# Open Clusters in the Milky Way

Ethan Raisbeck
201245451

Supervised by Dr Andrei Igoshev

Submitted in accordance with the requirements for the
module MATH5872M: Dissertation in Data Science and Analytics
as part of the degree of

Master of Science in Data Science and Analytics

The University of Leeds, School of Mathematics

September 2022

# Abstract

Open clusters are small collections of stars born from the same molecular cloud that are loosely gravitationally bound to each other. Identifying them for further study is crucial for increasing our knowledge of astrophysics and understanding of the evolution of our Galaxy. The recent public release of *GAIA* DR3 provides a remarkable amount of precision data concerning almost 2 billion stars. In this work we use three unsupervised density based clustering algorithms, DBSCAN, HDBSCAN and OPTICS, to assess their capabilities in detecting open clusters within *GAIA* DR3. Clustering is carried out in two regions of space using the 5 parameter full astrometric solution of radial ascension and declination coordinates, parallax angle and proper motion with respect to radial ascension and declination. We later investigate changes in clustering performance with the addition of radial velocity and absolute magnitude data. Results are compared to previously documented clusters from Cantat-Gaudin et al. (2018), Hunt & Reffert (2021), Dias et al. (2021) and Castro-Ginard et al. (2022). Additionally, Hertzsprung–Russell diagrams are used for assessing membership assignment between the clustering algorithms. We conclude that HDBSCAN is the best clustering algorithm in terms of performance and computational expense, followed closely by DBSCAN. OPTICS produces almost identical results to DBSCAN but with an unforgiving computational expense, making it unviable for use on any *GAIA* dataset. Due to significant segments of missing data, clustering with radial velocity measurements is largely inadequate for aiding open cluster detection. We also conclude that clustering with absolute magnitude data fundamentally yields no benefits.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Open Clusters

Star clusters are collections of stars that are grouped together more densely than their surrounding stars. There are two types of star cluster, the open cluster and globular cluster. Globular clusters are large, dense, spherical clusters which usually contain very old stars, and are strongly bound together. Open clusters are much smaller, typically containing a few hundred, or in large cases thousands of stars. The stars are all born together from the same molecular cloud, hence they are approximately the same age and have similar chemical compositions. Unlike globular clusters they are bound together loosely, meaning that over time stars will accumulate enough energy from gravitational interactions to escape the cluster. As a result, open clusters do not exist forever, and they will gradually get smaller over hundreds of millions of years, until they disappear completely.

*Figure 1.1*: The NGC3603 open cluster.
*Accessed from https://esahubble.org/images/opo1022a/*

Open clusters are typically found where there is a high gas density. Over many millions of years the force of gravity works accumulate gas and dust together to form stars. The stars formed within the cluster are bound weakly, therefore they are highly susceptible to being dispersed by gravitational interactions with other large objects. Gravitational interactions are also known as tidal forces, and can create tidal tails on open clusters, characterised by a trail of stars moving slightly behind the cluster. Due to tidal forces, open clusters are more likely to be prominent in the spiral arms of the Milky Way, close to the galactic plane. The closer they are to the galactic centre, the more likely they are to be dispersed by interactions, as there is an increase in density of large objects. This is also the reason for older open clusters often being found towards the end of the spiral arms. There are over 1000 open clusters that have been confirmed to exist in the Milky Way. However, it is believed that this may only be a small fraction of the actual number. With the new data release 3 (DR3) from the *GAIA* mission, as well as the upcoming DR4 and DR5, it is very likely that a lot more will be discovered.

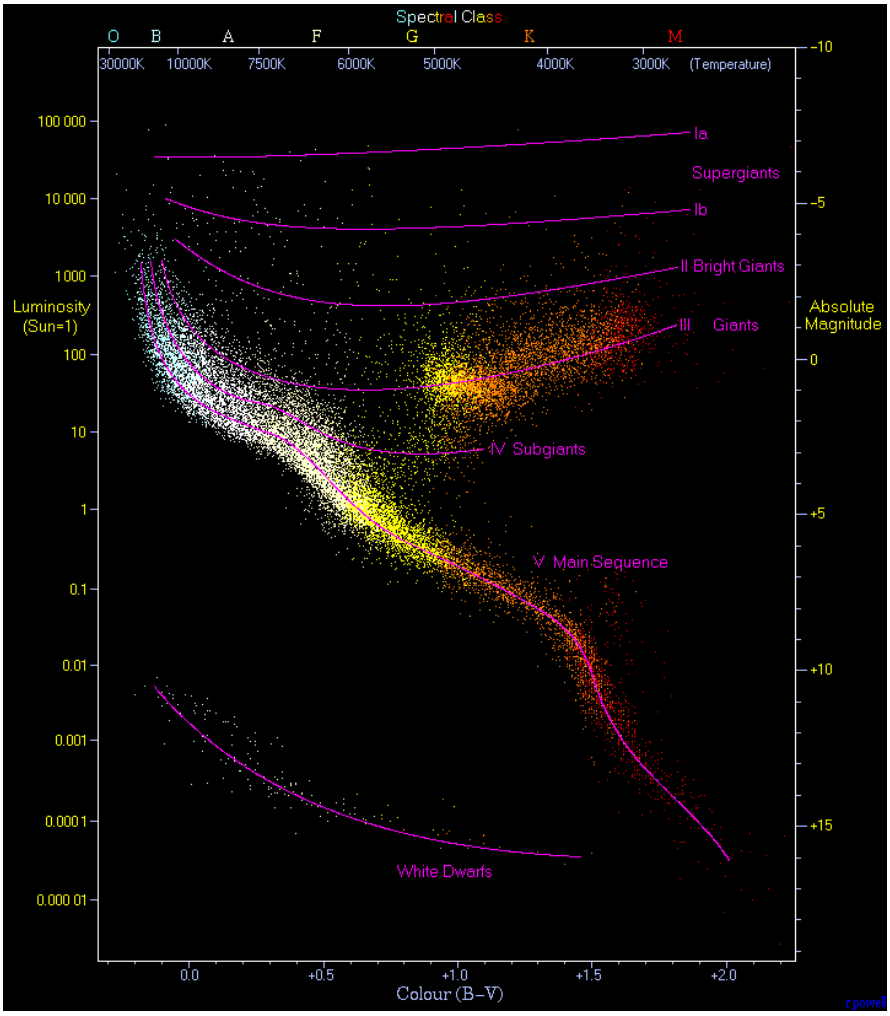



*Figure 1.2*: Hertzsprung–Russell Diagram.
*Accessed from: https://en.wikipedia.org/wiki/Hertzsprung-Russell_ diagram*

By learning about the behaviour and life-cycle of open clusters, scientists are able to understand how our Galaxy has evolved to be as it is today. Eventually, it may allow us to trace the exact history of our solar system, as our Sun was likely also once part of an open cluster. Additionally, open clusters are havens for astrophysicists to be able to study stellar structure and evolution. Because all the stars in open clusters formed at the same time and from the same nebula, their age, location, and composition are approximately the same. The difference in each star within the cluster can be assumed to be based almost entirely on its size, which makes comparing stars within the cluster very simple.

As open clusters have only have a life span of a few hundred million years, usually all of the stars within them are young and still on the main sequence. This can be plotted on a colour-magnitude diagram, named a Hertzsprung–Russell diagram or HR diagram (Figure 1.2). The HR diagram shows the colour of a star which is determined by its temperature, and also dictates which spectral class it belongs too. Additionally, it shows the brightness of the star, also named its luminosity or absolute magnitude. All stars begin on the main sequence, and as they evolve they can move move off it to become red giants.

This means that HR diagrams can be used to identify an open cluster, and can even tell the age of the cluster based off how many stars have moved off the main sequence. Unfortunately, since the colour and magnitude of a star are physically related, it makes it incompatible with algorithms which are available to identify the clusters, as there must be no relation between variables. Although they cannot be used directly in the algorithms, HR diagrams are useful for verifying the true existence of a cluster. Very frequently, clustering algorithms which aim to detect open clusters do so incorrectly. What they have detected is still a cluster of stars which exhibit an increase in density in the phase space of their clustering dimensions. However, the cluster of stars is not a real physical open cluster but a mathematical cluster, and in this work we will refer to these as statistical clusters. These non-physical statistical clusters can be quite troublesome, and HR diagrams are the solution to distinguish between a physical and statistical cluster by looking for stars which are mostly on the main sequence. This can even be taken care of in an automated process by an artificial neural network or supervised machine learning tool, which is able to learn the shape of an HR diagram corresponding to an open cluster.

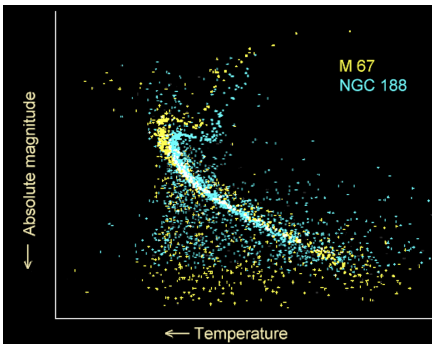



*Figure 1.3*: HR diagram of two open clusters, showing most stars on the main sequence. *Accessed from: https://en.wikipedia.org/wiki/Hertzsprung-Russell_diagram*

## 1.2 Astronomical Background

**Right Ascension and Declination** In astronomy, it is useful to be able to define a coordinate system which can describe the position in the sky of any stellar object. In this work we use the celestial coordinate system, which defines a position in terms of right ascension $(\alpha)$ and declination $(\delta)$, abbreviated to RA and DEC. To understand how this works, we first visualise the celestial sphere (Figure 1.4).

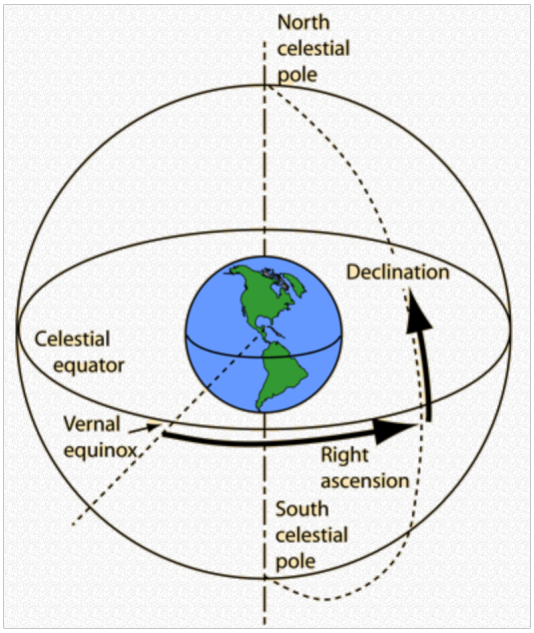



*Figure 1.4*: Celestial coordinate system.
*Accessed from: http://hyperphysics.phy-astr.gsu.edu/hbase/eclip.html*

The celestial sphere is virtual sphere whereby this coordinate system can be defined. The system uses RA and DEC in the same way that latitude and longitude are used respectively. RA is the same as the celestial equator, which runs in line with the equator of the Earth. DEC runs perpendicular to right ascension, intersecting the RA line at the spring and autumn equinoxes. Both RA and DEC are measured in degrees from the spring equinox point, also named the Vernal equinox in Figure 1.4. Typically, RA values are shown as (hours, minutes seconds), where 1 hour is equivalent to 15 degrees. DEC values are typically shown as (degrees, minutes, seconds).

**Parallax** Another important capability is being able to measure the distance to objects which are being observed. This is achieved through measuring the angle of the apparent change in position of an object, with respect to background objects much further away. This angle is named the parallax angle $(\varpi)$, and is shown in Figure 1.5. After obtaining the parallax angle, the distance to the object can be calculated using simple trigonometry as in Equation 1.4. In this equation, $d$ and $\theta$ are measured in parsecs and radians respectively. This calculation also makes use of the small angle approximation to eliminate the $tan()$

function. To create the right angled triangle to make this equation usable, the parallax measurements must be taken 6 months apart. Units of measurement required for this calculation are defined as follows:

- Arcseconds or millicarcseconds are the typical unit of a parallax angle

$$1 \text{ arcsecond} = \frac{1}{3600} \text{of a degree} \tag{1.1}$$

- 1 Astronomical Unit (AU) is the distance from the Earth to the Sun

$$1 \text{ AU} = 1.496 \times 10^{11} \text{m} \tag{1.2}$$

- 1 parsec (pc) is the distance to an object where the parallax angle is 1 arcsecond

$$1 \text{ pc} = 1 \text{ arcsecond} = 3.086 \times 10^{16} \text{m} \tag{1.3}$$

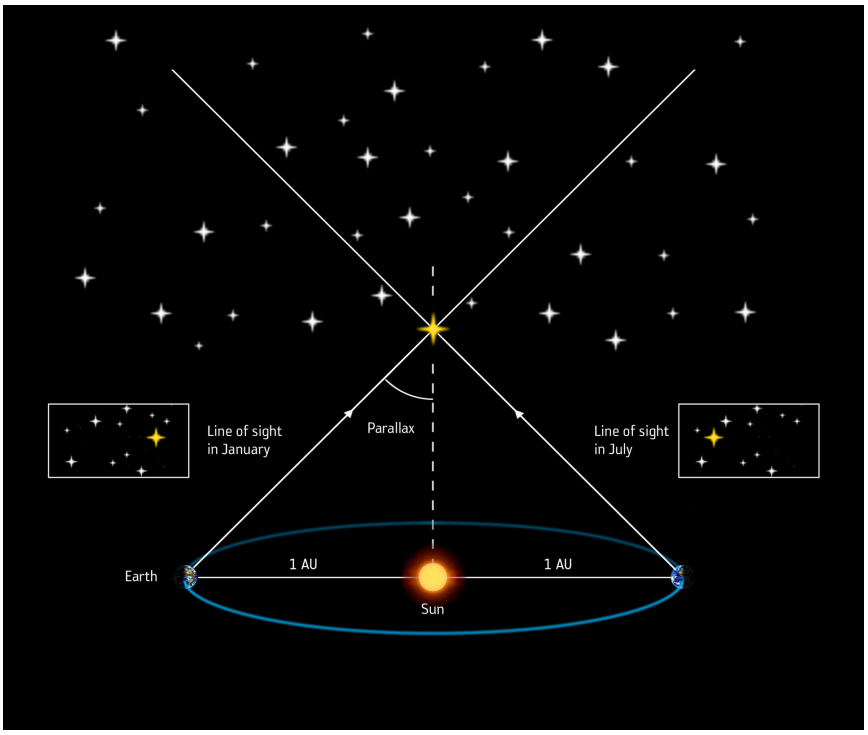



*Figure 1.5*: Measuring the parallax angle.
*Accessed from: https://www.cosmos.esa.int/web//dr3-how-far-away-are-the-stars*

$$d = \frac{1AU}{tan(\theta)} \approx \frac{1AU}{\theta} \tag{1.4}$$

**Proper motion** Defined as the observed motion of a stellar object, proper motion is given by a 2 dimensional vector in the direction of right ascension ($\mu_\alpha$) and declination ($\mu_\delta$). It can be calculated through taking the change in distance of the object when measured at two points in time for each vector component, as in Equation 1.5. In this equation $\alpha$ and $\delta$ are the radial ascension and declination coordinates of the object taken at times 1 and 2, and $\Delta t$ is the change in time between measurements.

$$\mu_\alpha = \frac{\alpha_2 - \alpha_1}{\Delta t}, \quad \mu_\delta = \frac{\delta_2 - \delta_1}{\Delta t} \tag{1.5}$$

Proper motion is given by the angle between the change in position of a star. The angle is measured in two dimensions with respect to RA and DEC, giving the two dimensional vector. The proper motion angle is illustrated in Figure 1.6.

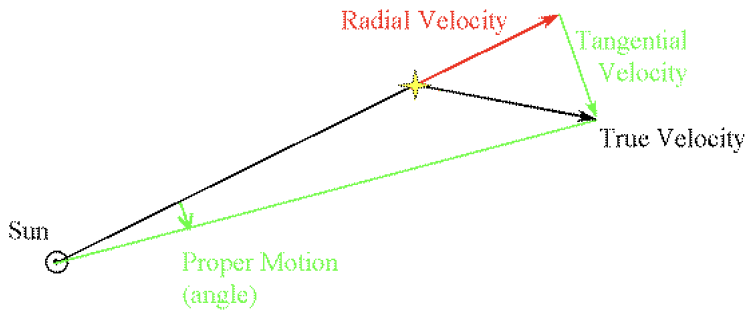



*Figure 1.6*: Proper motion and radial velocity of a star.
*Accessed from: https://www.astronomy.ohio-state.edu/pogge.1/Ast162/Unit1/motions.html*

Proper motion is measured by the rate of change of angle, in milliarcseconds/year. This can be converted into a velocity named the tangential velocity through Equation 1.6, where $\nu_t$, $\mu$, and $D$ are tangential velocity, proper motion and distance, measured in km/s, milliarseconds/year and kpc respectively.

$$\nu_t = 4.74 \mu D \tag{1.6}$$

It is then possible to calculate the true velocity of the star using the tangential $\nu_t$ and radial velocity $\nu_r$ with Pythagoras' Theorem, as in Equation 1.7.

$$\nu^2 = \nu_t^2 + \nu_r^2 \tag{1.7}$$

**Radial Velocity** Measuring the rate of change of distance of a stellar object with respect to the Earth is named radial velocity, illustrated in Figure 1.6. The radial velocity of a star can be measured by taking advantage of the Doppler Effect. This is the change in wavelength of light emitted from a star, resulting from which direction that object is moving relative to an observer. If a star is moving away from the Earth, the observed wavelength of light

is higher (red shift), and if it is moving towards the Earth the wavelength is lower (blue shift). A radial velocity spectrometer can calculate the change in wavelength in the light from a star resulting from the Doppler effect, compared to a reference star. Radial velocity is calculated from the change in observed and reference wavelengths as in Equation 1.8, where $v, c, \lambda$ and $\lambda_0$ are the radial velocity, speed of light, reference wavelength and the observed wavelength respectively.

$$\frac{v}{c} = \frac{\lambda - \lambda_0}{\lambda_0} \tag{1.8}$$

If the star is moving away from us, the radial velocity will be negative, and vice versa. Stars typically have radial velocities of the order of $\pm$ several hundreds of kilometres per second. Stars which exhibit similar radial velocities, hence moving in the same direction, are more likely to be part of an open cluster.

**Apparent and Absolute Magnitude** The magnitude of a star refers to how luminous it is, or the intensity of light which it emits. Apparent magnitude specifically is a measure of how much light we on Earth perceive a star to emit. Of course, due to the inverse square law, the intensity of light from a star we measure on Earth has decreased with the square of its distance from us. Additionally, light can be absorbed or scattered by gas, dust and other objects in space, named extinction. The actual luminosity of a star is described with its absolute magnitude, which is defined as the apparent magnitude of a star when observed at a distance of 10 parsecs. Both apparent and absolute magnitude are unitless, and are measured as a reverse logarithmic scale, meaning that considering a magnitude increase from 1 to 2, this actually equates to a decrease in brightness of 2.512 times.

The magnitude system was first introduced by the Roman astronomer Claudius Ptolemy, and was later changed whereby the definition of a magnitude of zero would be the apparent brightness of the star Vega. Very bright objects can also have a negative magnitude; For example, the Sun has an apparent magnitude of -26.8 and absolute magnitude of 4.8. The absolute magnitude of a star can be calculated as defined in Equation 1.9, where $M, m$ and $p$ are the absolute magnitude, apparent magnitude and parallax in milliarcseconds respectively.

$$M = m - 5\log_{10}\left(\frac{1000}{p}\right) - 5 \tag{1.9}$$

## 1.3 *GAIA*

### 1.3.1 *GAIA* mission

*GAIA* is a space mission lead by the European Space Agency (Prusti et al. (2016)) with the aim of creating the most detailed map to date of our Galaxy. It will achieve this by taking measurements of approximately 1% of all the stars in the Milky Way, which equates to almost 2 billion stars. The *GAIA* satellite is equipped with high precision measuring devices, which can build a detailed profile of stars across the Galaxy. It is positioned at the L2 Lagrange point, 1.5 million km from Earth, where the gravitational fields of the Earth and the Sun cancel each other out. This puts the satellite in a stable, consistent orbit and allows for continuous uninterrupted use of its telescopes. The mission launched in 2013 and is expected to last until 2025, by which there will have been five data releases containing increasingly more accurate data.

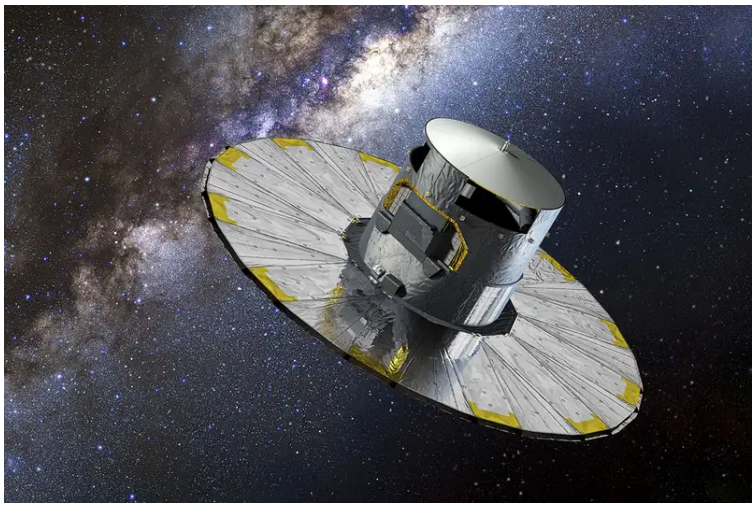


*Figure 1.7*: Artist impression of the *GAIA* satellite.
*Accessed from: https://spaceflight101.com/gaia/gaia-mission-science*

There have been previous attempts to map the Milky Way, most importantly by the *Hipparcos* satellite, launched in 1989. After being decommissioned in 1993, *Hipparcos* was the first ever attempt at measuring precise information of stars required to understand their position and movement. Resulting from the mission, the *Hipparcos Catalogue* (Lindegren et al. (1997)), containing very accurate information on approximately 118,218 stars, and the *Tycho-2 Catalogue* (Hog et al. (2000)) containing less accurate information on just over 2.5 million stars was published. In comparison, the *GAIA* satellite measures positions of stars at an accuracy of 24 millicarcseconds, almost 10 times than of *Hipparcos*, as well as doing this for almost 2 billion stars. This standard of data provides a renaissance in astronomy, with many new discoveries about our Galaxy already unveiled concerning kinematic structures, asteroids, dark matter and many more.

### 1.3.2 Measuring Instruments

The *GAIA* satellite contains two identical telescopes which both funnel the light collected through a series of mirrors and a beam combiner into the measuring instruments. For determining properties of stars, *GAIA* is equipped with an astrometric instrument, photometric instrument and a radial velocity spectrometer (RVS). Figure 1.8 illustrates all measuring equipment within *GAIA*.

**Astrometric instrument** Responsible for measuring the position of stars in the sky. This instrument makes use of a technique first implemented by *Hipparcos*. Right ascension, declination, parallax and proper motion values are measured with this instrument.

**Photometric instrument** Measures the spectra of light from a star, using separate blue and red photometers. The type of light a star emits yield key information such as its colour, temperature, chemical composition and magnitude.

**Radial velocity spectrometer** Determines the radial velocity of a star with a low wavelength spectrometer through taking advantage of the Doppler effect. The grating, afocal field correctors and prismatic lenses are also part of the RVS unit.

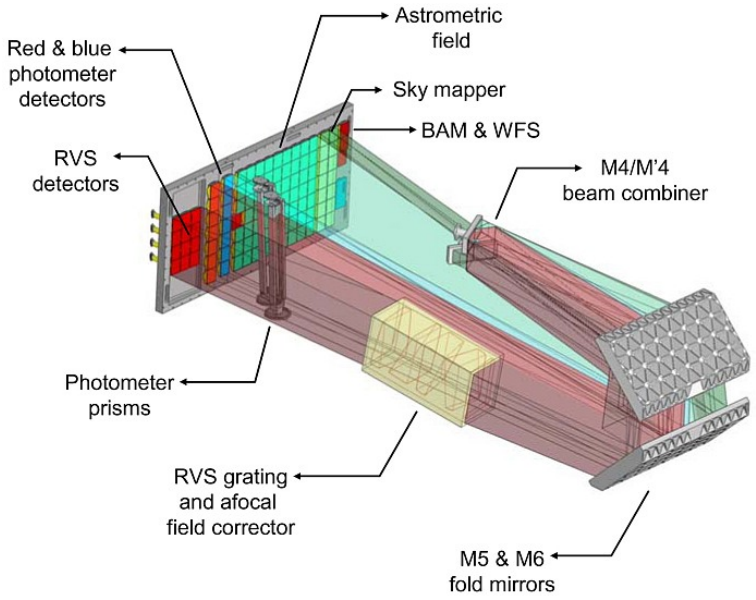



*Figure 1.8*: Measuring instruments within the *GAIA satellite.*
*Accessed from: https://www.cosmos.esa.int/web/gaia/astrometric-instrument*

## 1.4 Clustering Algorithms

In machine learning, clustering algorithms are a type of unsupervised data mining technique which is used to group together data points that are similar to each other. There are many types of clustering algorithms, each with different strengths and weaknesses, making them all useful

for specific tasks. Figure 1.9 illustrates how different clustering algorithms yield varying results on the same set of data. For this research, we have certain requirements of the chosen clustering algorithms which are outlined as follows:

- Do not require a set amount of clusters
- Be efficient for use on a very large dataset
- Clusters can be of any size, shape and density
- Deal with background noise

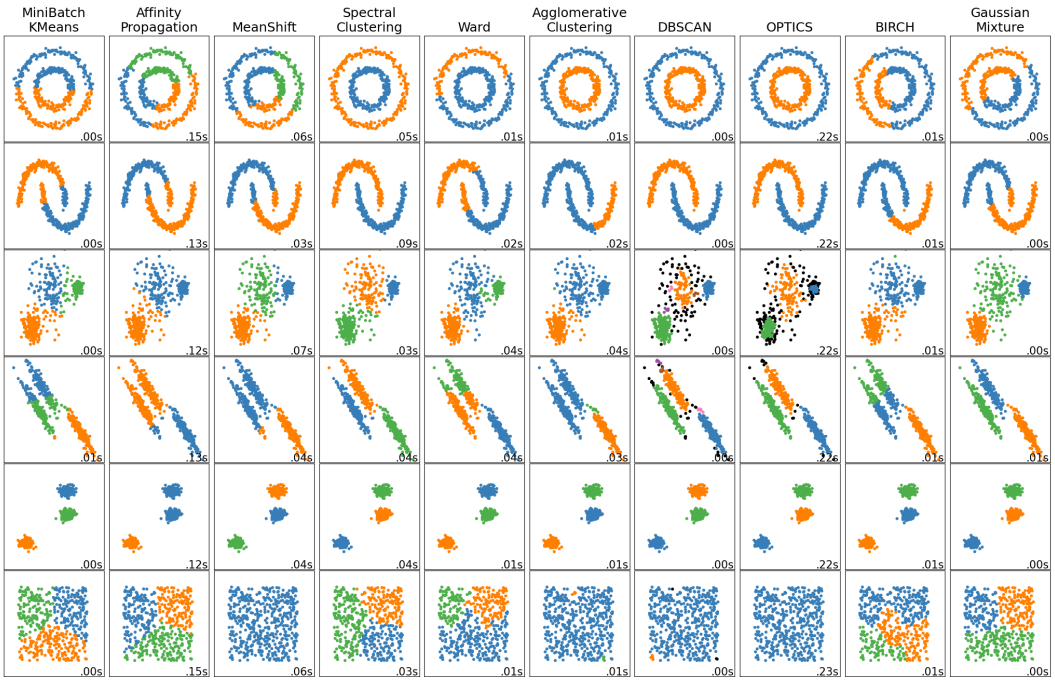



*Figure 1.9*: Clustering algorithms providing different results on the same dataset.
*Accessed from: https://scikit-learn.org/stable/modules/clustering.html*

Through observing Figure 1.9, we are intuitively drawn to the DBSCAN and OPTICS algorithms, given they fulfil most the the stated requirements. They both are able to create clusters of different shapes and sizes, whilst not including any noise points in clusters. They also do not require the number of clusters desired as an input, which is crucial for detecting open clusters where the true number in a region of space is unknown.

### 1.4.1 DBSCAN

DBSCAN (Density Based Spatial Clustering of Applications with Noise) is an algorithm which works by clustering points that are in areas of higher density compared to its surroundings, created by Ester et al. (1996). This algorithm has had much success in published literature for

finding open clusters with the *GAIA* dataset, such as in Castro-Ginard et al. (2018) and Hunt & Reffert (2021), hence it is an obvious choice for this work. It is trivial to understand why there has been so much success with this algorithm, given it is density based and is able to find small clusters amongst lots of noise. Its relatively simple fundamentals also allow for adequate computational costs with use on very large datasets.

The principle behind the DBSCAN algorithm for categorising a cluster is that for a given radius around each data point, given by the parameter $\varepsilon$, there must be a certain number of data points, given by the parameter $min\_points$. Each data point which meets this condition is named a core point. A data point that is within $\varepsilon$ range of core point but does not satisfy the $min\_points$ requirement is labelled as a non-core point. All of the core and non-core points that are within the $\varepsilon$ distance are assigned to one cluster. Any data point that is outside the $\varepsilon$ distance from a core point is labelled as noise. Therefore, a low $min\_points$ and a high $\varepsilon$ will result in large clusters and vice versa. As an example, Figure 1.10 shows the $\varepsilon$ distance given by $\epsilon$ and has a $min\_points$ value of 3, as there are 3 points required within $\varepsilon$ distance of a point to create a core point.

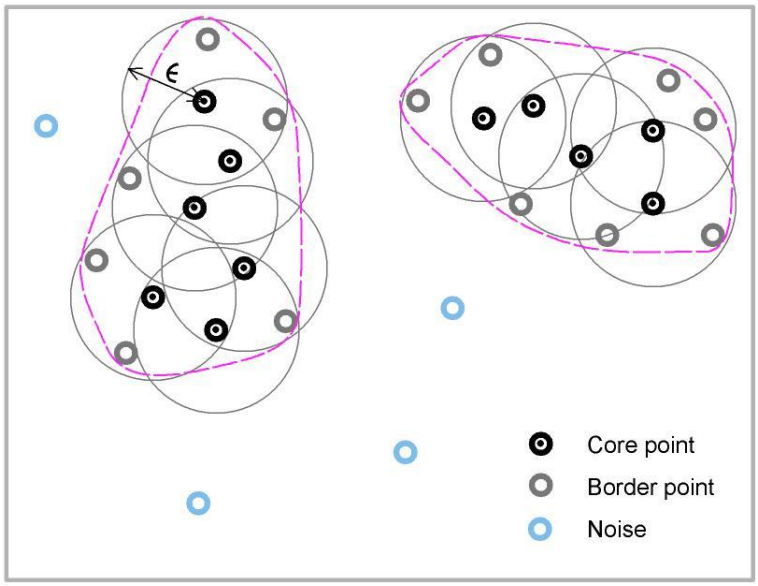



*Figure 1.10*: DBSCAN clustering methodology.
*Accessed from: Olsson et al. (2011)*

DBSCAN calculates the distance between data points using the $k^{th}$ nearest neighbour distance, which is a Euclidean distance. Calculating the distance between two points in a 5D phase space is carried out as in Equation 1.10, where $\alpha, \delta, \varpi, \mu_\alpha, \mu_\delta$ are the RA, DEC, parallax and proper motions in RA and DEC.

$$d(i,j) = \sqrt{(\alpha_i - \alpha_j)^2 + (\delta_i - \delta_j)^2 + (\varpi_i - \varpi_j)^2 + (\mu_{\alpha,i} - \mu_{\alpha,j})^2 + (\mu_{\delta,i} - \mu_{\delta,j})^2} \quad (1.10)$$

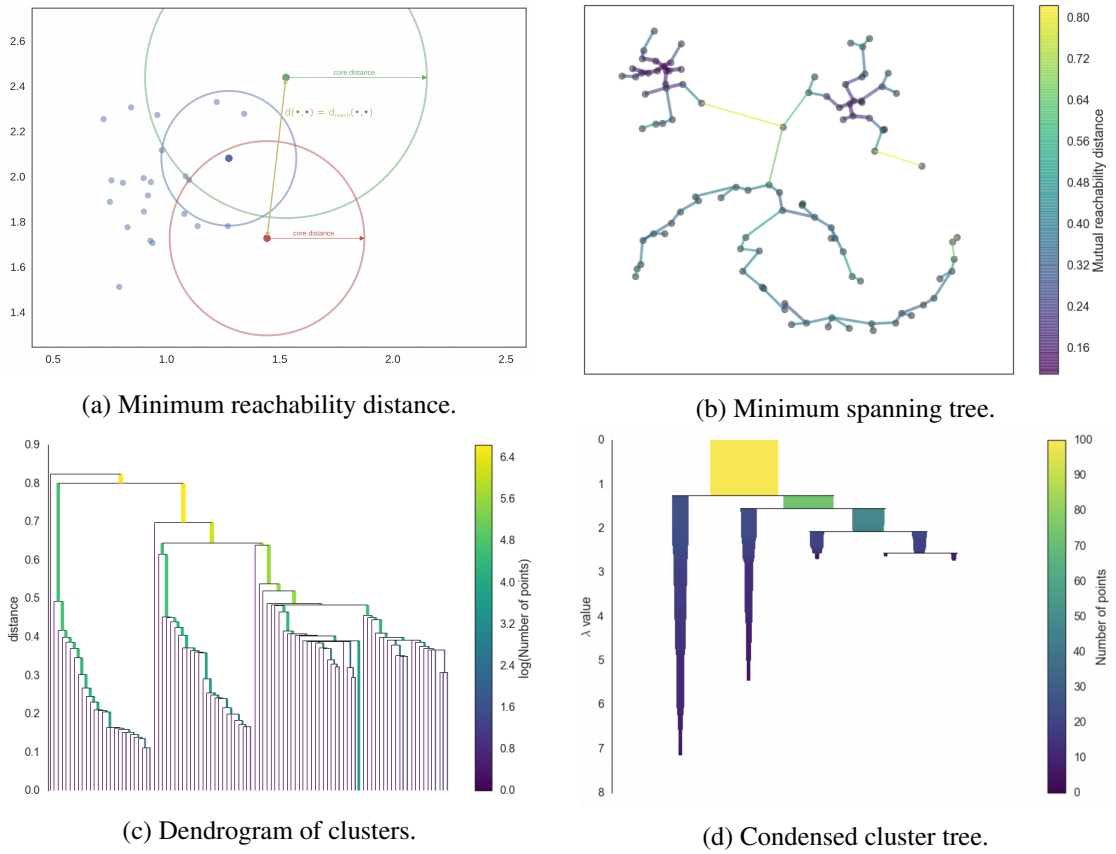
Despite having proven success in finding open clusters, DBSCAN does have drawbacks. The input parameters, most importantly the $\varepsilon$ and $min\_points$ parameters, are user specified and require calculation. No values of these parameters will be optimal for the whole dataset, and particularly with the $\varepsilon$ parameter, a very small change can result in many more or less clusters being identified. Due to the fundamental existence of the $\varepsilon$ parameter, clusters restricted to certain densities. This is a disadvantage of DBSCAN for detecting open clusters, since real open clusters do not have a consistent density throughout their structure.

### 1.4.2 HDBSCAN

As a result of the limitations of DBSCAN, a number of closely related iterations have been created to attempt to mitigate these issues (Khan et al. (2014)). One of these is HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise), proposed by McInnes et al. (2017), which has been reported to be more sensitive and effective at detecting open clusters from the *GAIA* dataset by Hunt & Reffert (2021). Despite this algorithm being far more complicated than DBSCAN, it remains approximately the same with respect to computational cost, which is beneficial for its scalability on large datasets. HDBSCAN focuses on the two main flaws of DBSCAN, being its fixed density of clusters and parameter tuning. It integrates over all $\varepsilon$ values to automatically give the best result, allowing a variable density in clusters. This also removes the need for the user to manually calculate the $\varepsilon$ value, making the implication of the algorithm much simpler to use.

HDBSCAN still makes use of the $k^{th}$ nearest neighbour distance to measure density. McInnes et al. (2017) name the distance to the $k^{th}$ nearest neighbour as the core distance, as in Figure 1.11a. A high core distance would therefore equate to a low density of data points and vice versa. The key difference compared to DBSCAN, is that the core distance, which acts similar to the $\varepsilon$ distance, is variable depending on local density. The distances between the centres of every circle which encompasses $min\_points$ amount of data points is then calculated, which is named the minimum reachability distance. This factors in the distance between points in Euclidean space, as well as the local density of the data points. After obtaining all of the mutual reachability distances, a minimum spanning tree can be plotted as in Figure 1.11b, by joining together points which have the highest minimum reachability distance first, and sequentially adding more which have decreasing minimum reachability distances. This yields a minimum spanning tree with a hierarchy of edges dictated by their length, which can be shown in a dendrogram as in Figure 1.11c.

The cluster hierarchy in Figure 1.11c shows that if we were to cut off the graph using a horizontal line, we could obtain different amounts of clusters. However, setting a cut off value for minimum reachability distance is the same principle as DBSCAN, and would not allow for varying densities of clusters. The HDBSCAN method involves using a condensed cluster tree, as in Figure 1.11d. This works through moving down the hierarchy tree and introducing the $min\_cluster\_size$ parameter, where clusters stop splitting if the size of the split cluster is less




(a) Minimum reachability distance.

(b) Minimum spanning tree.

(c) Dendrogram of clusters.

(d) Condensed cluster tree.

*Figure 1.11*: Obtaining clusters from the minimum reachability distance in HDBSCAN. *Accessed from: https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html*

than this value. From the condensed cluster tree there are two available methods for choosing the clusters. Depending on the requirements of the data, its possible to select clusters which are the largest, which is done with the $exess\_of\_mass$ parameter. This method requires the condition of not selecting any clusters that are sub clusters of one previously chosen. Alternatively if smaller clusters are desired, the $leaf$ method should be used, which makes each leaf node into a cluster.

### 1.4.3 OPTICS

OPTICS (Ordering the Points To Identify the Clustering Structure), created by Ankerst et al. (1999), is technically not a clustering algorithm. OPTICS orders data into a density based structure and it has been shown to yield results similar to DBSCAN. Arguably, OPTICS could be considered to be an improvement over DBSCAN due to the $\varepsilon$ parameter not being required, making implementation of the algorithm far easier. Its major improvement over DBSCAN is the ability to include clusters which have varying densities, by not being restricted to a single value of $\varepsilon$. Its theoretical benefits and ease of use compared to DBSCAN make evaluating its ability to detect open clusters in the *GAIA* dataset definitely worth investigating. We are aware that the run

time of OPTICS does scale with the square of the number of data points, which usually means it will be very computationally expensive for large datasets. This is most likely the reason why it is very difficult to find literature articles which use OPTICS for detecting open clusters within *GAIA*, except for closely related purposed such as cluster membership assignment as carried out by Cánovas et al. (2019).

OPTICS has a different approach for measuring the density of points, making use of core distance and reachability distance metrics as illustrated in Figure 1.12. The core distance is the same metric used in HDBSCAN, being the distance to the furthest of the k nearest neighbours that satisfy the $min\_points$ condition. If a point is within the core distance it is said to be a core point. Reachability distance is either equal to the core distance if the point is a core point, or equal to the Euclidean distance if it is not. The reachability distances are then ordered into a hierarchy, similar to HDBSCAN. Contrast to HDSBCAN however, clusters are chosen based on a cut off reachability distance, which is the $max\_eps$ parameter. This translates to clusters being allowed varying densities up to a maximum density, controlled by $max\_eps$.

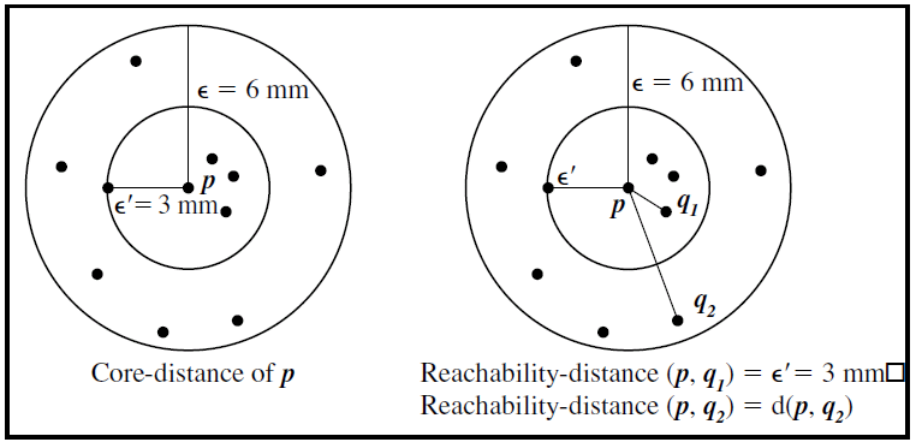



*Figure 1.12*: Core distance and reachability distance.
*Accessed from:*
*https://intellipaat.com/community/1759/python-implementation-of-optics-clustering-algorithm*

## 1.5 Overview

The overall aim of this work is to find open clusters within *GAIA* DR3. To assess the optimal method of achieving this, three clustering algorithms will be tested, assessing their capabilities and performance against previously confirmed existing open clusters. We will also experiment with clustering using radial velocity and absolute magnitude data in the clustering algorithms to investigate if this improves clustering.

**Chapter 1** An introduction containing information on open clusters and HR diagrams, background astronomy relevant to this work, an overview of the *GAIA* mission including its

aims and how it collects data, and details about each clustering algorithm that will be used.

**Chapter 2** A literature review of previous attempts to use clustering algorithms to discover open clusters. The review will look at different clustering methods and how they perform, and highlight key discoveries that are important for future open cluster discoveries, finishing with a summary of key points.

**Chapter 3** The process of requesting and refining the data is explained. The comparison data of previously documented clusters from other research is introduced. Parameter selection methods for the algorithms are detailed and the final parameter choices stated.

**Chapter 4** Results from each algorithm are shown against the comparison data for two regions of space. The predicted clusters from each algorithm are shown tables and graphs, with an HR diagram and detailed view of a single cluster looking into membership assignment. Performance metrics are calculated to quantify performance against the comparison dataset. Clustering with radial velocity and absolute magnitude are tested to assess their impact.

**Chapter 5** Conclusions are drawn from the results outlined in the previous chapter. We discuss the benefits and drawbacks found with respect to open cluster detection from each algorithm. Additionally we conclude the impact on clustering from the clustering with radial velocity and absolute magnitude.

# Chapter 2

# Literature Review

Records of the first sightings of open clusters are documented by Claudius Ptolemy in the second century AD, with the first proper observations with a telescope made by Galileo Galilei in 1609. Owing to the rapid increase in the capabilities of modern technology, detecting open clusters has become far easier, especially with satellite data and data mining methods. The *GAIA* mission has played a major role in this, prompting previous attempts of deploying machine learning algorithms on *GAIA* data releases to search for open clusters. In this literature review, we will cover some of algorithms and techniques used for this, from the most primitive to the most modern.

## 2.1 Gao (2017)

Utilising the first *GAIA* data release, this literature makes use of the very simplistic $k^{th}$ nearest neighbour distance (KNND) algorithm for detecting open clusters. Based on only 38,386 stars within 100 parsecs of Earth, they discover 57 new open cluster members amongst 2 open clusters, identified as *Hyades* and *Coma Berenices*. This was one of the first attempts to use machine learning techniques for discovering open clusters, and at the time of publication the effectiveness of the KNND method had never been tested for this purpose before.

The KNND method is perhaps the most basic way of determining the density of a phase space. It is actually used as a foundation for more complicated algorithms such as DBSCAN, and we outline how to calculate the distance between a data point and its $k^{th}$ nearest neighbour in Equation 1.10. If the distance from a chosen point to the $k^{th}$nearest neighbour is small, it follows that the point will be situated in an area of high density and vice versa. This can be calculated for every star and sorted to yield the points with the highest densities.

In this publication, the authors used a 3D phase space using the right ascension, declination and parallax values of stars to calculate the Euclidean distance to the KNN. Following this, a 2D proper motion space is used to verify cluster membership. It is inherently disadvantageous to perform clustering without more relevant parameters included in the algorithm, such as the proper motion or radial velocity. The main reason for this being it leaves the method more prone

to detecting statistical clusters as opposed to a physical open cluster. A more robust method would be to construct a 5D phase space including proper motions and calculate the KNND's within this.

Similar to the $\varepsilon$ parameter in the DBSCAN method, this paper uses a threshold distance, taken to be 3 parsecs, to separate cluster members from background stars. There is no parameter selection technique referenced in the paper, leading to the reasonable assumption that the introduction of this constraint is sub-optimal and could be improved. Having said this, the KNND algorithm is also highly advantageous because of this simplicity, which also translates into low computational costs. This literature has proven the effectiveness of the KNND algorithm for detecting open clusters. It achieves this whilst having inherent flaws which would hinder performance, such as the use of a 3D phase space and simplistic threshold distance to separate the cluster from background stars. The power of data mining techniques such as clustering algorithms is strongly recognised, even without working to their full potential.

## 2.2 Castro-Ginard et al. (2018)

After some initial success by machine learning methods to identify open clusters from work such as Gao (2017), it became logical to reason that the only effective way of searching for open clusters amongst the billions of stars observed by *GAIA* would be through unsupervised machine learning methods. Only one year later, density based clustering algorithms had started to be applied to the *GAIA* data releases. An important step in this area of research was by Castro-Ginard et al. (2018), who successfully used DBSCAN with an artificial neural network (ANN) to detect 31 new open cluster candidates, and being able to confirm around 70% of them. To achieve this, they used data from the TGAS (*Tycho-Gaia Astrometric Solution*) for detecting possible open clusters, and the *GAIA* DR2 data for validating clusters with the ANN. Five parameters were used within DBSCAN, including the position, parallax and 2D proper motions for each star.

In terms of parameter choices, the $min\_points$ parameter was chosen to be 8, through iterative testing on simulated samples and selecting the best performing one. This is contrary to the recommendation of Ester et al. (1996), who suggests that it should be two times the number of parameters used in DBSCAN. $\varepsilon$ estimation is very important as the results of the algorithm are sensitive to change in this parameter. As such, this was carried out using an automated method for each region of space that was selected.

The idea of the method was firstly to randomly re-sample the data within the range of the parameters in the dataset, over a set number of iterations. This should remove the low KNND values caused by the dense clusters, assuming their overall signature is small. The difference can be shown by producing a KNND plot of the *GAIA* data, and the average of all iterations of the re-sampled data. Finally, the $\varepsilon$ value calculated from this method is taken to be the average of the KNND contribution from the clusters. We outline this method in more detail below:

- Create the $k^{th}$ nearest neighbour distance plot from the chosen region of the *GAIA* data, and store the minimum $\varepsilon$ value as the parameter $\varepsilon_{KNN}$.
- Use a Gaussian kernel density estimator to generate 30 random samples of the same number of stars, according to the distribution of each of their parameters. Take an average over all 30 distributions, and store the minimum $\varepsilon$ value on a KNND plot as $\varepsilon_{rand}$.
- The final $\varepsilon$ value is given by $\varepsilon = (\varepsilon_{KNN} + \varepsilon_{rand})/2$.

The ANN used to verify the clusters is trained on Hertzsprung–Russell diagrams which are created from the colour and magnitude data within *GAIA* DR2. 296 images were used to train the classifier in the training set, yielding a prediction precision of 97.95% when evaluated on the test set. This proves than neural nets are capable of distinguishing statistical clusters from physical clusters, although Castro-Ginard et al. (2018) note that a larger, more diverse dataset should be used for training before this classifier is fully deployed to the *GAIA* data.

## 2.3 Hunt & Reffert (2021)

Owing to the success of clustering algorithms for detecting open clusters, various different algorithms have been trialed for this purpose to assess their effectiveness. A very insightful paper which evaluates the performance of three clustering algorithms was published by Hunt & Reffert (2021). The algorithms chosen were DBSCAN, HDBSCAN and Gaussian mixture models (GMMs), and they were implemented on the *GAIA* DR2 data. The results of the study conclude that out of the three methods, HDBSCAN was overall the most effective method for detecting new clusters with the *GAIA* data, with 41 new possible cluster candidates detected from the study.

DBSCAN was shown to have a accuracy of 50% to 62% to correctly predicting open clusters. The parameter selection for DBSCAN compared two methods; Firstly, the same method devised by Castro-Ginard et al. (2018), which involves randomly re-sampling from a selected region of data and plotting a $k^{th}$ nearest neighbour distance (KNND) graph. Due to the computational cost of this method, the authors of the paper created their own method, which consisted of fitting a function to the KNND graph and extracting $\varepsilon$ as a parameter from the curve fit. The results of this literature show that DBSCAN can give effective results when it is run multiple times with different $\varepsilon$ parameter values and the results combined. This highlights the key issue with DBSCAN which is that there is not a perfect $\varepsilon$ value that is able to identify clusters at with different distances, sizes and densities.

HDBSCAN proved to be more precise to detecting true positive open clusters, having a accuracy of 82% to the comparison dataset. Unfortunately, it was discovered that the increase in accuracy also caused a large number of false positive statistical clusters. As there is no proper method to select the value of the $min\_cluster\_size$ parameter, this paper compares a number of different parameter values. It was found the HDBSCAN was very effective at detecting open

clusters at all distances and densities. They determine that the major downfall of this algorithm is its over sensitivity, leading to a huge number statistical clusters appearing as false positives, or even splitting a single real cluster into sub-clusters.

GMMs are very different to the previous two algorithms, mostly because they are not density based and do not have a way to deal with noise. GMMs work on the assumption that all data points belong to a number of Gaussian distributions. A mixture of the Gaussians are combined into a mixture model. The algorithm uses the expectation maximisation algorithm to fit the mixture model. Although GMMs are not supposed to be complex, they are found to run significantly slower than both DBSCAN and HDBSCAN, with the authors concluding that it is not viable for use on large datasets such as *GAIA*. Additionally, it had the lowest accuracy of any algorithm to detect real open clusters when run with its parameters optimised for maximum sensitivity, of only 33%.

It was found that when the selected parameters of these algorithms were chosen to produce their maximum sensitivity, the majority of the detected open cluster candidates were false positives. A two method approach very effectively dealt with this problem, drastically reducing the number of false positives with DBSCAN and HDBCSAN. The first method was to implement very loose restrains on a cluster's overall proper motion, and the radius which contained half of the stars within a cluster. Secondly, a method was devised to compare the density of the cluster candidate with the surrounding density of the background stars. The density was estimated using KNND values and a Mann-Whitney U significance test (McKnight & Najab (2010)) was carried out to determine if the two densities were consistent for the existence of an open cluster. This method alone with the DBSCAN algorithm took 51920 open cluster candidates and removed false positives until 1111 remained, showing how beneficial post processing of clustering algorithm results can be for open cluster discovery.

## 2.4 Ou et al. (2022)

Very recently, a paper which aims to discover kinematic substructures such as open clusters within *GAIA* eDR3 was published, with focus on the HDBSCAN algorithm. Although no new clusters were identified, known structures including the NGC 3201 globular cluster were found. This publication uses many interesting techniques to aid with clustering, and crucially focuses on incorporating uncertainties into HDBSCAN. Ou et al. (2022) stress the importance of doing this, since they discover that member allocation to clusters is not reliable when ignoring uncertainty in measurements.

The combination of multiple sources of data allows for a decreased probability of detecting statistical clusters. Taking advantage of the similar chemical compositions of member stars in clusters, chemical information was combined with the *GAIA* dataset from the *LAMOST* DR6 and *APOGEE* DR17 spectroscopic surveys. Extra radial velocity data from *GAIA* DR2 was combined with *GAIA* eDR3 to give a more comprehensive list. Interesting attention was given

to data quality as suggested by Lindegren et al. (2021), such as removing sources which have a magnitude above 19 to give more accurate data. Corrections were also given to remove the zero point bias in parallaxes, and limitations were placed on colour to remove low magnitude stars adjacent to high magnitude ones.

HDBSCAN being sensitive to uncertainties poses a slight problem, due to the fact that it does not support uncertainties in the algorithm natively. Although very inconvenient, this is possible to solve through randomly re-sampling the data within the uncertainty range of the parameters. This paper creates 100 different realisations, and out of these determines which detected clusters are stable throughout by calculating Jaccard coefficients. Ou et al. (2022) use the application of many techniques which are able to improve clustering results and provide insight into the robustness of a cluster. Accumulating information from different sources, using chemical compositions and radial velocity measurements, as well as re-sampling data within uncertainties are all very interesting novel techniques which could prove highly useful for future open cluster discoveries.

## 2.5 Summary

Reflecting on the methods used for detecting open clusters outlined in our literature review, we show the evolution in machine learning tools and techniques that have been used. Gao (2017) shows one of the earliest attempts at using clustering algorithms for *GAIA* data. The $k^{th}$ nearest neighbour distance values proved a simple and effective method to show increases in densities of the defined phase space resulting from open clusters. This algorithm is incorporated into DBSCAN, used by Castro-Ginard et al. (2018) along with an artificial neural network, which proved to be a large step towards more efficiently detecting and verifying open clusters. This method devised by Castro-Ginard et al. (2018) has continued to be used very recently to successfully detect open clusters, shown in Castro-Ginard et al. (2020) and Castro-Ginard et al. (2022), and has also been named the *OCfinder* method. It was also adapted by Hunt & Reffert (2021) which we review for effectively comparing three clustering algorithms, highlighting the benefit of the hierarchy based algorithm HDBSCAN. We also mention some interesting techniques used alongside the algorithms to aid detection of clusters, such as a dedicated false-positive removal system. The final literature we review from Ou et al. (2022) raises an insightful point regarding utilising uncertainty data within clustering algorithms, to improve robustness of open cluster discoveries. We recognise that including uncertainties can yield excellent results, as proved though use of the novel pyUPMASK algorithm (Pera et al. (2021)) in literature such as He et al. (2022). Ou et al. (2022) construct an inventive method for incorporating uncertainties, which is non trivial given the lack of native support for them in clustering algorithms, yet necessary given the impact it makes.

# Chapter 3

# Methods

## 3.1 *GAIA* DR3

The source of data for this work is *GAIA* DR3 which can be accessed from the European Space Agency, at:

`https://gea.esac.esa.int/archive/`

The database contains 1,811,709,771 sources, and has 1,467,744,818 sources which consist of what is defined as the full astrometric solution, containing 5 parameters being RA, DEC, parallax and 2D proper motions. An example of some data within the database is provided below in Figure 3.1.

| ra | dec | parallax | pmra | pmdec | ruwe | phot_g_mean_mag | bp_rp | radial_velocity |
|---|---|---|---|---|---|---|---|---|
| deg | deg | mas | mas.yr**-1 | mas.yr**-1 | | mag | mag | km.s**-1 |
| 192.7667713443572 | -62.19215501676542 | 0.2857028746791188 | -9.271159819194391 | -1.501409496417052 | 0.9850308 | 18.166086 | 1.899704 | |
| 192.762507462968 | -62.20320887941939 | 0.38846502317056436 | -8.380662745138407 | -0.42498720986835614 | 1.0178628 | 18.625536 | 2.1172943 | |
| 192.7389390723496 | -62.20229898931049 | 0.34715911759668533 | -5.233632464903331 | -1.6677104392812958 | 0.9907444 | 18.798605 | 2.1954918 | |
| 192.75552503109006 | -62.1957150412335 | 0.3974060191155544 | -6.736065362744942 | -1.0999000888401358 | 1.025243 | 18.69886 | 1.9755726 | |
| 192.73939980167782 | -62.20498171655579 | | | | | 20.992735 | 2.1601048 | |
| 192.7435387039311 | -62.20299658978524 | -0.20007504705611223 | -6.7317199240763035 | -0.22862021536971944 | 1.1145608 | | | |
| 192.74686562519068 | -62.20840177342692 | 0.12998446760186438 | -6.061410277128573 | 0.6309160510249571 | 1.0398064 | 19.705296 | 2.9346008 | |
| 192.75469898580064 | -62.20673494438954 | -0.21625024390686726 | -5.802662964535693 | -0.77998142287132 | 1.0301359 | 18.848051 | 3.702938 | |
| 192.76214352640918 | -62.1978048008764 | 0.16016208255493292 | -6.559956634318462 | -0.7698968642957572 | 1.1649953 | 19.108135 | 2.3799953 | |
| 192.76627039101493 | -62.20106023146533 | 0.2658071514155298 | -8.263606734356001 | -1.708153902564386 | 1.102224 | 17.489836 | 1.9991093 | |
| 192.76243028340724 | -62.19814157002947 | 0.08195050271247983 | -4.767661587161586 | 2.044284017262784 | 1.2590698 | 20.702291 | | |
| 192.740289784288 | -62.213383323370415 | 0.006675401655787061 | -5.3459486484753915 | 0.396603197021026 | 1.0502738 | 19.846575 | 2.2979927 | |
| 192.7446893729054 | -62.20971138766427 | | | | | 20.982908 | 1.8538857 | |
| 192.74482635341846 | -62.20999137110566 | | | | | 20.768208 | 2.678358 | |
| 192.74067843791408 | -62.20665784355341 | | | | | 21.053728 | 1.8043575 | |
| 192.74567592043238 | -62.207357298480126 | -0.9223721032691377 | -8.594708005899616 | -0.7378360665950177 | 1.0518335 | 20.348627 | 2.437294 | |
| 266.62232283814154 | -48.26594490316889 | 0.3922899405686899 | -5.51627875432193 | -5.30082785267032 | 0.96535784 | 18.447943 | 1.1404419 | |
| 266.61285172957923 | -48.2629464489521 | 0.28631597640738565 | -1.0065190604242262 | -0.2228522034074359 | 1.0288635 | 19.081198 | 1.0266933 | |
| 266.6196543415409 | -48.26135920706331 | -0.028294190108811734 | -0.7178856822033026 | -6.737152370139528 | 1.0242481 | 18.74233 | 1.1104584 | |
| 266.6241320773894 | -48.267339633727204 | 0.08838654473180702 | -0.6199756026802816 | -6.849120246233264 | 1.0746526 | 19.646416 | 1.2135334 | |

*Figure 3.1*: Segment from the *GAIA* database.
*Accessed from: https://gea.esac.esa.int/archive/*

Figure 3.1 also shows missing data within the database, with radial velocity data being of most interest as there is none shown. This is because only 33,812,183 sources have radial velocity measurements, which is approximately 0.2% of the database.

### 3.1.1 Selecting the data

Obtaining the data from the database is done through the Astronomical Data Query Language (ADQL). Since the database is so vast, data from small regions of space are queried. The function used for this written in Python is outlined as follows:

```
def get_GAIA(ra, dec, angle):
    Gaia.ROW_LIMIT = -1
    Gaia.MAIN_GAIA_TABLE = "gaiadr3.gaia_source"

    query = Gaia.launch_job_async  ("SELECT * "
                                    "FROM gaiadr3.gaia_source "
                                    "WHERE parallax_over_error > 5
    and ruwe < 1.4 and phot_g_mean_mag < 17 and
    1=CONTAINS(POINT('ICRS',ra,dec),CIRCLE('ICRS', "+str(ra)+
    " , " + str(dec) + " , " + str(angle) + "))"
                                    "ORDER by source_id")
    return query.get_results()
```

The function incorporates several cuts to the requested data to provide optimal data quality. Firstly, as suggested by Castro-Ginard et al. (2020), a lower limit to the apparent magnitude of the requested stars is set at 17. This serves to reduce the size of the dataset, whilst also removing stars which typically have a higher error in measurement because they are not very bright, hence are more likely to be measured incorrectly. Furthermore, the reduced unit weight error (ruwe) is set to include stars only less than 1.4, as used by Jadhav et al. (2021) and Apellániz et al. (2021), and that also have a parallax error of $\varpi/\sigma_{\varpi} > 5$, as recommended by Fabricius et al. (2021) . These constraints had the effect of reducing the data size by approximately 40%, leaving only good quality data. Data was selected in two regions, which we define as Region A and Region B, where Region B is closer to the galactic center. RA and DEC coordinates at the centre of these regions is shown in Table 3.1, and the area of the sky is covered through 2 degrees of rotation.

| | **Region A** | **Region B** |
|---|---|---|
| RA (deg) | 272.90 | 266.42 |
| DEC (deg) | -18.54 | -29.01 |
| Number of stars | 171,065 | 174,851 |

*Table 3.1*: Requested regions of data from *GAIA* DR3.

All of the data from both regions can be visualised as a heatmap showing the distribution of stars across the region, as in Figure 3.2 and 3.3. As expected, there is a dense cluster of stars in Region B, being close to the galactic centre, where there should be more open clusters.

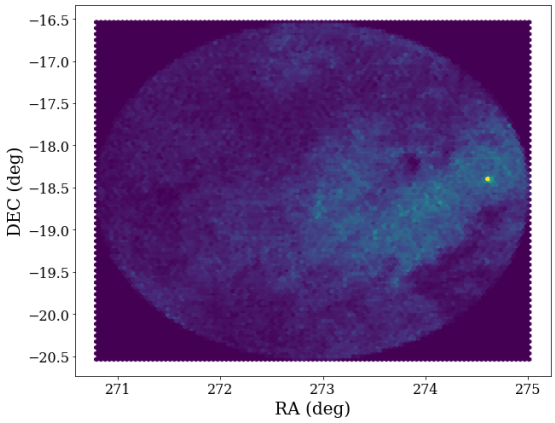



*Figure 3.2*: Density of stars in Region A.

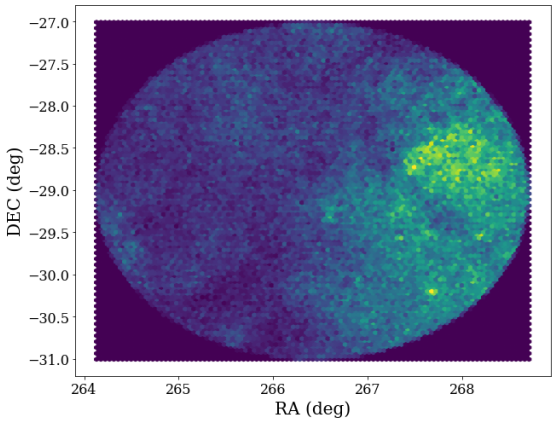



*Figure 3.3*: Density of stars in Region B.

### 3.1.2 Data pre-processing

The first step of pre-processing is to select the required parameters of the data from the query. To work with the full astrometric solution of 5 parameters, this is transferred to a *pandas* DataFrame as follows:

```
data = pd.DataFrame(gaia_request,
                    columns=['ra','dec','parallax','pmra','pmdec'])
```

Normalisation of data is generally a standard part of pre-processing for clustering algorithms. The algorithms usually benefit from having data that is normally distributed. Knowing this, we used the *MinMaxScaler* from *scikit-learn* to normally distribute the data between -1 and 1 as follows:

```
scaler = MinMaxScaler()
scaler.fit(data)
data_normalised = scaler.transform(data)
```

Notably however, we discovered that normalising the spatial coordinates can cause complications when trying to perform parameter determination steps, as well as making visualising the discovered clusters within the new coordinate system quite unintuitive. We found that not performing normalisation lead to perfectly adequate results and opted to perform all clustering without any normalisation steps.

Within all the DataFrames created from the imported data, missing data was not an issue. There was no missing data for any parameter except for radial velocity, as the requirements of the query filtered out very unluminous stars that would likely have missing data. Radial velocity data proved quite a challenge to incorporate into clustering, especially knowing that 99.8% of stars in *GAIA* DR3 do not contain radial velocity measurements. Despite this, we discovered only 79.3% missing data within Region A. Imputation would still not be ideal as it would be

likely that a large portion of the imputed values would be very far off the true value, leading to incorrect clustering. The other option would be to only select data which has radial velocity values, however after attempting this no physical clusters were found with any algorithm. We conclude that imputation is the only reasonable option to attempt clustering, with the assumption being that clustering predictions would suffer. Imputation of the radial velocity data is carried out with the *SimpleImputer* from *scikit-learn* using the "mean" approach as follows:

```
imputer = SimpleImputer(strategy='mean')

imputed_data = pd.DataFrame(imputer.fit_transform(data),
                            columns=['ra','dec','parallax',
                            'pmra','pmdec','radial_velocity'])
```

Pre-processing for clustering with absolute magnitude involved implementing the calculation shown in Equation 1.9 from the apparent magnitude data. Apparent magnitude is imported into a DataFrame and replaced with the absolute magnitude as follows:

```
data = pd.DataFrame(gaia_request,
                    columns=['ra','dec','parallax',
                    'pmra','pmdec','phot_g_mean_mag'])

plx = np.array(data['parallax'])
app_mag = np.array(data['phot_g_mean_mag'])

data['Abs_mag'] = appmag - 5 * np.log10(np.array(1000/plx)) + 5

del data['phot_g_mean_mag']
```

## 3.2 Comparison data

To evaluate the performance of each clustering algorithm, we compare the clusters found by the algorithms in this work against previously confirmed clusters. Unfortunately there is not a universal open cluster catalogue, so already discovered clusters must be combined from various independent studies into our own known cluster catalogue. We find this process rather arduous, since each study presents its catalogue in a different style. It requires care when combining the discovered clusters into a single DataFrame to avoid duplicates, especially since some clusters which are the same are given slightly different names. For this study, discovered clusters were aggregated from Cantat-Gaudin et al. (2018), Hunt & Reffert (2021), Dias et al. (2021) and Castro-Ginard et al. (2022). From Hunt & Reffert (2021), this also includes clusters from the Milky Way Star Clusters Catalogue, its first edition published by Kharchenko et al. (2013), which is a large compiled list of open clusters. For each region of space, we plot the clusters on

a graph showing their position with respect to RA and DEC in Figure 3.4 and 3.5, where each cluster being assigned a different colour. Samples of code used to import the catalogue, format it correctly, then combine the catalogues and find the average properties of the cluster are shown below:

```
'''Using the dias catalogue as an example'''
dias_catalogue = Vizier.query_region(coord.SkyCoord(ra=ra,
              dec=dec, unit=(u.deg, u.deg), frame='icrs'),
              width="4d", catalog=["J/MNRAS/504/356/table12"])

np.save('dias_catalogue.npy', dias_catalogue[0])

np.load('dias_catalogue.npy', allow_pickle=True)

dias_catalogue = pd.DataFrame(hunt,
                    columns=['RA_ICRS', 'DE_ICRS','plx', 'Name'])

dias_catalogue.rename(columns = {'Name' : 'Cluster'},
inplace = True)

'''Repeat for all catalogues and then aggregate together'''
cluster_members = pd.concat([cantat_catalogue, hunt_catalogue,
castro_catalogue, dias_catalogue])

cluster_center = cluster_members.groupby('Cluster').mean()
```

Combining the previously discovered clusters, we find 19 open clusters in Region A, displayed in Figure 3.4 with information regarding each cluster given in Table 3.2. In the combined catalogue, we find two clusters named LP_1218 and Liu_218. We were not able to find any source to verify if these two clusters were in fact the same one. Since the average position of these two clusters identical to within 0.1 degrees of both RA and DEC, as well as parallax values being identical to within 0.01 mas, we decided to remove LP_1218, as it also did not have any accompanying stars available from the database where it was queried. Th NGC_6613 and UBC_96 clusters were removed for being outside the detection area of the *GAIA* data requested. This leaves 16 clusters within Region A that are theoretically all detectable by the clustering algorithms. Most clusters in this have a parallax of less than 1, meaning they more than 1kpc away from Earth. In the middle of the plot, we see lots of stars very spread out with a high parallax. This is the UPK_5 cluster which has a parallax angle of 1.769, meaning it is much closer to Earth than the other clusters, hence it covers a larger area of the sky and is perceived as being more spread out across Figure 3.4. It is also interesting to note what appears to be a very elongated tidal tail from the Sgr_OB7 cluster, shown in orange on Figure 3.4.

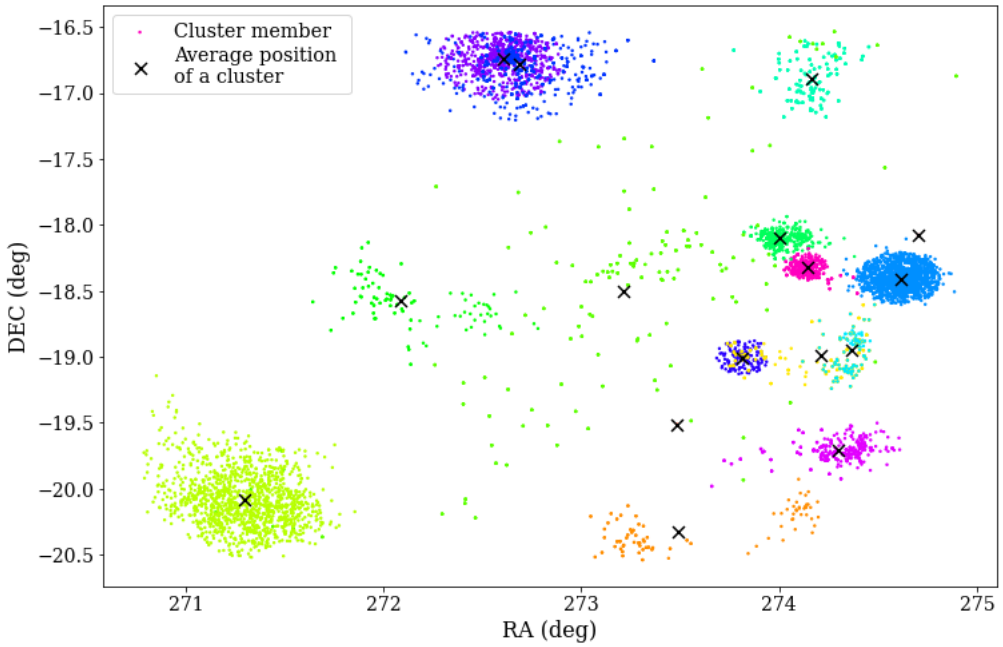



*Figure 3.4*: Previously discovered clusters by other studies for Region A.

| Cluster | RA | DEC | PLX | pmRA | pmDEC |
|---|---|---|---|---|---|
| FSR_0035 | 271.298 | -20.087 | 0.341 | -0.356 | -1.931 |
| NGC_6554 | 272.084 | -18.574 | 0.539 | 0.423 | -0.639 |
| Gulliver_15 | 272.607 | -16.738 | 0.488 | -1.011 | -1.627 |
| NGC_6561 | 272.688 | -16.778 | 0.643 | 0.071 | -0.756 |
| UPK_5 | 273.201 | -18.517 | 1.769 | 0.636 | -8.358 |
| LP_0288 | 273.484 | -19.514 | 0.399 | 0.117 | -2.063 |
| Sgr_OB7 | 273.492 | -20.325 | 0.737 | -0.077 | -0.222 |
| Markarian_38 | 273.815 | -19.008 | 0.543 | 0.643 | -2.248 |
| Liu_1218 | 273.999 | -18.102 | 0.355 | -0.259 | -1.447 |
| Collinder_469 | 274.144 | -18.323 | 0.398 | 0.025 | -1.652 |
| NGC_6596 | 274.166 | -16.891 | 0.589 | 0.771 | -1.841 |
| Turner_2 | 274.212 | -18.993 | 0.553 | 0.621 | -2.277 |
| Dias_5 | 274.298 | -19.712 | 0.741 | 1.697 | -0.873 |
| Turner_3 | 274.365 | -18.944 | 0.568 | 0.535 | -2.398 |
| NGC_6603 | 274.614 | -18.407 | 0.326 | 0.082 | -2.026 |
| LP_2113 | 274.705 | -18.078 | 0.475 | -0.309 | -1.976 |

*Table 3.2*: Average properties of clusters shown in Figure. 3.4

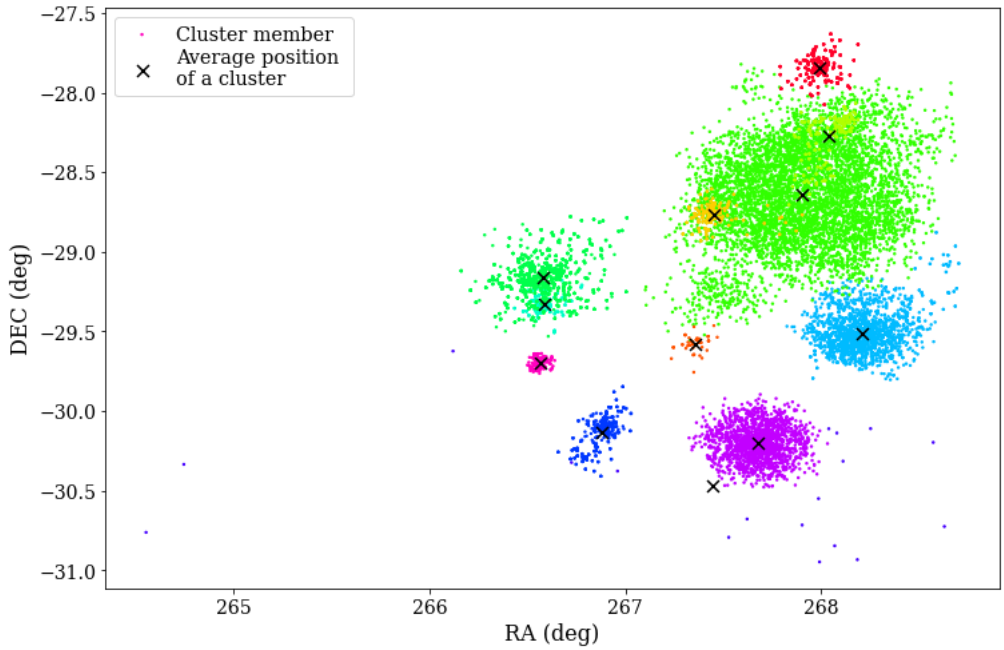



*Figure 3.5*: Previously discovered clusters by other studies for Region B.

| Cluster | RA | DEC | PLX | pmRA | pmDEC |
|---|---|---|---|---|---|
| BH_245 | 266.563 | -29.699 | 0.371 | -0.550 | -2.059 |
| Collinder_347 | 266.587 | -29.325 | 0.322 | 0.406 | -1.848 |
| Liu_1624 | 266.579 | -29.163 | 0.483 | 1.693 | -0.765 |
| NGC_6451 | 267.677 | -30.203 | 0.301 | -0.114 | -1.346 |
| NGC_6475 | 267.448 | -30.471 | 3.484 | 3.082 | -5.523 |
| Ruprecht_130 | 266.876 | -30.135 | 0.391 | 0.382 | -1.721 |
| Ruprecht_133 | 267.903 | -28.645 | 0.324 | -0.315 | -1.906 |
| Ruprecht_134 | 268.208 | -29.512 | 0.355 | -1.616 | -2.498 |
| Ruprecht_168 | 268.040 | -28.272 | 0.289 | -0.151 | -2.042 |
| UBC_336 | 267.988 | -27.841 | 0.354 | 0.680 | 0.131 |
| UBC_571 | 267.354 | -29.579 | 0.337 | -1.062 | -2.882 |
| UBC_91 | 267.454 | -28.765 | 0.417 | -0.575 | -1.187 |

*Table 3.3*: Average properties of clusters shown in Figure 3.5.

Figure 3.5 shows quite a different distribution of open clusters, displaying only a large group where all clusters are in close proximity. In the combined catalogue for this region we determined that vdBergh-Hagen_245 was a duplicate for BH_245, hence it was removed. UBC_570, NGC_6425, Basel_5 and Czernik_37 were also all removed for not being possible to detect by the clustering algorithms, as they were outside in the area covered by the requested *GAIA* data. This leaves a total of 12 open clusters in the region. We immediately see that the open cluster distribution aligns very much with the increase in density of stars in Region B, shown in Figure 3.3.

## 3.3 Parameter selection

For all clustering algorithms, optimal parameters are always very difficult to choose objectively, and a parameter choice can never be completely correct. To fairly test how well the algorithms can detect open clusters, we decide to evaluate their accuracy on detecting clusters within the comparison data, where each algorithm produces the same amount of clusters that are in the comparison data for the respective region. Having input parameters that would produce more clusters than in the comparison data could be beneficial for detecting new clusters. However this comes with a huge increase in number and density of clusters, many of which will be statistical. Without a dedicated program to determine which clusters are correctly predicted to match the comparison data, it would inaccurate to decide subjectively. Ideally we would compare the precision and recall of results across varying input parameters for each algorithm, as conducted by Hunt & Reffert (2021). Unfortunately this is not an option for this work because evaluating performance would subject to inaccurate results, hence we choose a fixed number of clusters for each algorithm to fairly assess performance.

The *min_points* parameter is required for the minimum number of stars required to create a cluster in all three algorithms. As suggested by Ester et al. (1996), this parameter is always set to two times the number of parameters of the data. For example, in the full astrometric solution where there are 5 input parameters, *min_points* would be 10. Additionally, when the $k^{th}$ nearest neighbour distance is calculated, $k$ is always $k = min\_points - 1$, as it already includes the star where the distance is being measured from.

### 3.3.1 DBSCAN

As we previously mentioned in Sect 1.4.1, the $\varepsilon$ parameter for DBSCAN is non-trival to determine, as well as greatly affecting the clustering outcome. Although we will use the value of $\varepsilon$ that will give the same number clusters as in the comparison data, we decide to test the *OCfinder* $\varepsilon$ determination method proposed by Castro-Ginard et al. (2018) to see how it performs. This methodology is explained in detail in Section 2.2. To begin, we determine the bandwidth parameter of a Gaussian kernel density estimator in order to randomly re-sample the data according to the distribution of its parameters. The bandwidth is responsible for smoothing the Gaussian distribution, to create a balance between its bias and variance. This was carried out with *GridSearchCV* from *scikit-learn* using 5 fold cross validation, with scoring done using the maximum log-likelihood estimation.

```
GridSearchCV(KernelDensity(kernel="gaussian"),
                          {"bandwidth": np.arange(0.01,2,0.01)},
                           n_jobs=-1, cv=5)
```

Notably, this process took several hours to run using a 6 core Ryzen 5 5600X CPU, and adds significant time to clustering with DBSCAN if this method is used. Using the optimal

bandwidth, we created KNND arrays by iterating over 30 samples, as in Castro-Ginard et al. (2018). The histograms of each array were averaged and plotted in Figure 3.6, with the original histogram of the data.

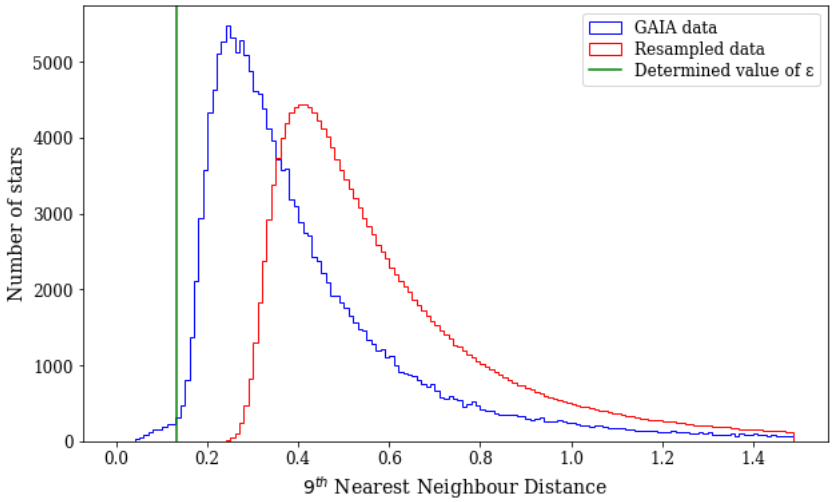



*Figure 3.6*: Castro-Ginard et al. (2018) methodology for $\varepsilon$ determination (Region A).

The methodology of Castro-Ginard et al. (2018) gives quite a high $\varepsilon$ value. This is likely to lead to a very high amount of false positives resulting from statistical clusters. To understand how $\varepsilon$ effects the number of clusters found, different values can be iterated over and the number of clusters recorded. The hand-picked $\varepsilon$ values represent $\varepsilon$ where the number of clusters is 16 (which is the same amount as in the comparison data) and is shown on both the KNND and number of cluster graphs. To speed up this process, as recommended by Hunt & Reffert (2021), the k-distance tree algorithm is specified for DBSCAN. Each iteration is represented as a line graph in Figure 3.7.

```
def get_epstrials(data, min_, max_, step, min_samples):
    epstrials = np.arange(min_, max_, step).tolist()
    e_dict = {}

    for eps in eps_trials:
        dbscan = DBSCAN(eps = eps, min_samples = min_samples,
        algorithm="kd_tree", n_jobs=-1).fit(data)
        num_clusters = len(np.unique(dbscan.labels)) - 1 )
        edict.update({ eps : num_clusters }

    return e_dict
```

The Castro-Ginard et al. (2018) method would produce 53 clusters for region A, a large overestimate to the amount expected. Initially, we assumed this could be to do with data qual-

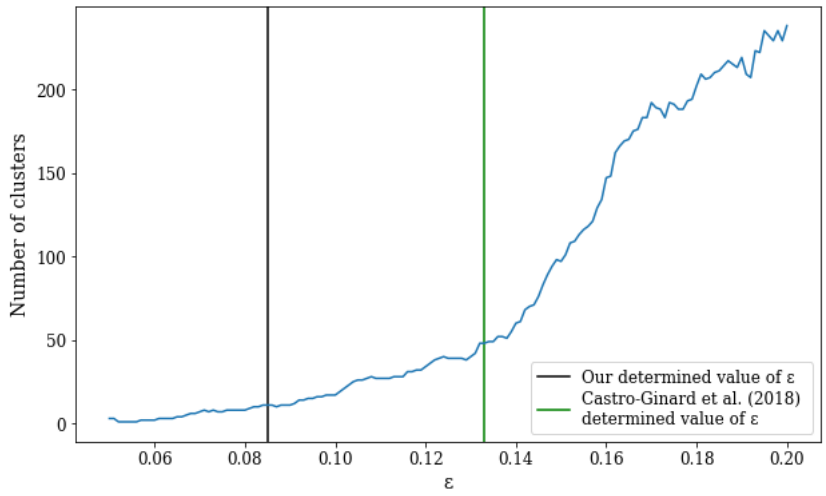



*Figure 3.7*: Number of clusters detected with the full astrometric solution in Region A using our method and the method described in Castro-Ginard et al. (2018).

ity cuts that were made when querying the data, however after testing with far more relaxed requirements for the data quality cuts, similar results were obtained. We theorise this happens because size of the region used is too large, which means the *GridSearchCV* has optimised the bandwidth of the kernel density estimator to include statistical clusters as open clusters. For future research, much smaller regions should be used when attempting this method. Castro-Ginard et al. (2018) does vary the size of the region as a variable parameter, however this is criticised by Hunt & Reffert (2021) for not being very computationally efficient. Unfortunately no valid results were obtained from this method, and the final parameters used (Table 3.4) were all obtained from iterating over $\varepsilon$ values as in Figure 3.7, using our method originally stated in Section 3.3.

| **Parameter** | **Full astrometric solution** | **Radial velocity** | **Absolute magnitude** |
|---|---|---|---|
| eps ($\varepsilon$), Region A | 0.085 | 0.1085 | 0.248 |
| min_points, Region A | 10 | 12 | 12 |
| eps ($\varepsilon$), Region B | 0.096 | | |
| min_points, Region B | 10 | | |

*Table 3.4*: All parameters used with DBSCAN.

### 3.3.2 HDBSCAN

Unlike DBSCAN, HDBSCAN has the more intuative *min_cluster_size* parameter. There is no parameter selection method like with DBSCAN, and the methodology from Castro-Ginard et al. (2018) cannot be applied, as HDBSCAN does not have a set $\varepsilon$ value. We proceed with iterating over varying values of *min_cluster_size*, to plot the amount of clusters detected with the parameter change. HDBSCAN also gives a choice between the *excess_of_mass* or *leaf* clustering

methods. We found that as mentioned in HDBSCAN documentation, *excess_of_mass* gives one or two large clusters hence is useless for this work, meaning the *leaf* method is always used.

```
def get_mcls_trials(data, min_, max_, step, min_samples):
    mcls_trials = np.arange(min_, max_, step).tolist()
    mcs_dict = {}

    for m in mcls_trials:
        hdbscan = hdbscan.HDBSCAN(
        min_samples=min_samples, min_cluster_size = m,
        cluster_selection_method="leaf").fit(data)
        num_clusters = len(np.unique(hdbscan.labels) - 1)
        mcs_dict.update({ h : num_clusters })

    return mcs_dict
```

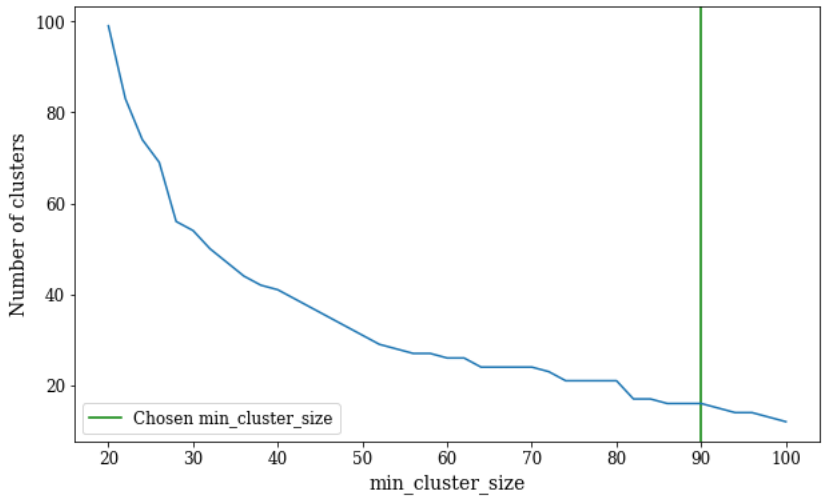



*Figure 3.8*: min_cluster_size parameter selection for Region A using the full astrometric solution.

In Table 3.5, we state all parameters used to generate results with HDBSCAN for the full astrometric solution and with the separate addition of radial velocity and absolute magnitude.

| **Input data** | **min_cluster_size** | **min_points** |
|---|---|---|
| Full astromeric solution, Region A | 90 | 10 |
| Full astromeric solution, Region B | 108 | 10 |
| Radial Velocity, Region A | 120 | 12 |
| Absolute Magnitude, Region A | 35 | 12 |

*Table 3.5*: All parameters used with HDBSCAN.

### 3.3.3 OPTICS

We carry out the OPTICS parameter selection in much the same way as in HDBSCAN. We display a graph in Figure 3.9 to show the number of clusters found for the max_eps parameter. There were not as many values iterated over for OPTICS parameter selection since it quickly came to out attention that performing many iterations of OPTICS was very computationally expensive. Python code to return the number of clusters for each max_eps value is shown below:

```
def get_maxepstrials(data, min, max_, step, min_samples):
    maxepstrials = np.arange(min, max_, step).tolist()
    maxe_dict = {}

    for e in maxeps_trials:
        optics = OPTICS(max_eps = e, min_samples = min_samples,
        algorithm="kd_tree", n_jobs=-1).fit(data)
        num_clusters = len(np.unique(optics.labels) - 1
        edict.update({ e : num_clusters })

    return maxe_dict
```

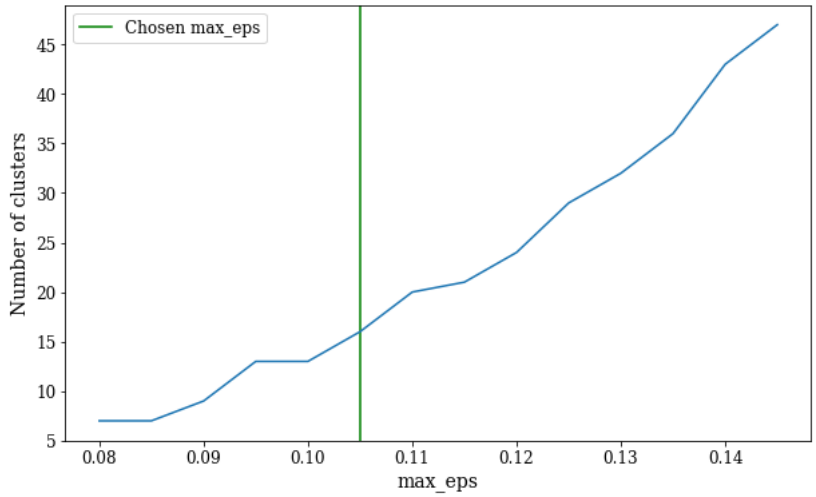



*Figure 3.9*: max_eps parameter selection for Region A using the full astrometric solution.

| Input data | max_eps | min_points |
|---|---|---|
| Full astromeric solution, Region A | 0.105 | 10 |
| Full astromeric solution, Region B | 0.092 | 10 |

*Table 3.6*: All parameters used with OPTICS.

# Chapter 4

# Results and Discussion

To describe the process of our analysis, we present a flowchart showing the basic principles (Figure 4.1). The end product of the analysis is to show graphs of the predicted open clusters from each algorithm, overlayed with the average position of confirmed open clusters from other research, which is done for two regions of space when clustering with only the full astrometric solution. For the full astrometric solution, we also display HR diagrams and view of a single cluster against the background stars for each algorithm, to show the differences in membership assignment between each algorithm. We choose the NGC_6603 cluster for this, characterised in Table 3.4, as it is detected with all three algorithms and has an adequate number of members. Our results are complemented by tables for each algorithm, detailing the average properties of all detected open clusters for the whole analysis, and the amount of stars each cluster contains.

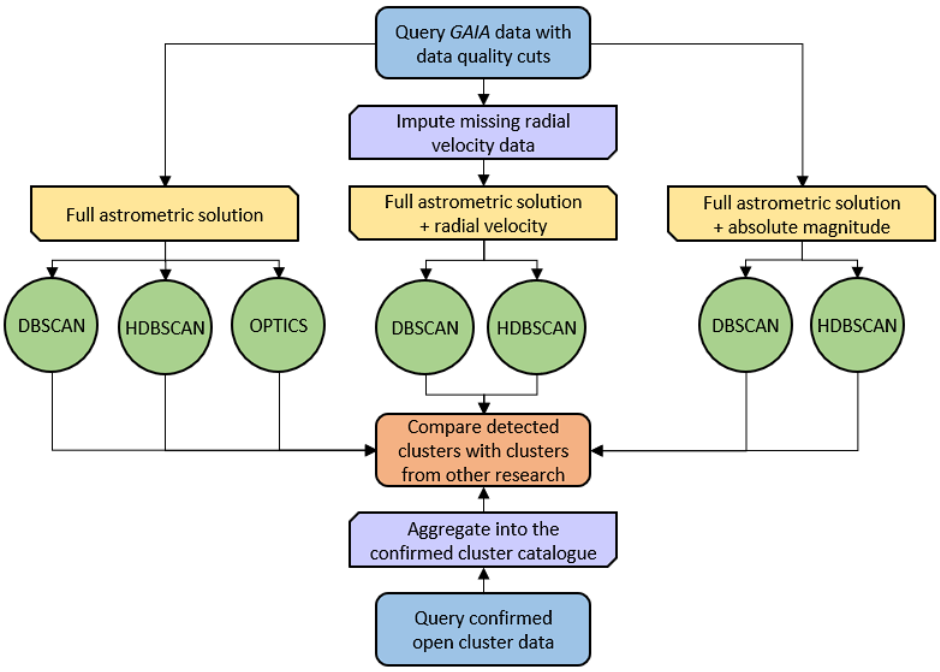



*Figure 4.1*: Flowchart describing data analysis.

## 4.1 Full Astrometric Solution

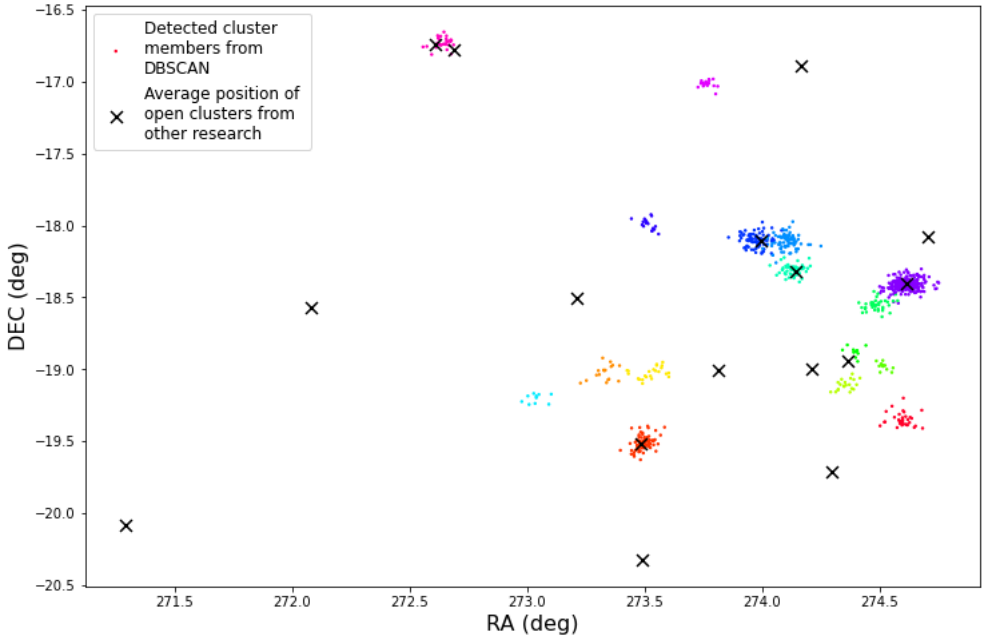



*Figure 4.2*: DBSCAN full astrometric solution clustering results (Region A) compared to previously discovered clusters from other research.

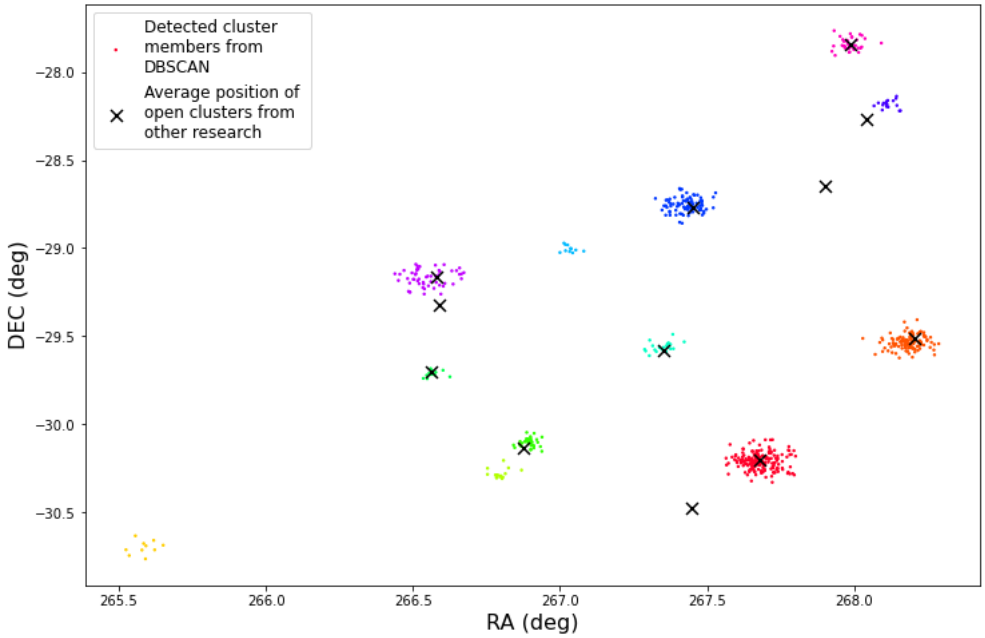
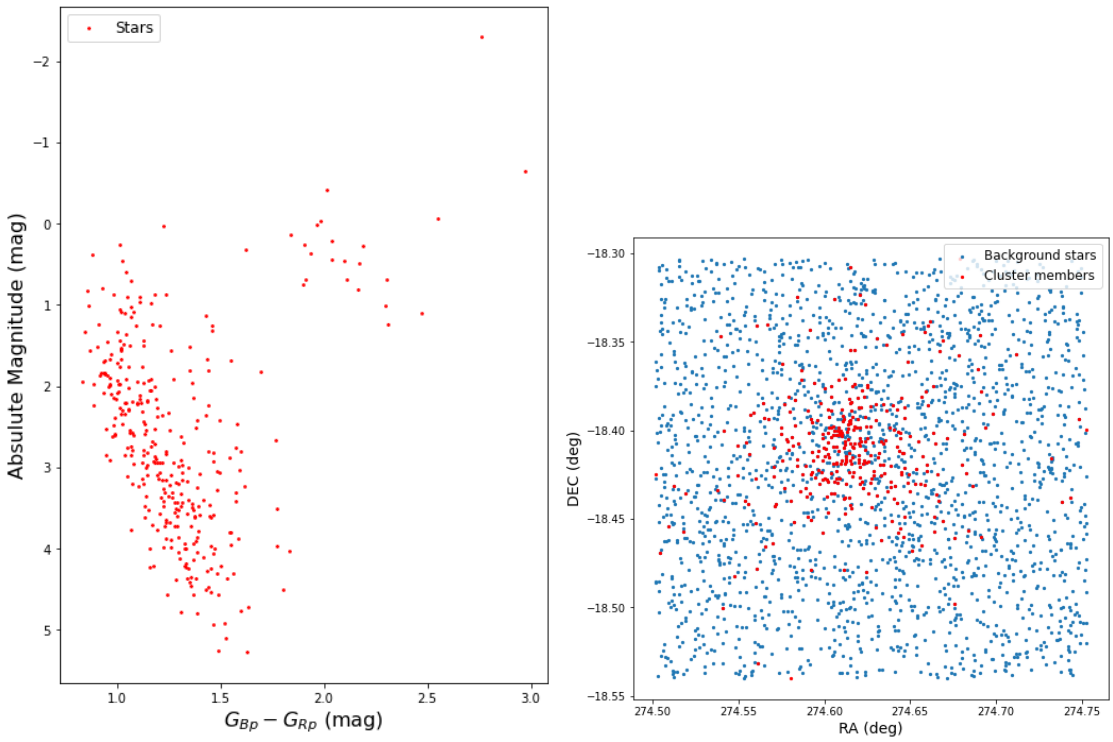



*Figure 4.3*: DBSCAN full astrometric solution clustering results (Region B) compared to previously discovered clusters from other research.

| | Region A | | | | Region B | | | |
|---|---|---|---|---|---|---|---|---|
| **Index** | **RA (deg)** | **DEC (deg)** | **PLX (mas)** | **Stars per cluster** | **RA (deg)** | **DEC (deg)** | **PLX (mas)** | **Stars per cluster** |
| 0 | 272.641 | -16.732 | 0.675 | 29 | 265.585 | -30.698 | 0.623 | 10 |
| 1 | 273.038 | -19.203 | 0.324 | 10 | 266.550 | -29.169 | 0.481 | 53 |
| 2 | 273.330 | -19.023 | 0.341 | 17 | 266.568 | -29.713 | 0.333 | 11 |
| 3 | 273.498 | -19.512 | 0.379 | 82 | 266.796 | -30.282 | 0.367 | 18 |
| 4 | 273.516 | -17.991 | 0.391 | 17 | 266.895 | -30.101 | 0.367 | 35 |
| 5 | 273.526 | -19.025 | 0.328 | 23 | 267.033 | -29.004 | 0.572 | 11 |
| 6 | 273.767 | -17.015 | 0.348 | 22 | 267.348 | -29.560 | 0.375 | 20 |
| 7 | 273.974 | -18.101 | 0.365 | 82 | 267.429 | -28.755 | 0.443 | 85 |
| 8 | 274.105 | -18.111 | 0.353 | 86 | 267.681 | -30.208 | 0.338 | 181 |
| 9 | 274.134 | -18.309 | 0.406 | 57 | 267.986 | -27.843 | 0.336 | 40 |
| 10 | 274.358 | -19.108 | 0.552 | 20 | 268.116 | -28.183 | 0.290 | 19 |
| 11 | 274.396 | -18.877 | 0.552 | 11 | 268.184 | -29.533 | 0.401 | 116 |
| 12 | 274.490 | -18.552 | 0.372 | 36 | | | | |
| 13 | 274.519 | -18.979 | 0.596 | 12 | | | | |
| 14 | 274.599 | -19.349 | 0.507 | 34 | | | | |
| 15 | 274.616 | -18.411 | 0.330 | 339 | | | | |

*Table 4.1*: Average properties of all open clusters detected with DBSCAN.




(a) (b)

*Figure 4.4*: The predicted members of the NGC_6603 open cluster by DBSCAN (Table 4.1 Region A Index 15) shown in an HR diagram in (a) and against background stars in (b).

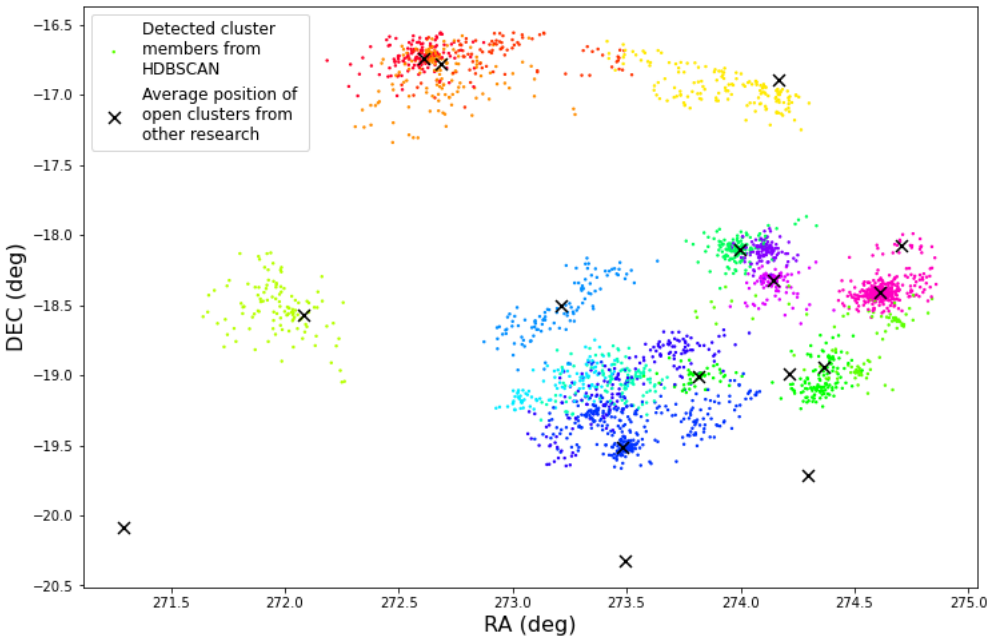



*Figure 4.5*: HDBSCAN full astrometric solution clustering results (Region A) compared to previously discovered clusters from other research.

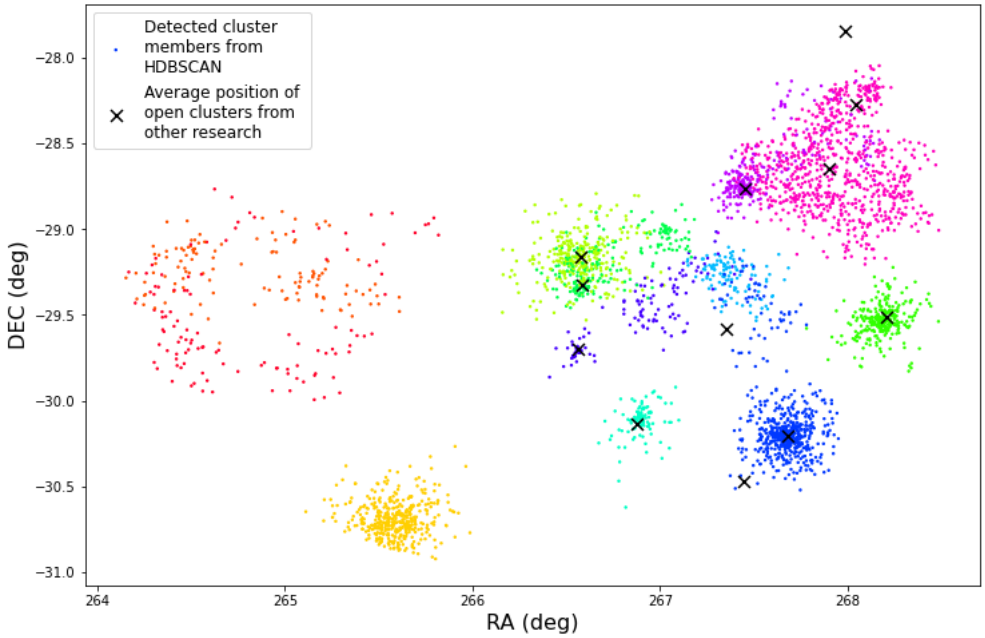
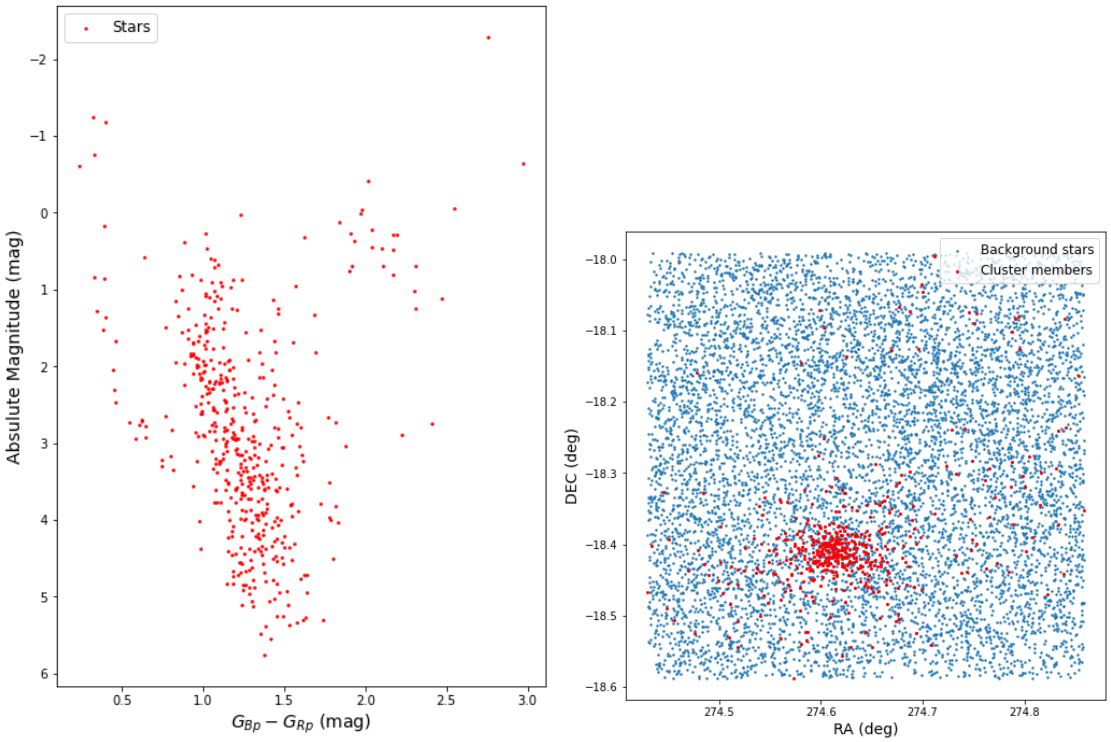



*Figure 4.6*: HDBSCAN full astrometric solution clustering results (Region B) compared to previously discovered clusters from other research.

| | Region A | | | | Region B | | | |
|---|---|---|---|---|---|---|---|---|
| **Index** | **RA (deg)** | **DEC (deg)** | **PLX (mas)** | **Stars per cluster** | **RA (deg)** | **DEC (deg)** | **PLX (mas)** | **Stars per cluster** |
| 0 | 271.973 | -18.527 | 0.521 | 117 | 264.809 | -29.514 | 0.529 | 113 |
| 1 | 272.586 | -16.730 | 0.526 | 137 | 264.817 | -29.248 | 0.509 | 121 |
| 2 | 272.677 | -16.848 | 0.582 | 160 | 265.588 | -30.691 | 0.618 | 400 |
| 3 | 272.958 | -16.725 | 0.594 | 92 | 266.571 | -29.167 | 0.485 | 313 |
| 4 | 273.209 | -18.531 | 0.347 | 105 | 266.735 | -29.191 | 0.457 | 206 |
| 5 | 273.210 | -19.070 | 0.342 | 99 | 266.883 | -29.493 | 0.344 | 122 |
| 6 | 273.449 | -19.043 | 0.383 | 123 | 266.898 | -30.120 | 0.395 | 108 |
| 7 | 273.513 | -19.053 | 0.483 | 173 | 267.381 | -29.279 | 0.378 | 108 |
| 8 | 273.539 | -19.365 | 0.379 | 329 | 267.560 | -28.648 | 0.408 | 234 |
| 9 | 273.930 | -16.947 | 0.485 | 153 | 267.662 | -30.132 | 0.353 | 567 |
| 10 | 273.987 | -18.097 | 0.387 | 137 | 267.947 | -28.636 | 0.340 | 799 |
| 11 | 274.102 | -18.115 | 0.358 | 128 | 268.184 | -29.534 | 0.406 | 303 |
| 12 | 274.148 | -18.352 | 0.402 | 96 | | | | |
| 13 | 274.190 | -19.019 | 0.556 | 178 | | | | |
| 14 | 274.493 | -18.745 | 0.579 | 91 | | | | |
| 15 | 274.627 | -18.394 | 0.359 | 478 | | | | |

*Table 4.2*: Average properties of all open clusters detected with HDBSCAN.




(a) (b)

*Figure 4.7*: The predicted members of the NGC_6603 open cluster by HDBSCAN (Table 4.2 Region A Index 15), shown in an HR diagram in (a) and against background stars in (b).

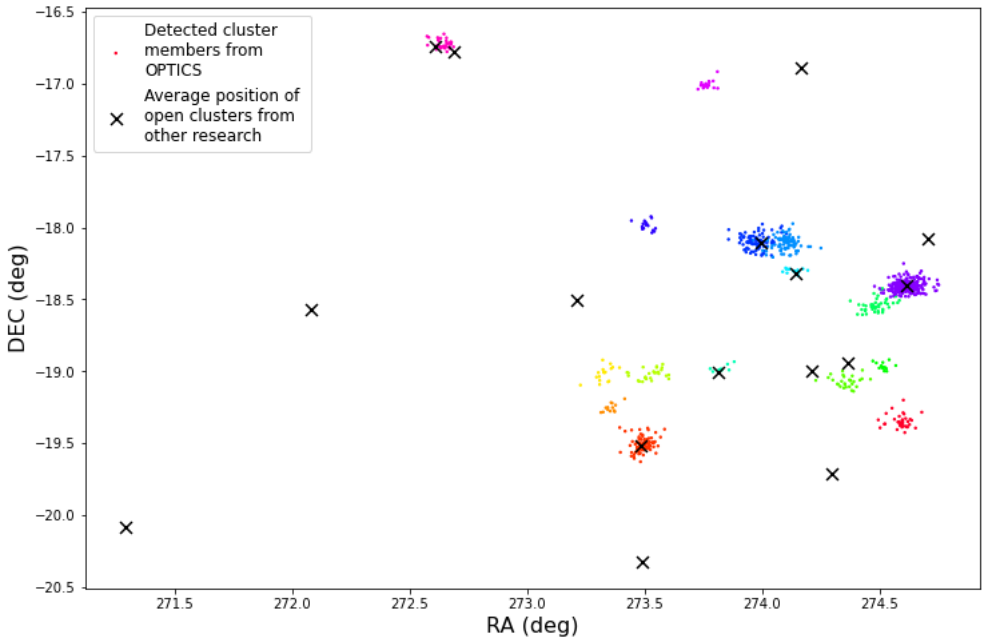



*Figure 4.8*: OPTICS full astrometric solution clustering results (Region A) compared to previously discovered clusters from other research.

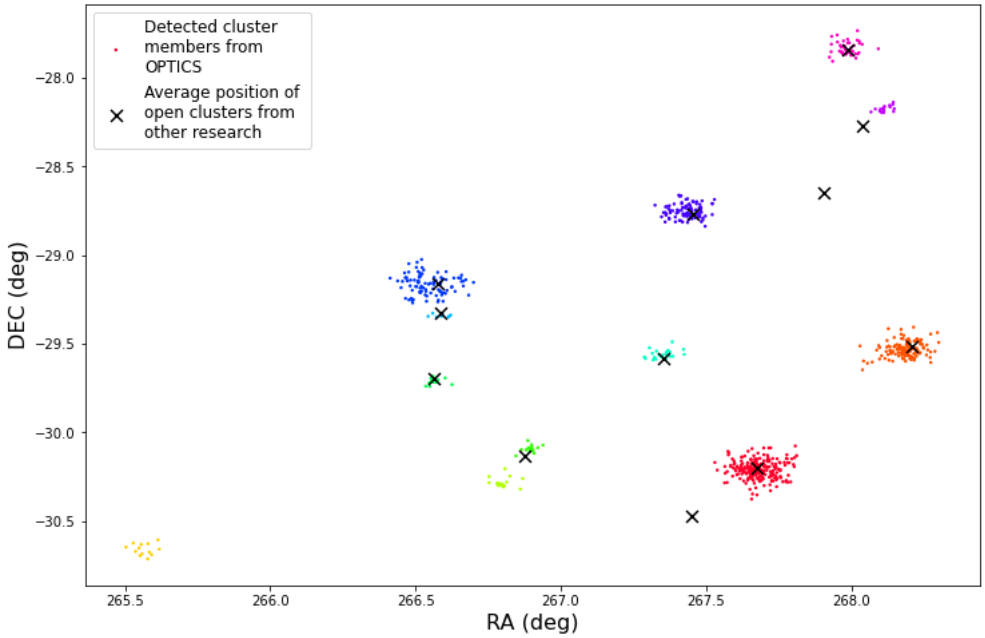
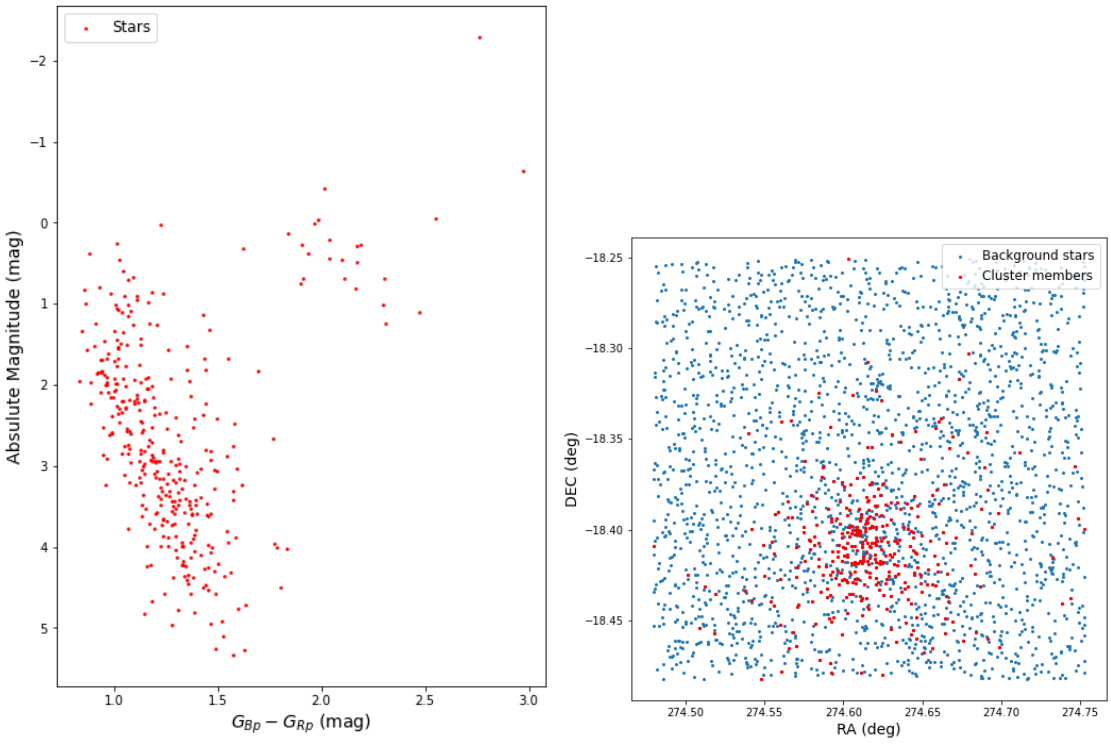



*Figure 4.9*: OPTICS full astrometric solution clustering results (Region B) compared to previously discovered clusters from other research.

| | Region A | | | | Region B | | | |
|---|---|---|---|---|---|---|---|---|
| **Index** | **RA (deg)** | **DEC (deg)** | **PLX (mas)** | **Stars per cluster** | **RA (deg)** | **DEC (deg)** | **PLX (mas)** | **Stars per cluster** |
| 0 | 272.643 | -16.725 | 0.674 | 33 | 265.565 | -30.659 | 0.646 | 13 |
| 1 | 273.327 | -19.017 | 0.344 | 17 | 266.545 | -29.166 | 0.480 | 96 |
| 2 | 273.357 | -19.253 | 0.352 | 13 | 266.567 | -29.711 | 0.329 | 13 |
| 3 | 273.495 | -19.509 | 0.380 | 91 | 266.588 | -29.340 | 0.355 | 11 |
| 4 | 273.511 | -17.985 | 0.385 | 16 | 266.800 | -30.282 | 0.366 | 18 |
| 5 | 273.527 | -19.019 | 0.326 | 26 | 266.895 | -30.091 | 0.374 | 17 |
| 6 | 273.767 | -17.008 | 0.348 | 23 | 267.351 | -29.558 | 0.377 | 22 |
| 7 | 273.828 | -18.981 | 0.567 | 10 | 267.433 | -28.751 | 0.441 | 90 |
| 8 | 273.976 | -18.097 | 0.364 | 91 | 267.680 | -30.208 | 0.341 | 227 |
| 9 | 274.099 | -18.108 | 0.352 | 98 | 267.986 | -27.833 | 0.344 | 45 |
| 10 | 274.137 | -18.306 | 0.417 | 18 | 268.115 | -28.175 | 0.294 | 18 |
| 11 | 274.360 | -19.070 | 0.552 | 29 | 268.183 | -29.531 | 0.402 | 131 |
| 12 | 274.488 | -18.539 | 0.372 | 42 | | | | |
| 13 | 274.518 | -18.965 | 0.606 | 14 | | | | |
| 14 | 274.592 | -19.348 | 0.507 | 37 | | | | |
| 15 | 274.617 | -18.409 | 0.331 | 350 | | | | |

*Table 4.3*: Average properties of all open clusters detected with OPTICS.




(a) (b)

*Figure 4.10*: The predicted members of the NGC_6603 open cluster by OPTICS (Table 4.3 Region A Index 15), shown in an HR diagram in (a) and against background stars in (b).

Prior to beginning the discussion, we define the following metric to measure the precision of the algorithms to predict the same clusters detected by other research, where $TP$ is True Positive, and $FP$ is False Positive:

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

The success or failure of an algorithm to predict a cluster correctly is determined by hand. This requires care to be taken looking at the cluster position with respect to RA, DEC and parallax. If a cluster appears to be in the same position as an existing cluster on the displayed graphs, it does not necessarily have to be predicting the correct one, due to the graph only showing two dimensional space which does not include parallax. For determining if a predicted cluster is the same cluster as in the comparison data, we choose to use a similar method to Hunt & Reffert (2021), which is to label it as a correct prediction if it is within one tidal radius of the cluster from the comparison data.

| **Algorithm** | **Region A Precision (%)** | **Region B Precision (%)** | **Average Precision (%)** |
|---|---|---|---|
| DBSCAN | 50 | 75 | 63 |
| HDBSCAN | 70 | 75 | 75 |
| OPTICS | 56 | 80 | 68 |

*Table 4.4*: Percentage of clusters found by each algorithm using the full astrometric solution that are found by other research.

It is clear through the displayed graphs and Table 4.4 that every algorithm has had some degree of success in correctly finding clusters that are in the comparison data. Firstly, we compare DBSCAN and OPTICS, since they are both similar algorithms in principle, and initially we see they both give very similar results. There are slight deviations in precision with OPTICS being better in both regions, but crucially it is trivial to observe upon first glance that the predicted clusters compared between Figures 4.2 and 4.3, and from Figures 4.8 and 4.9 are almost identical. In fact, apart from a single exception in both regions, they find the exact same clusters. Upon closer observation from Figures 4.4a, 4.10a and Figures 4.4b and 4.10b, the predicted members of each cluster can differ, however this could just due to the chosen input parameters and not the algorithm fundamentally. Having stated the similarity in results for these two algorithms, it is also imperative to acknowledge the severe disadvantage of OPTICS, being its computational cost. Running with on a single CPU core, the OPTICS operation to cluster stars in Region A with parameters stated in Section 3.3.3, took approximately 50 times longer than DBSCAN, also running with parameters stated in Section 3.3.1. For unknown reasons, the OPTICS algorithm was also unable to take advantage of multiple CPU cores, unlike DBSCAN, which would have increased computation speed. This makes it very difficult to recommend OPTICS for operations on the *GAIA* database for identifying open clusters, even if it does possess

a slight increase in performance. The computational cost is especially problematic since the difference in performance will increase with the square of the data points as input to the algorithm. Clustering with OPTICS clearly shows why there are no records of it being used to detect clusters within the *GAIA* dataset. As such, we will not include OPTICS when testing clustering performance with the addition of radial velocity and absolute magnitude data. Furthermore, we cannot compare the performance of OPTICS with any other documented work with respect to *GAIA*, since to the best of our knowledge, there isn't any.

HDBSCAN is a completely different algorithm, with no discrete density cut off for defining its clusters. This can clearly be perceived in Figures 4.5 and 4.6, by the way it captures cluster members towards the edge of a cluster where the density may be less than in the center. We show this even more vividly in Figure 4.7b, which shows a cluster membership assignment capturing a much larger area of space. This is contrast to DBSCAN and OPTICS which show much smaller, condensed clusters because the density in these clusters cannot decrease past a certain limit, defined by the input parameters of the algorithms. The effect of this in Region A for instance, is approximately three times more stars being assigned cluster membership in HDBSCAN than in DBSCAN or OPTICS. This is also be seen through Table 4.2, which shows this increase in membership assignment through the number of stars per cluster, which is visibly higher than DBSCAN or OPTICS. Having said this, we note that the HR diagram of the cluster shown by Figure 4.7a does show a less defined main sequence, which may suggest that it could be overly sensitive to assigning cluster membership to stars that are in fact background stars. Interestingly, this hypothesis agrees with the findings of Ou et al. (2022), who emphasise that without factoring in uncertainties in HDBSCAN, membership assignment is unreliable. Despite this, HDBSCAN does boast a higher precision than DBSCAN and OPTICS, being able to detect more of the clusters in the comparison data across both regions, especially in Region A. Our precision results for HDBSCAN and DBSCAN are very similar to the results found by Hunt & Reffert (2021), which we mention in Section 2.2. We note that we would most likely be able to increase this even higher for HDBSCAN by decreasing the minimum cluster size parameter to include smaller clusters. All the clusters in the comparison data that are undetected by HDBSCAN are small, and are undetected because of the low sensitivity to statistical clusters set by the input parameters. Tuning the algorithm to to detect these smaller clusters would drastically increase the amount of false positives, but would also give the chance to increase its precision.

Another fascinating ability of HDBSCAN is its ability to successfully detect the cluster Ruprecht_133 (Figure 4.6), also given by Index 10 Region B in Table 4.2. We note that this cluster escaped detection from DBSCAN and OPTICS, despite its very high volume. This directly points towards the advantage of using HDBSCAN as a clustering algorithm. Ruprecht_133 clearly has a lower density than other clusters detected by HDBSCAN, being one of the largest by volume, but not by cluster members. The density of this cluster must be beyond the density threshold of DSBCAN and OPTICS, and would require a raise in the $\varepsilon$ and $max\_eps$ parameters in order detect, which would in turn induce far more unwanted statistical clusters.

## 4.2 Radial Velocity

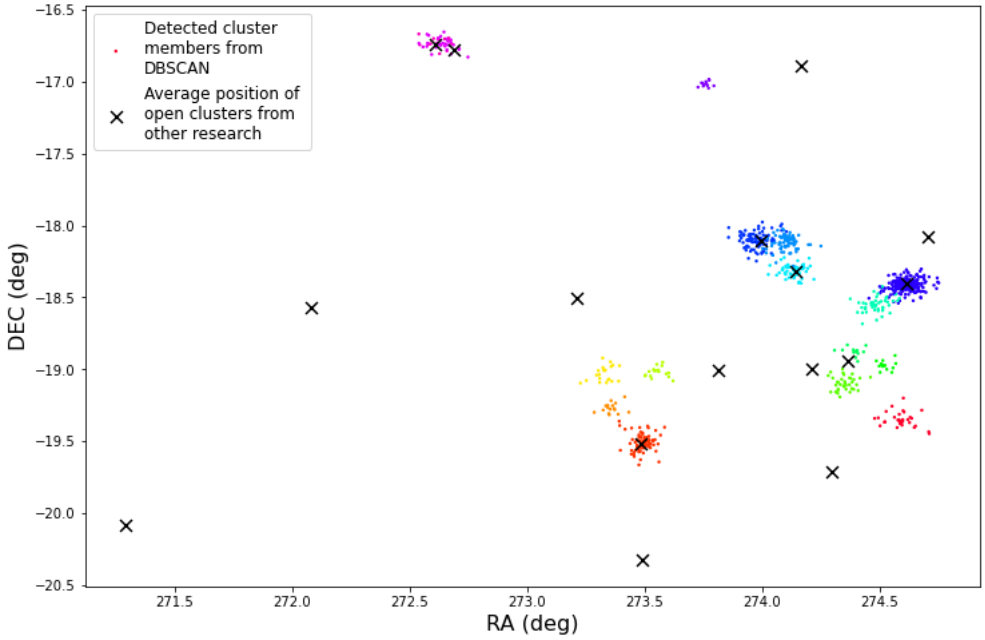



*Figure 4.11*: DBSCAN full astrometric solution with radial velocity clustering results (Region A) compared to previously discovered clusters from other research.

| Index | RA (deg) | DEC (deg) | PLX (mas) | pmRA (mas/yr) | pmDEC (mas/yr) | R_Vel (km/s) | Stars per cluster |
|---|---|---|---|---|---|---|---|
| 0 | 272.619 | -16.729 | 0.518 | -1.003 | -1.664 | -4.747 | 13 |
| 1 | 272.638 | -16.739 | 0.669 | -0.002 | -0.783 | -4.747 | 31 |
| 2 | 273.329 | -19.030 | 0.342 | -0.852 | -1.800 | -4.747 | 21 |
| 3 | 273.361 | -19.270 | 0.361 | 0.105 | -2.307 | -4.747 | 16 |
| 4 | 273.494 | -19.511 | 0.381 | 0.161 | -2.160 | -4.747 | 87 |
| 5 | 273.557 | -19.020 | 0.333 | -0.446 | -1.207 | -4.747 | 17 |
| 6 | 273.762 | -17.017 | 0.365 | 1.154 | -1.026 | -4.747 | 12 |
| 7 | 273.976 | -18.101 | 0.365 | 0.006 | -1.205 | -4.747 | 94 |
| 8 | 274.101 | -18.117 | 0.358 | -1.112 | -2.641 | -4.747 | 84 |
| 9 | 274.136 | -18.313 | 0.407 | 0.082 | -1.689 | -4.747 | 62 |
| 10 | 274.353 | -19.097 | 0.551 | 0.682 | -2.447 | -4.747 | 37 |
| 11 | 274.392 | -18.877 | 0.553 | 0.517 | -2.474 | -4.747 | 13 |
| 12 | 274.488 | -18.549 | 0.372 | -0.021 | -0.495 | -4.747 | 45 |
| 13 | 274.526 | -18.972 | 0.596 | 1.008 | -1.844 | -4.747 | 14 |
| 14 | 274.593 | -19.349 | 0.514 | -0.718 | -0.039 | -4.747 | 33 |
| 15 | 274.616 | -18.413 | 0.331 | 0.147 | -2.056 | -4.747 | 332 |

*Table 4.5*: All open clusters detected with DBSCAN in Region A using the full astrometric solution and radial velocity.

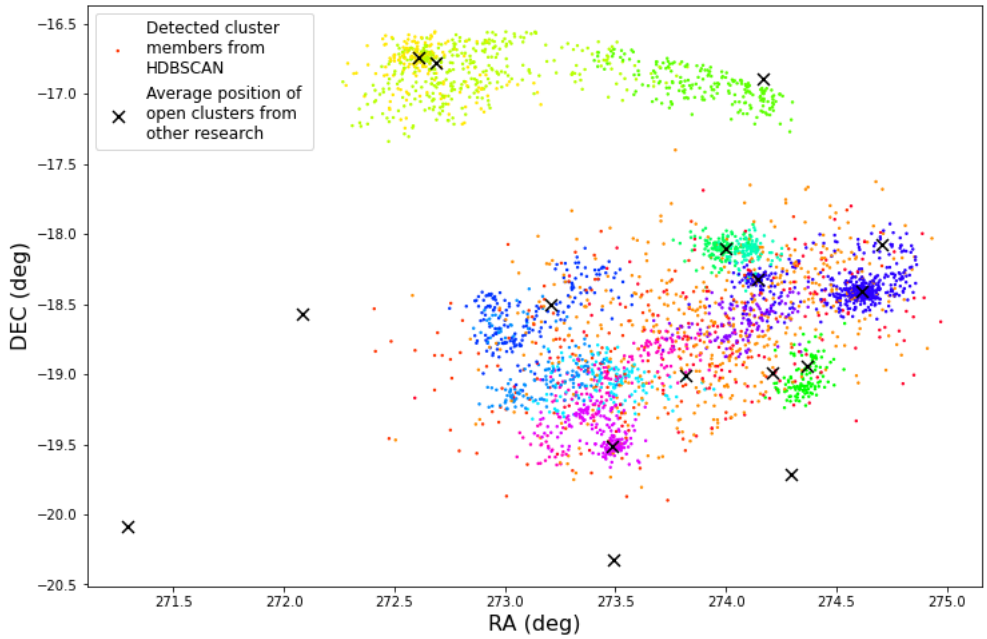



*Figure 4.12*: HDBSCAN full astrometric solution with radial velocity clustering results (Region A) compared to previously discovered clusters from other research.

| **Index** | **RA (deg)** | **DEC (deg)** | **PLX (mas)** | **pmRA (mas/yr)** | **pmDEC (mas/yr)** | **R_Vel (km/s)** | **Stars per cluster** |
|---|---|---|---|---|---|---|---|
| 0 | 272.639 | -16.804 | 0.512 | -0.828 | -1.769 | -4.747 | 179 |
| 1 | 272.849 | -16.836 | 0.563 | 0.469 | -1.215 | -4.747 | 353 |
| 2 | 273.121 | -18.576 | 0.358 | 0.057 | -1.775 | -4.747 | 181 |
| 3 | 273.162 | -18.966 | 0.344 | -0.778 | -2.083 | -4.747 | 154 |
| 4 | 273.439 | -19.421 | 0.391 | 0.073 | -2.178 | -4.747 | 199 |
| 5 | 273.468 | -19.055 | 0.378 | -0.225 | -1.225 | -4.747 | 141 |
| 6 | 273.498 | -19.056 | 0.487 | 0.296 | -1.591 | -4.747 | 164 |
| 7 | 273.499 | -18.823 | 0.409 | -0.194 | -1.827 | -19.053 | 163 |
| 8 | 273.920 | -16.948 | 0.489 | 0.163 | -1.784 | -4.747 | 194 |
| 9 | 273.970 | -18.634 | 0.376 | -0.342 | -2.138 | 15.114 | 516 |
| 10 | 273.974 | -18.100 | 0.380 | 0.009 | -1.198 | -4.747 | 128 |
| 11 | 274.066 | -18.640 | 0.368 | -0.571 | -2.131 | -4.747 | 129 |
| 12 | 274.081 | -18.697 | 0.335 | -0.524 | -2.323 | 41.960 | 153 |
| 13 | 274.103 | -18.120 | 0.359 | -1.109 | -2.644 | -4.747 | 120 |
| 14 | 274.355 | -19.020 | 0.556 | 0.598 | -2.432 | -4.747 | 121 |
| 15 | 274.542 | -18.378 | 0.377 | 0.118 | -1.986 | -4.747 | 644 |

*Table 4.6*: All open clusters detected with HDBSCAN in Region A using the full astrometric solution and radial velocity.

| Algorithm | Precision (%) |
|---|---|
| DBSCAN | 50 |
| HDBSCAN | 63 |

*Table 4.7*: Percentage of clusters found by each algorithm using the full astrometric solution and radial velocity that are found by other research.

Clustering making use of radial velocity data with DBSCAN surprisingly makes no difference to its precision of detecting clusters in the comparison data. Upon closer observation of Table 4.5, we notice that the algorithm has not actually used any radial velocity values to cluster with. It appears that the amount of missing data is too great, and all clusters have an average radial velocity of the mean value all radial velocity data, which was imputed in the pre-processing stage. HDBSCAN shows far different results, as it does actually manage to use radial velocity data for clustering. We observe three clusters with a huge volume, given by index numbers 8, 9 and 15 in Table 4.6. These are all the clusters which the algorithm has grouped together based on similar radial velocity values. Unfortunately, we can see that these are not real physical clusters, given their size, dispersion of stars, as well as parallaxes that are not large enough to allow this. Being able to detect these clusters highlights natural precision of HDBSCAN, which can be very advantageous. Regrettably there is not much else to comment on regarding our results. We can only state that clustering with radial velocity data in *GAIA* DR3 would require high ingenuity to create a method of incorporating it to yield beneficial results.

Using radial velocity as an additional parameter along with the full astrometric solution is a very novel technique, only being viable since the release of *GAIA* DR3. There are very few records of radial velocity being used in clustering algorithms for detecting open clusters, most likely because of the lack of data. Ye et al. (2022) uses radial velocity data for open cluster membership refinement, where *GAIA* DR3 data would be adequate, but Alegre Aldeano (2022) do report using radial velocity for clustering. Unfortunately, they do not state their approach in handling the missing data, but do yield results from the clustering. We can only assume that the region of space used by Alegre Aldeano (2022) did have ample stars with radial velocity measurements, since in our experiments removing all stars without radial velocity measurements resulted in no clusters being detected. Radial velocity remains challenging to implement into clustering algorithms in *GAIA* DR3.

## 4.3 Absolute Magnitude

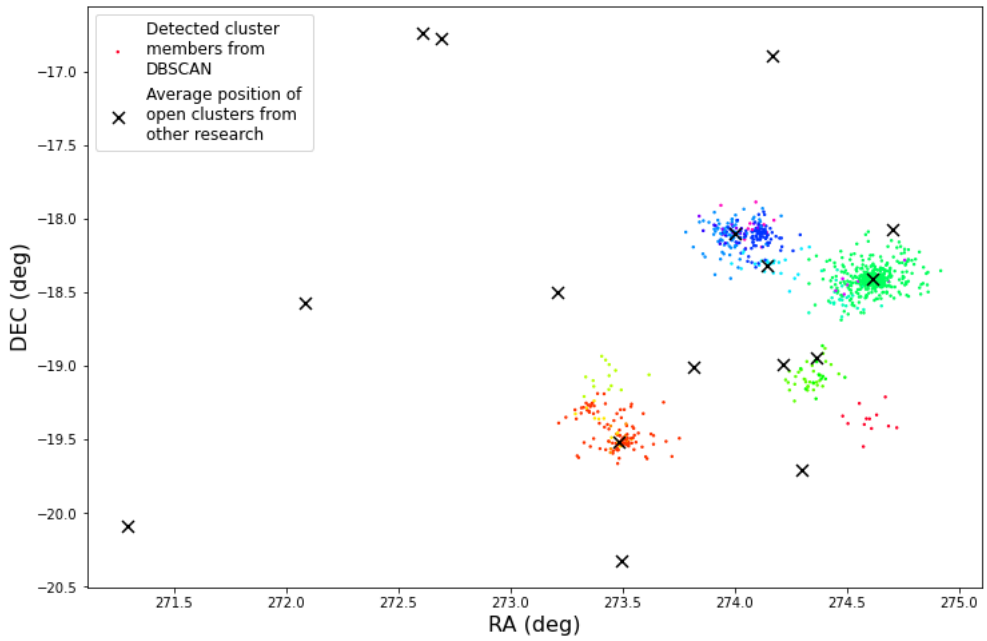



*Figure 4.13*: DBSCAN full astrometric solution with absolute magnitude clustering results (Region A) compared to previously discovered clusters from other research.

| Index | RA (deg) | DEC (deg) | PLX (mas) | pmRA (mas/yr) | pmDEC (mas/yr) | Magnitude (mag) | Stars per cluster |
|---|---|---|---|---|---|---|---|
| 0 | 273.426 | -19.414 | 0.342 | 0.142 | -2.156 | 2.031 | 12 |
| 1 | 273.428 | -19.080 | 0.510 | 0.340 | -1.571 | 5.028 | 13 |
| 2 | 273.471 | -19.449 | 0.367 | 0.153 | -2.183 | 3.515 | 101 |
| 3 | 273.476 | -19.495 | 0.421 | 0.127 | -2.191 | 4.374 | 9 |
| 4 | 273.966 | -18.099 | 0.363 | 0.007 | -1.199 | 2.240 | 13 |
| 5 | 273.977 | -18.109 | 0.365 | 0.018 | -1.217 | 3.366 | 85 |
| 6 | 273.991 | -18.074 | 0.380 | 0.005 | -1.187 | 1.646 | 12 |
| 7 | 274.069 | -18.066 | 0.418 | -1.196 | -2.632 | 4.658 | 13 |
| 8 | 274.106 | -18.121 | 0.348 | -1.109 | -2.644 | 3.211 | 93 |
| 9 | 274.159 | -18.313 | 0.401 | 0.065 | -1.674 | 3.215 | 25 |
| 10 | 274.330 | -19.078 | 0.541 | 0.646 | -2.386 | 4.495 | 31 |
| 11 | 274.373 | -19.059 | 0.552 | 0.643 | -2.432 | 5.014 | 13 |
| 12 | 274.479 | -18.537 | 0.365 | -0.021 | -0.482 | 3.320 | 24 |
| 13 | 274.590 | -18.420 | 0.321 | 0.132 | -2.045 | 0.351 | 15 |
| 14 | 274.599 | -19.372 | 0.471 | -0.700 | -0.052 | 4.148 | 12 |
| 15 | 274.615 | -18.407 | 0.348 | 0.141 | -2.056 | 3.024 | 497 |

*Table 4.8*: All open clusters detected with DBSCAN in Region A using the full astrometric solution and absolute magnitude.

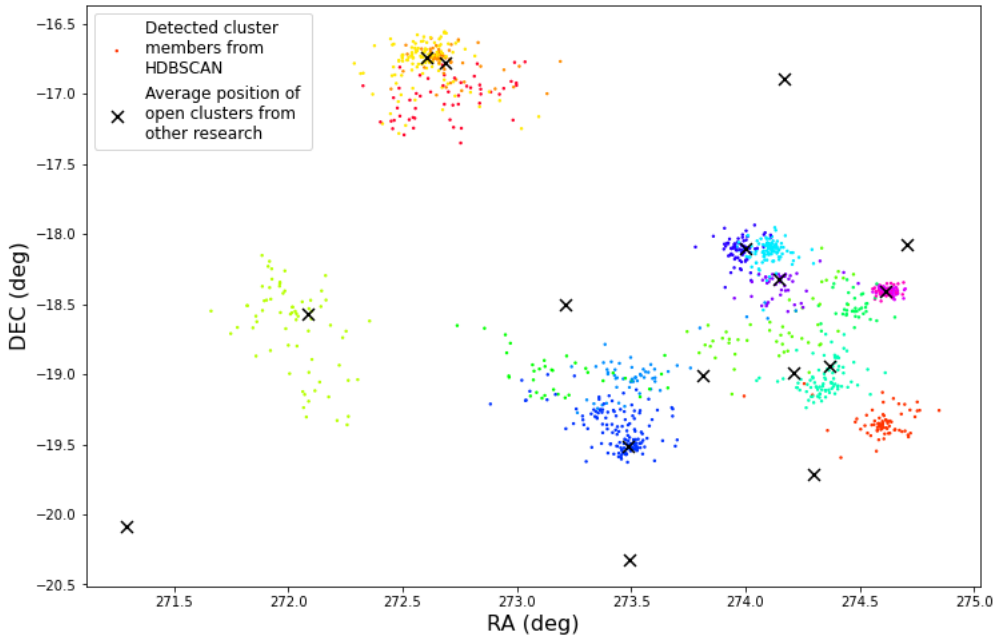



*Figure 4.14*: HDBSCAN full astrometric solution with absolute magnitude clustering results (Region A) compared to previously discovered clusters from other research.

| **Index** | **RA (deg)** | **DEC (deg)** | **PLX (mas)** | **pmRA (mas/yr)** | **pmDEC (mas/yr)** | **Magnitude (mag)** | **Stars per cluster** |
|---|---|---|---|---|---|---|---|
| 0 | 272.027 | -18.669 | 0.512 | 0.313 | -1.095 | 4.805 | 70 |
| 1 | 272.599 | -16.780 | 0.520 | -0.890 | -1.606 | 4.330 | 118 |
| 2 | 272.689 | -17.022 | 0.444 | -0.694 | -2.997 | 4.507 | 54 |
| 3 | 272.698 | -16.773 | 0.618 | 0.032 | -0.845 | 3.911 | 59 |
| 4 | 273.204 | -18.991 | 0.412 | -0.610 | -2.922 | 4.443 | 36 |
| 5 | 273.443 | -19.403 | 0.376 | 0.040 | -2.202 | 3.628 | 148 |
| 6 | 273.535 | -18.993 | 0.346 | -0.253 | -1.209 | 3.352 | 55 |
| 7 | 273.976 | -18.110 | 0.361 | 0.016 | -1.212 | 3.308 | 77 |
| 8 | 274.103 | -18.123 | 0.355 | -1.113 | -2.644 | 3.318 | 109 |
| 9 | 274.138 | -18.670 | 0.381 | -1.020 | -3.309 | 3.671 | 48 |
| 10 | 274.157 | -18.367 | 0.392 | 0.064 | -1.653 | 3.530 | 46 |
| 11 | 274.346 | -19.031 | 0.541 | 0.603 | -2.435 | 4.536 | 84 |
| 12 | 274.474 | -18.522 | 0.377 | -0.034 | -0.490 | 3.429 | 38 |
| 13 | 274.593 | -19.344 | 0.489 | -0.699 | -0.046 | 4.338 | 74 |
| 14 | 274.616 | -18.409 | 0.337 | 0.133 | -2.054 | 3.294 | 49 |
| 15 | 274.624 | -18.409 | 0.303 | 0.166 | -2.060 | 2.099 | 48 |

*Table 4.9*: All open clusters detected with HDBSCAN in Region A using the full astrometric solution and absolute magnitude.

| Algorithm | Precision (%) |
|---|---|
| DBSCAN | 31 |
| HDBSCAN | 56 |

*Table 4.10*: Percentage of clusters found by each algorithm using the full astrometric solution and absolute magnitude that are found by other research.

Clustering with absolute magnitude promised some interesting results and has never been attempted before in published literature. Disappointingly, we see drops in precision from both DBSCAN and HDBSCAN. Whilst some clusters are still predicted correctly, the underlying problem with absolute magnitude clustering in both algorithms is that they are splitting a single physical cluster into multiple clusters according to variations in absolute magnitude within that cluster. This effect is more prominent in DBSCAN, and examples can be observed in Table 4.8 at indexes 8,9 and 10 or at indexes 14 and 15. We may expect HBDSCAN to see this effect more given its sensitivity to clustering with the radial velocity data, yet this is not the case, and it gives a higher sensitivity than DBSCAN to predicting the comparison data. The results from these algorithms are logical, given that the absolute magnitudes within a cluster should vary slightly. Despite being aware of absolute magnitude not being a defining characteristic of stars within an open cluster, it was still interesting to test even without yielding beneficial results.

# Chapter 5

# Conclusions

## 5.1 Summary

In this work we aimed to use three density based clustering algorithms to identify open clusters within the Milky Way. We first queried the data from the *GAIA* database using the Astronomical Database Query Language. The data was queried with data cuts to yield good quality, precise data and passed through the necessary pre-processing steps. We aggregated previously discovered clusters from Cantat-Gaudin et al. (2018), Hunt & Reffert (2021), Dias et al. (2021) and Castro-Ginard et al. (2022) to form comparison data to evaluate the performance of each algorithm. The parameters chosen for each clustering operation were calculated to give the same number of clusters as in the comparison data, so as to fairly compare their predictive performance. All algorithms found a significant percentage of the open clusters within the comparison data and we are able to verify the successful detection of the NGC_6603 cluster with Hertzsprung–Russell diagrams. Initially, we used the full astrometric solution as input parameters for the clustering algorithms, consisting of the radial ascension and declination coordinates, parallax angle, and proper motion with respect to radial ascension and declination. We then experimented with the addition of radial velocity, followed by absolute magnitude data to assess their impact on clustering performance.

We determined that HDBSCAN was the algorithm which had the highest accuracy to the comparison data, and produced the most realistic cluster membership assignment for individual stars. Having said this we would like to express concern that the membership assignment may not be as accurate to true positives as other algorithms, following the results of the HR diagram in Figure 4.7a. HDBSCAN does incur a processing speed decrease when compared to DBSCAN, however we reason that the increase in performance makes this sacrifice worthwhile. DBSCAN proves why it has been such a popular choice of algorithm to discover open clusters. We find it produces good stable results and does so at an inexpensive computational cost. Whilst the membership assignment to the clusters are not as realistic as HDBSCAN, DBSCAN does have the advantage of detecting smaller clusters without a drastic increase in the detection of statistical clusters. We conclude that OPTICS is not an appropriate algorithm for use on this

data. The huge demand of computational power is not worth the performance yields that cannot be said to be significantly different to DBSCAN, and we strongly recommend that this algorithm should not be used as a data mining tool on large datasets such as *GAIA*.

Clustering with the addition of radial velocity data to the full astrometric solution did not provide any benefit. We show that the huge portion of missing data makes using it for clustering severely inadequate. Both DBSCAN and HDBSCAN show decreases in predictive accuracy, and did not make use of the radial velocity data in the correct way because of the vast amount of imputed data. Multiple imputation methods may be used improve this, but the only realistic method we see for beneficially making use of radial velocity data would be to query a specific region of space with enough measurements, as we assume was done by Alegre Aldeano (2022).

For the first time in published literature, we attempted to detect open clusters with clustering algorithms making use of absolute magnitude data alongside the full astrometric solution. Although absolute magnitudes of stars do not directly characterise open clusters, we nevertheless investigated its clustering ability with curiosity. Unfortunately, results from DBSCAN and HDBSCAN show the splitting of real physical clusters into smaller sub-clusters, which were grouped according to the variation in absolute magnitude of the cluster members. We conclude that the implementation of absolute magnitude alongside the full astrometric solution in clustering algorithms does not provide any benefit to identifying open clusters.

## 5.2 Future Research

Discovering new open clusters is an incredibility exciting prospect, and the possibility of making these discoveries will only increase as more accurate data is released. Optimising clustering algorithms to detect open clusters is of extreme importance to achieve any reasonable results. In this work, we limit the amount of clusters which are being detected to equate to the amount in the comparison data, whereas in reality the algorithms should be allowed more freedom to discover a greater number. The problem with doing this is that algorithms are extremely prone to detecting statistical clusters, and if allowed to do so, would've made it too difficult to judge if a cluster was predicted correctly as in the comparison data for this work. There is a critical need for an automated statistical cluster removal system to be used as standard practice, such as one devised by Hunt & Reffert (2021), which can efficiently distinguish between a physical and statistical cluster. Ultimately, the most reliable method for this would be an artificial neural network or a supervised machine learning tool, however creating one with much simpler conditions which require less time and computational power would be highly advantageous for a database as large as *GAIA*.

Another pressing need in this field of research is the need for a universally accepted open cluster database, which would contain confirmed and unconfirmed clusters. As we previously alluded too, the process of accumulating previously discovered open clusters from different catalogues which are all presented differently is not only extremely inefficient, but makes it

impossible to be certain if an open cluster candidate has already been proposed before. A universal catalogue would be an investment sure to provide widespread benefits across the scientific community, and eventually lead to more open cluster discoveries.

In this work we attempted to use radial velocity and absolute magnitude as additional parameters for clustering, since additional parameters have the potential to improve the ability of clustering algorithms to detect open clusters. The most ideal scenario possible for aiding in their detection would be to have precise information on all characteristics of stars which are similar within open clusters. In the near future, the increase in availability of radial velocity data will be certain to improve clustering to find open clusters. What could also be possible, is to use other parameters such as the chemical compositions of stars and extinction levels of the stars spectra within the these algorithms. *GAIA* has already began to release this kind of data in very small quantities, and although collecting this information will take a lot of time, it will be very interesting to see future work make use of this for open cluster discoveries.

# Bibliography


Alegre Aldeano C., 2022

Ankerst M., Breunig M. M., Kriegel H.-P., Sander J., 1999, ACM Sigmod record, 28, 49

Apellániz J. M., González M. P., Barbá R., 2021, Astronomy & Astrophysics, 649, A13

Cánovas H., et al., 2019, Astronomy & Astrophysics, 626, A80

Cantat-Gaudin T., et al., 2018, Astronomy & Astrophysics, 618, A93

Castro-Ginard A., Jordi C., Luri X., Julbe F., Morvan M., Balaguer-Núñez L., Cantat-Gaudin T., 2018, Astronomy & Astrophysics, 618, A59

Castro-Ginard A., et al., 2020, Astronomy & Astrophysics, 635, A45

Castro-Ginard A., et al., 2022, Astronomy & Astrophysics, 661, A118

Dias W. S., Monteiro H., Moitinho A., Lépine J. R., Carraro G., Paunzen E., Alessi B., Villela L., 2021, Monthly Notices of the Royal Astronomical Society, 504, 356

Ester M., Kriegel H.-P., Sander J., Xu X., et al., 1996, in kdd. pp 226–231

Fabricius C., et al., 2021, Astronomy & Astrophysics, 649, A5

Gao X.-H., 2017, Research in Astronomy and Astrophysics, 17, 058

He Z., et al., 2022, VizieR Online Data Catalog, pp J–ApJS

Hog E., et al., 2000, Technical report, The Tycho-2 catalogue of the 2.5 million brightest stars. Naval Observatory Washington DC

Hunt E. L., Reffert S., 2021, Astronomy & Astrophysics, 646, A104

Jadhav V. V., Pennock C. M., Subramaniam A., Sagar R., Nayak P. K., 2021, Monthly Notices of the Royal Astronomical Society, 503, 236

Khan K., Rehman S. U., Aziz K., Fong S., Sarasvady S., 2014, pp 232–238

Kharchenko N., Piskunov A., Schilbach E., Röser S., Scholz R.-D., 2013, Astronomy & Astrophysics, 558, A53

Lindegren L., et al., 1997, Astronomy and Astrophysics-A&A, 323, 49

Lindegren L., et al., 2021, Astronomy & Astrophysics, 649, A2

McInnes L., Healy J., Astels S., 2017, J. Open Source Softw., 2, 205

McKnight P. E., Najab J., 2010, The Corsini encyclopedia of psychology, pp 1–1

Olsson P.-O., Pippig K., Harrie L., Stigmar H., 2011, Mapping and Image Science, pp 22–29

Ou X., Necib L., Frebel A., 2022, arXiv preprint arXiv:2208.01056

Pera M. S., Perren G. I., Moitinho A., Navone H. D., Vazquez R. A., 2021, arXiv preprint arXiv:2101.01660

Prusti T., et al., 2016, Astronomy & astrophysics, 595, A1

Ye X., Zhao J., Oswalt T. D., Yang Y., Zhao G., 2022, arXiv preprint arXiv:2207.14229