se llevó a cabo un estudio sobre la composición estelar de galaxias y su descomposición dinámica, haciendo uso de los métodos de aprendizaje automático HC, FCM y EAC, en conjunto con las técnicas de eliminación de outliers RCut e IF. Se utilizaron los resultados obtenidos para comparar con los generados por métodos existentes y ya probados en el área como Abadi y AGMM, utilizando métricas externas e internas para evaluar el éxito de los experimentos.

Como resumen de las conclusiones extraídas en cada capítulo podemos destacar:

1. Se pueden obtener resultados cercanos a los obtenidos con Abadi utilizando HC, siempre y cuando se entienda de las limitaciones provenientes de utilizar una sola feature para aplicar métodos de clusterización como el mencionado. Aun así, dado el alto costo computacional y de memoria de éste en comparación con Abadi, no se lo puede recomendar como alternativa.

2. La tendencia de HC a obtener resultados cercanos a los de Abadi es un indicativo extra del éxito de este último como algoritmo para identificar componentes.

3. Dada la naturaleza de ciertas subcomponentes como el Bulge y el Halo que van en contra de los conceptos usuales de clusterización, no es posible para métodos de clustering que no asignan un significado físico a las features del espacio dinámico lograr identificar las subcomponentes mencionadas correctamente.

4. Las métricas internas utilizadas, Davies–Bouldin y Silhouette, no son de utilidad para analizar el desempeño de algoritmos de clustering en el problema presentado, dada la naturaleza de las componentes reales de una galaxia y como estas difieren de, lo que entienden estas métricas, son buenos clusters.

5. Al igual que con Abadi, HC reconfirma ciertas tendencias reflejadas por los resultados obtenidos por AGMM.

6. FCM también presenta tendencias similares a los resultados obtenidos con Abadi, lo cual lo hace una alternativa válida (siempre y cuando se entiendan sus limitaciones). Sin embargo, el método no muestra un comportamiento aceptable al buscar más de dos clusters para ser considerado una alternativa a AGMM.

7. Los resultados obtenidos por EAC distan tanto de los buscados que no es posible recomendar la aplicación de dicho método de la forma explorada en este trabajo. Además, el tamaño de los datos usualmente manejados en el área es tan alto que el costo computacional y de memoria de EAC es varias órdenes de magnitud mayor que los vistos en Abadi o AGMM, desincentivando de esta manera cualquier aplicación de este método en la práctica, al menos con la implementación utilizada.

Cabe aclarar que, cuando hablamos de estos algoritmos como posibles alternativas a las ya propuestas en el área, hablamos estrictamente respecto a la tarea de encontrar clusters de forma similares. Dado que los métodos usados en este trabajo son de uso general en minería de datos, no es posible para ellos etiquetar cada cluster con la etiqueta correspondiente, tarea que queda a cargo de quien aplica dichos métodos.

Además, no fue posible concluir con ninguno de los métodos explorados si remover outliers beneficia o no la tarea de buscar los clusters deseados. Remover outliers dificulta además el análisis al buscar un número variable de clusters, dado que remover partículas estelares de los datos cambia en muchos casos la cardinalidad de la partición.

Para finalizar, dado el alcance propuesto por la tesina y el costo en tiempo y memoria de los métodos utilizados, se sugieren los siguientes trabajos a futuro:

1. Extender el dataset propuesto para realizar experimentos con galaxias espirales.

2. Normalización del espacio dinámico antes de correr FCM, como así también antes de correr HC.

3. Explorar otro tipo de preprocesamientos sobre el espacio dinámico antes de aplicar los métodos propuestos.

4. Aplicar diferentes variaciones sobre el primer paso del algoritmo de EAC, como se explica al comienzo del Capítulo 5.

5. Correr EAC variando los valores de k de cada k-means, buscando además la cota inferior y superior de dicha variación como hiperparametros.

6. Evaluar otras técnicas de eliminación de outliers provenientes de minería de datos.

7. Explorar otras features del dataset que podrían contener más información útil para encontrar componentes con los métodos propuestos. Como la edad o la metalicidad.

Apéndice A

Gráficos complementarios

Dada la gran cantidad de gráficos producidos para la realización de este trabajo no nos es posible incluirlos a todos en este mismo documento. Por lo cual, de estar interesado en revisarlos, puede hacerlo entrando en el siguiente repositorio: github.com/originalnicodr/GalaxyMLDecompositionThesis.

Glosario

AGMM Auto Gaussian Mixture 30–37, 44–50, 58–61, 65, 66

AI Inteligencia Artificial 10

CM Matriz de Confusión 12

DBI Índice de Davies–Bouldin 13, 14

DM Minería de datos 3, 5

EAC Evidence Accumulation Clustering ..... 4, 51–61, 66, V

FCM Fuzzy Clustering 4, 38, 40–50, 60, 61, 65–67

FN Falsos negativos 12, 67

FP Falsos positivos 12, 67

GT Ground Truth 3, 12, 14, 22, 51, 55, 67

HC Clustering Jerárquico 4, 18–37, 44, 55, 60, 61, 64, 65, 67

IF Isolation Forest 15, 16, 27–29, 33, 35, 36, 42, 49, 60, 64, 65

ML Aprendizaje automático 3–5, 10

RCut Corte en distancia radial 15, 17, 26–29, 35, 36, 42, 49, 60, 64, 65

SSc Puntaje de Silhouette 13

TN Verdaderos negativos 12, 67

TNG The Next Generation Illustris Simulations 2–4, 7

TP Verdaderos positivos 12, 13, 67

VPC Curva de Perfil de Velocidades 14

Índice de figuras

Índice de tablas

Bibliografía

Abadi, M. G., Navarro, J. F., Steinmetz, M., and Eke, V. R. (2003). Simulations of galaxy formation in a λ cold dark matter universe. ii. the fine structure of simulated galactic disks. *The Astrophysical Journal*, 597(1):21.

Aranganayagi, S. and Thangavel, K. (2007). Clustering categorical data using silhouette coefficient as a relocating measure. In *International conference on computational intelligence and multimedia applications (ICCIMA 2007)*, volume 2, pages 13–17. IEEE.

Borne, K. D. (2010). Astroinformatics: data-oriented astronomy research and education. *Earth Science Informatics*, 3(1-2):5–17.

Carroll, B. W. and Ostlie, D. A. (2006). An introduction to modern astrophysics and cosmology. *An introduction to modern astrophysics and cosmology/BW Carroll and DA Ostlie. 2nd edition. San Francisco: Pearson.*

Cristiani, V. A. (2020). Descomposición dinámica de galaxias simuladas. B.S. thesis.

Dave, R. and Krishnapuram, R. (1997). Robust clustering methods: a unified view. *IEEE Transactions on Fuzzy Systems*, 5(2):270–293.

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227.

Du, M., Ho, L. C., Zhao, D., Shi, J., Debattista, V. P., Hernquist, L., and Nelson, D. (2019). Identifying kinematic structures in simulated galaxies using unsupervised machine learning. *The Astrophysical Journal*, 884(2):129.

Fluke, C. J. and Jacobs, C. (2020). Surveying the reach and maturity of machine learning and artificial intelligence in astronomy. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(2):e1349.

Fox, P. (2011). The rise of informatics as a research domain. In *Proceedings of WIRADA Science Symposium, Melbourne, Australia*, volume 15, pages 125–131.

Fred, A. L. and Jain, A. K. (2005). Combining multiple clusterings using evidence accumulation. *IEEE transactions on pattern analysis and machine intelligence*, 27(6):835–850.

Governato, F., Brook, C., Brooks, A. M., Mayer, L., Willman, B., Jonsson, P., Stilp, A., Pope, L., Christensen, C., Wadsley, J., et al. (2009). Forming a large disc galaxy from az¡1 major merger. *Monthly Notices of the Royal Astronomical Society*, 398(1):312–320.

Hey, A. J., Tansley, S., Tolle, K. M., et al. (2009). *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA.

Hubble, E. (1936). The realm of the nebulae, yale univ.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Marinacci, F., Pakmor, R., and Springel, V. (2014). The formation of disc galaxies in high-resolution moving-mesh cosmological simulations. *Monthly Notices of the Royal Astronomical Society*, 437(2):1750–1775.

Márquez, D. G., Félix, P., García, C. A., Tejedor, J., Fred, A. L., and Otero, A. (2019). Positive and negative evidence accumulation clustering for sensor fusion: An application to heartbeat clustering. *Sensors*, 19(21):4635.

Mitchell, T. M. et al. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877.

Mo, H., Mao, S., and White, S. D. (1998). The formation of galactic discs. *Monthly Notices of the Royal Astronomical Society*, 295(2):319–336.

Mo, H., Van den Bosch, F., and White, S. (2010). *Galaxy formation and evolution*. Cambridge University Press.

Obreja, A., Macciò, A. V., Moster, B., Dutton, A. A., Buck, T., Stinson, G. S., and Wang, L. (2018). Introducing galactic structure finder: the multiple stellar kinematic structures of a simulated milky way mass galaxy. *Monthly Notices of the Royal Astronomical Society*, 477(4):4915–4930.

Park, M.-J., Sukyoung, K. Y., Dubois, Y., Pichon, C., Kimm, T., Devriendt, J., Choi, H., Volonteri, M., Kaviraj, S., and Peirani, S. (2019). New horizon: On the origin of the stellar disk and spheroid of field galaxies at z= 0.7. *The Astrophysical Journal*, 883(1):25.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pillepich, A., Springel, V., Nelson, D., Genel, S., Naiman, J., Pakmor, R., Hernquist, L., Torrey, P., Vogelsberger, M., Weinberger, R., et al. (2018). Simulating galaxy formation with the illustristng model. *Monthly Notices of the Royal Astronomical Society*, 473(3):4077–4106.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Scannapieco, C. e. a., Wadepuhl, M., Parry, O., Navarro, J., Jenkins, A., Springel, V., Teyssier, R., Carlson, E., Couchman, H., Crain, R., et al. (2012). The aquila comparison project: the effects of feedback and numerical methods on simulations of galaxy formation. *Monthly Notices of the Royal Astronomical Society*, 423(2):1726–1749.

Schaye, J., Crain, R. A., Bower, R. G., Furlong, M., Schaller, M., Theuns, T., Dalla Vecchia, C., Frenk, C. S., McCarthy, I., Helly, J. C., et al. (2015). The eagle project: simulating the evolution and assembly of galaxies and their environments. *Monthly Notices of the Royal Astronomical Society*, 446(1):521–554.

Sharma, S., Batra, N., et al. (2019). Comparative study of single linkage, complete linkage, and ward method of agglomerative clustering. In *2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon)*, pages 568–573. IEEE.

Shen, S., Mo, H., White, S. D., Blanton, M. R., Kauffmann, G., Voges, W., Brinkmann, J., and Csabai, I. (2003). The size distribution of galaxies in the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society*, 343(3):978–994.

Sparke, L. S. and Gallagher III, J. S. (2007). *Galaxies in the universe: an introduction*. Cambridge University Press.

Tissera, P. B., White, S. D., and Scannapieco, C. (2012). Chemical signatures of formation processes in the stellar populations of simulated galaxies. *Monthly Notices of the Royal Astronomical Society*, 420(1):255–270.

Van Albada, T. S., Bahcall, J. N., Begeman, K., and Sancisi, R. (1985). Distribution of dark matter in the spiral galaxy ngc 3198. *Astrophysical Journal, Part 1 (ISSN 0004-637X), vol. 295, Aug. 15, 1985, p. 305-313.*, 295:305–313.

Van de Hulst, H., Raimond, E., and Van Woerden, H. (1957). Rotation and density distribution of the andromeda nebula derived from observations of the 21-cm line. *Bulletin of the Astronomical Institutes of the Netherlands, Vol. 14, p. 1*, 14:1.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa,

F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Vogelsberger, M., Genel, S., Springel, V., Torrey, P., Sijacki, D., Xu, D., Snyder, G., Bird, S., Nelson, D., and Hernquist, L. (2014). Properties of galaxies reproduced by a hydrodynamic simulation. *Nature*, 509(7499):177–182.

White, S. D. and Frenk, C. S. (1991). Galaxy formation through hierarchical clustering. *The Astrophysical Journal*, 379:52–79.

White, S. D. and Rees, M. J. (1978). Core condensation in heavy halos: a two-stage theory for galaxy formation and clustering. *Monthly Notices of the Royal Astronomical Society*, 183(3):341–358.

Yu, H. and Hou, X. (2022). Hierarchical clustering in astronomy. *Astronomy and Computing*, page 100662.

Zhao, Y., Wang, X., Cheng, C., and Ding, X. (2020). Combining machine learning models and scores using combo library. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, New York, USA.