university of groningen

faculty of science and engineering

Master's thesis

Finding structures in integral of motion space of the Milky Way halo and proving their statistical significance

March 11, 2021

Sarah Sofie Lövdal
s3017834

University of Groningen
Department of Computing Science

Supervisors:

Prof. Dr. Michael Biehl
Prof. Dr. Amina Helmi

Abstract

Purpose: The Milky Way halo has long dynamical time scales meaning that its components retain imprints of their history. Galaxies grow hierarchically by accreting smaller structures and stars in the acquired object will retain similar orbital parameters: the integrals of motion. Clustering is non-trivial in this space. This thesis develops a data-driven and statistically based method for finding clusters in integral of motion space, together with evaluating their significance. Furthermore, we extract subgroups in velocity space and derive a membership probability given the measurement uncertainties in the underlying data. ***Methods:*** Our clustering is based on an exhaustive use of the single linkage algorithm using four features: energy, angular momentum, angular momentum in z-direction and circularity. In order to evaluate each candidate cluster we generate an artificial halo by scrambling the velocity components of our data set. We use Poisson statistics with $\Delta N_{C_i} > 3\sigma_i$ to determine the significance in overdensity of each cluster compared to the artificial halo and expected density in the same region. Furthermore, it is common that an accreted structure is divided into different streams or clumps throughout the galaxy due to phase mixing and the nature of merger events. We therefore apply the HDBScan algorithm in velocity space to get an indication of substructure within a cluster. Finally, we determine the probability of a star belonging to a specific cluster given the measurement uncertainties in the underlying data. We linearly propagate the errors of the raw data into our clustering features and subsequently model each star and cluster as a probability density function in four dimensions. The membership probability of a star is indicated using the Mahalanobis distance between the probability density functions of the star and the cluster respectively, as a function of the cumulative chi-square distribution with four degrees of freedom. We apply our methods on the Gaia eDR3 RVS sample within 5 kpc. ***Results:*** The majority of stars in our data set, 55%, are assigned to some significant cluster. The method detects 419 clusters. The majority of clusters depict clear substructure in velocity space, with clusters being divisible into 1-4 components. The membership probability results in a catalogue with stars close to the mean of a cluster obtaining high confidence in membership, and stars further away from the mean, or with a large uncertainty, obtaining a lower probability. ***Conclusions:*** The thesis provides a data-driven and probabilistic way of extracting clusters in integral of motion space, together with analysing subgroups in velocity space and cluster membership probability given measurement errors in the underlying data. While the methods can be further polished, some improvements have been made compared to more heuristic methods of previous works. Further analysis is needed to determine the precise properties of the detected clusters and their correspondence to previously established structures.

Contents

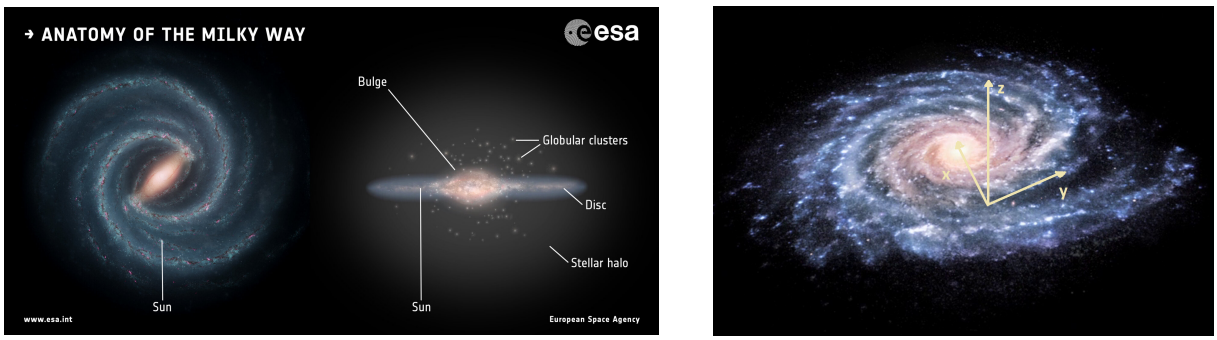
1 Introduction

According to the currently accepted Lambda cold dark matter cosmological model, galaxies grow hierarchically by merging with smaller structures [1]. In the Milky Way, footprints from such events can be recovered in the stellar halo. These footprints are called stellar streams and represent Galactic fossils which are still orbiting the Milky Way [2]. In addition to accreted dwarf galaxies, stellar streams can also originate from disrupted globular clusters, an originally tightly bound collection of stars that over time has dissolved due to gravitational impact. An overview of the Milky Way can be seen in Figure 1a, where the galactic halo is seen as the low density spherical "cloud" surrounding the dense bulge and disc.

Astronomers can identify streams by looking at the orbital properties of the stars, possibly also linked together with some of their chemical or spatial information. One of the most reliable ways to find stellar streams is to observe them as clusters in integral of motion space. This space involves specific orbital parameters which are considered to be conserved in a merger event. In an axisymmetric galaxy such as the Milky Way, the integrals of motion are the energy of the star, together with the angular momentum along the z-axis of the galaxy plane (see Figure 1b). Hence, establishing statistical significance in clusters in integral of motion space can help recover the formation history of the Milky Way and confirm galaxy formation theories in general.

In order to get reliable estimates of the integrals of motion of a star, very precise measurements of its position, velocity and distance from our solar system are needed. Obtaining such measurements becomes increasingly difficult the more distant and faint the objects are. Luckily, in the past few years the amount and quality of data mapping stars in the Milky Way has increased significantly. Especially the Gaia survey - covering more than a billion stars - has contributed to enhancing the knowledge about our galaxy, its structure and formation [3–5]. Still, only a minority of stars covered in the Gaia survey include measurements for all observables needed to accurately compute the required orbital parameters pertaining to the integral of motion space. In addition to this, much of the available data has large measurement uncertainties, making the reliability of the results more difficult to assess. Even though a significant amount of research has been done into identifying clusters in integral of motion space, defining clustering criteria as well as handling the uncertainty in the underlying data has proven to be difficult, and has often been done relying on human input and various heuristics.

To complicate things further, clustering in itself is an ill-defined problem. Falling in the category of unsupervised machine learning, there is no reference solution or labels to verify the accuracy of your clustering [6]. How do you define a cluster in the first place? And even if it looks like a dense collection of data points, how can you prove its statistical significance? For the application of finding clusters in integral of motion space, astronomers have so far relied heavily on manual selection and clustering evaluation. This actually works well in the most conspicuous cases, as the astronomers know exactly what kind of distributions to expect in certain regions. Problems with this arise when the dimensionality and amount of data increases, and cases become less and less distinguishable to the human eye.



(a) (b)

Figure 1: (a) Structural overview of the Milky Way. (b) The Milky Way overlaid with a Cartiesian coordinate system centered around the sun (adapted from ESA image gallery). Image credit: NASA/JPL-Caltech and ESA.

In computing science there is a whole field related to evaluating clustering results [7], and so called internal evaluation is often applied to assess the fit. This involves various techniques to measure how well separated the clusters are, as well their density - for example using the Dunn index [8] or Silhouette coefficient [9]. However, many clustering algorithms and evaluation methods assume that the whole data set should be clustered, or that every data point belongs to some cluster. This is not necessarily true for our application, as we expect to sift trough plenty of background noise (incidentally located stars) in our search for dense clusters. Therefore the method we develop will have to distinguish between noise and significant signal, and we will have to look further than many of the most common clustering algorithms and evaluation methods.

Furthermore, we have an issue with measurement uncertainties in the underlying data. Say that we are able to say that a group of data points display significant clustering given their measured values. How can we be sure that a data point is part of the cluster in case the true values of its energy or angular momentum differ from the measured ones? The importance of taking the measurement uncertainties into account becomes even more pertinent when trying to determine what is noise and what is signal in this data set.

The purpose of this thesis is to develop an efficient and precise method to identify clusters in integral of motion space, together with proving their statistical significance. The work consists of three parts. First, we present a data-driven and statistically based clustering method which provides an exhaustive detection of signal in the data set. Second, we extract substructure within an established cluster by looking at velocity space - this would reveal the different stellar streams originating from the same accreted object. Third, we present a method to assess cluster membership probability given measurement errors in the underlying data. We would like to answer the following research questions. How can we develop a clustering method for integral of motion space whose elements are mathematically based to the largest extent possible? How many clusters can we detect in the Milky Way halo and what is their statistical significance? What properties do they display in velocity space? How much do the uncertainties in the underlying data affect the reliability of clustering?

Section 2 describes the background related to topics treated. Section 3 describes the methods we have used in order to answer our research questions and Section 4 displays the results. Finally, in Section 5 we analyse the results and discuss possible future work.

2 Background

This section presents the background necessary to understand which quantities are needed for reliable clustering in the search for streams, together with their derivations from the available astrometric data. It covers stellar streams and their properties, the origin of our data set, and how to compute the integrals of motion. Previous research related to clustering in integral of motion space is covered in order to review existing approaches to the problem and the possible improvements which can be made.

2.1 Stellar streams

A stellar stream is a once independent globular cluster or dwarf galaxy, which has been "eaten up" and merged into a larger galaxy [10]. They often appear on the sky as elongated strings, due to the orbits of the accreted objects being impacted by tidal forces - hence the name. During a merger event, stars of the orbiting satellite object are gradually absorbed by the larger galaxy, in a process called tidal disruption. In the Milky Way the best place to observe streams is in the stellar halo (see Figure 1a). The halo has a sparse distribution of stars, implying minimal gravitational interaction with nearby objects. This contributes to preserving their dynamics and trajectories over time. The halo is mainly populated by old stars and the dynamical time scales are long - the objects here preserve their orbits for a long time, and hence we can observe traces of accretion events even after billions of years. The stars of a stream move along similar orbits, but do not necessarily move together as clumps in space, since their spatial distribution has been stretched out by tidal forces and phase mixing.

Streams will also display clustering in velocity space. This could be in the form of string-like structures or separate subgroups. The latter can occur when a stream wraps around its orbit, in which case two portions of the same stream could cross each other in space. Since we observe the stream from a single region of the galaxy, the solar neighbourhood, we do not have data from every location of the orbit of the stream. The data we have available would then show different subgroups in velocity space for the same stream.

Stellar streams are also expected to display some patterns in the metallicity, which in astronomical terms is the abundance of all elements heavier than helium. The iron abundance (Fe/H) is a typically dominating measurable. The metallicity of stars originating from a single globular cluster have a distinctly lower dispersion than an arbitrary sample from the galaxy, due to originating from the same birth place [11]. The metallicity of a dwarf galaxy would show a larger dispersion, while supposedly still being narrower than for the full Milky Way. Stars of a stream may also fall on roughly the same isochrone line when plotted in a color-magnitude diagram, in case the stars originate from the same generation of star formation. This diagram describes the life cycle of a group of stars born around the same time, based on their color (corresponding to temperature) and luminosity (brightness) [12].

It is theorised that the outer halo of the Milky Way might be entirely built up my mergers, whereby this area should be full of streams. Unfortunately this is very far from our solar system and it is difficult to obtain high quality data from this region. Furthermore, scientists predict the number of streams in the local stellar halo to be between 300 and 500, where each stream should contain no more than 5% of the stars [2, 13].

The phase-space in this context consists of six parameters that can uniquely determine the orbit of a star - the phase space coordinates. These are listed in Table 1. An integral of motion is a function of the phase-space coordinates that is constant along the orbit. As stars in a stellar stream move along similar orbits, this is the reason that we expect streams to appear as clusters in integral of motion space. The integral of motion space is described more closely in the next section.

2.2 Integral of motion space

For stars with orbits around an axisymmetric potential such as the Milky Way, the integrals of motion are the total energy E and the angular momentum along the z-axis of the galactic coordinate system, perpendicular to the galaxy plane, denoted L_z. These are orbital parameters which are independent of time if the system is static and axisymmetric. A third integral of motion may exist, but not for all orbits and galactic potentials, and it may not have an analytic form [14]. The two aforementioned values can characterise stellar streams, since they are considered to be conserved in a merger event, and the stars will preserve these similar characteristics in their orbits after the accretion. In addition to E and L_z, the total angular momentum L is often used as an additional feature, since it is considered quasi-conserved in a merger event. Similarly one can choose to use the perpendicular angular momentum $L_\perp$ (magnitude along the x- and y-axes) instead of L, as L_z is already a component in L. The energy and two features of angular momentum are referred to as the integral of motion space.

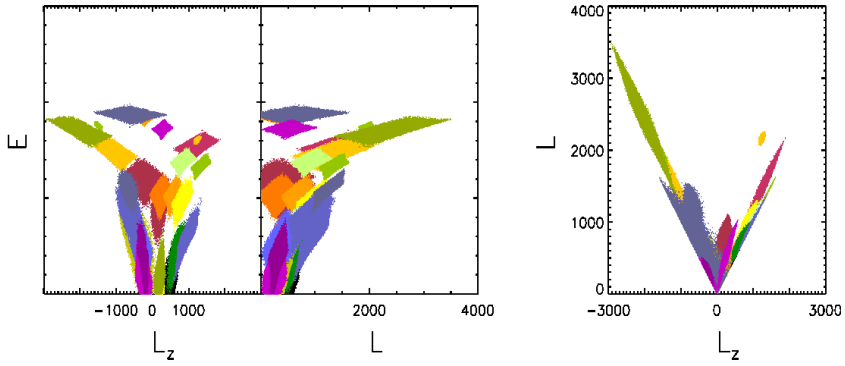
An example of clustering in this space can be seen in Figure 2, where Helmi et. al. performed a simulation of galaxy accretion events, followed by tracing of these events in integral of motion space 12 billion years later [2]. Figure 2a shows their initial distribution of halo stars in $En - L_z$ space and Figure 2b the same stars after simulating their dynamics forward in time. As can be seen, the values of the integrals of motion for each cluster are very similar to the initial distribution even after billions of years of orbiting the galaxy and adding noise representing observational errors.

Next, we describe how the integrals of motion are calculated based on various raw data products obtained from astrometric surveys.

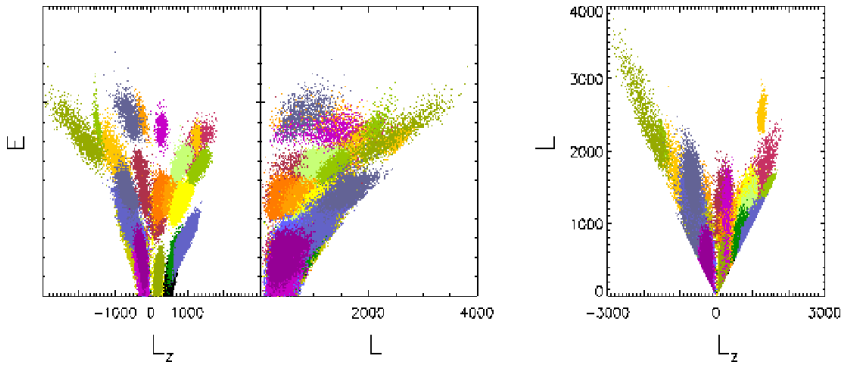
With precise instruments it is possible to measure how much a star moves over time with respect to more distant objects, even though this is impossible to see by the naked eye. This is called the proper motion μ and is often measured in two directions: right ascension μ_α and declination μ_δ. The position of a star on the sky at a certain moment in time can correspondingly be indicated as right ascension α and declination δ. These measurements are given in equatorial coordinates relative to the earth, but to study things in the Milky Way the observations are converted to the Galactic coordinate system (l, b). This is a spherical coordinate system where galactic longitude l corresponds to the angle between the sun, observed star and the galactic centre, and b indicates the angle between the disk ($b = 0$) and the star [15]. The conversion of equatorial to galactic coordinates can be done by multiplication by a rotation matrix $\mathbb{B}$. Given galactic coordinates the position of the star can be given in Cartesian coordinates, according to

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} d \cdot cos(b) \cdot cos(l) \\ d \cdot cos(b) \cdot sin(l) \\ d \cdot sin(b) \end{bmatrix} \tag{1}$$

The cartesian coordinates above are expressed with the sun as the origin, and this coordinate system can be seen in Figure 1b. The coordinates can also be expressed with the galaxy center as origin by translation of the



(a) Initial distribution



(b) Distribution after 12 Gyr, with added observational errors

Figure 2: Initial and final distribution of stars in the integral of motion space according to Helmi's merger simulation from 2000 [2]. This work simulated a number of merger events for a stellar halo and modelled the distribution of the stars in integral of motion space at the time at the merger (a), as well after 12 billion years (b).

heliocentric coordinates, depending on your needs. The distance (parsec) to a star can be calculated given the parallax π, according to

$$d = \frac{1}{\pi} \tag{2}$$

The radial velocity v_{rad} describes the component of the velocity vector of a star along the line of sight, and is measured using a star's Doppler spectrum (blueshift or redshift) [12]. The 3D space velocity vector of a star can be calculated given the radial velocity v_{rad}, proper motion μ, and distance d [16]. Here $k = 4.74$ is a unit conversion factor and $\mathbb{B}$ is a matrix defining the transformation to the Galactic coordinate system.

$$\vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \mathbb{B} \begin{bmatrix} v_{rad} \\ k \cdot \mu_\alpha \cdot d \\ k \cdot \mu_\delta \cdot d \end{bmatrix} \tag{3}$$

When calculating the integrals of motion we have also added corrections for the motion of the sun and around the galactic center to the velocity components, assuming $V_{LSR} = 232$ km/s (the velocity of the Local Standard of Rest) and a peculiar motion for the Sun of $(U_\odot, V_\odot, W_\odot) = (11.1, 12.24, 7.25)$ km/s [17]. We assume that the sun is located at a distance of 8.2 kpc from the Galactic Centre.

Yet another coordinate system used is the galactic cylindrical coordinate system: (R, ϕ, z). R is the distance from the galactic centre to the star, ϕ is the angle between the sun, galactic centre and the object, and z is height compared to the galactic plane, equivalent to the Cartesian z-coordinate. Given cartesian galactocentric coordinates, $R = \sqrt{x^2 + y^2}$ and $tan(\phi) = y/x$. Based on this, the energy can be calculated as

$$E = K + U = \frac{1}{2}v^2 + \Phi(R, z) \tag{4}$$

Here K is kinetic energy and U the potential energy. The potential energy represents the energy needed to escape the system that the star is orbiting, and depends on the mass distribution of the structure that is being orbited. This is expressed as a galactic potential Φ. Since the structure of the Milky way cannot be directly observed we can only make a guess for this expression, however, there is much support for that we can make reasonable assumptions. We will model the galactic potential of the Milky Way similarly to some previous work [18], with a Navarro–Frenk–White profiled halo, Miyamoto-Nagai disk and a Hernquist bulge. These components can be expressed as

$$\begin{aligned}
\Phi(R, z) &= \Phi_{halo} + \Phi_{disk} + \Phi_{bulge} \\
\Phi_{halo}(R, z) &= \frac{GM_{halo} r_s ln(1 + \sqrt{(\sqrt{R^2 + z^2})/r_s})}{(\sqrt{R^2 + z^2})(r_{vir}(ln(1 + c_h) - \frac{c_h}{1+c_h}))} \\
\Phi_{disk}(R, z) &= -\frac{GM_{disk}}{\sqrt{R^2 + (a_d + \sqrt{z^2 + b_d^2})^2}} \\
\Phi_{bulge}(R, z) &= -\frac{GM_{bulge}}{\sqrt{R^2 + z^2} + c_b}
\end{aligned} \tag{5}$$

where $a_d = 6.5$ kpc, $b_d = 0.26$ kpc, $c_b = 0.7$ kpc, $r_s = 21.5$ kpc, $r_{vir} = 258$ kpc, c_h=12, $G = 4.301 \times 10^{-6}$, $M_{halo} = 10^{12} M_\odot$, $M_{disk} = 9.3 \times 10^{10} M_\odot$ and $M_{bulge} = 3.0 \times 10^{10} M_\odot$ are constant parameters of the model.

The angular momentum can be derived given the velocity $\vec{v}$ and Cartesian coordinates according to

$$\begin{aligned}
L_x &= yv_z - zv_y \\
L_y &= zv_x - xv_z \\
L_z &= xv_y - yv_x \\
L_\perp &= \sqrt{L_x^2 + L_y^2} \\
L &= \sqrt{L_x^2 + L_y^2 + L_z^2}
\end{aligned} \tag{6}$$

These are the equations that have been used to compute the required integrals of motion based on the raw data measurements. The form of the equations is also of importance when modelling the errors in our clustering features, as we will do in Section 3.3.

Table 1: 6D phase space data fields in Gaia eDR3 [5, 19]. These components are the basis for computing the integrals of motion.

Data product	Symbol	Unit
Right ascension	α	milliarcsec
Declination	δ	milliarcsec
Parallax	π	milliarcsec
Proper motion (right ascension)	μ_α	milliarcsec/year
Proper motion (declination)	μ_δ	milliarcsec/year
Radial velocity	v_{rad}	km/s

2.3 Gaia eDR3

The Gaia mission is a European effort to map the stars in the local Milky Way [3]. The Gaia satellite was launched in 2013 and released its first data in 2016. This work uses data from the Gaia early Data Release 3 (Gaia eDR3), which for this work conveniently was released on the third of December 2020 [5]. The Gaia eDR3 catalogue contains five parameter astrometry (position, parallax and proper motion) for roughly 1.5 billion stars. A subset of 7.2 million stars contain measurements for radial (line of sight) velocity. While the catalogue also covers various other data products, this work is concerned with the above mentioned six-dimensional phase space coordinates. These are listed in Table 1. In the data catalogue, each measurement is associated with an individual measurement error. Out of the mentioned radial velocity sample (RVS), we make a selection in order to extract the halo stars - the area where we are interested in finding streams - together with some other quality cuts on the data. These are further described in Section 3.1.1.

2.4 Related work

Many works have already used clustering in integral of motion space successfully for identifying stellar streams in the Milky Way halo. In recent years, works using more advanced machine learning to find clusters have also been published.

As mentioned in Section 2.1, Helmi and de Zeeuw started already in the year 2000 to prepare methods for when the first data from Gaia would be released 16 years later [2] - specifically, performing a simulation investigating to what extent merger events can be recovered billions of years after they took place. They let 33 satellite galaxies with known properties be disrupted by a galactic potential similar to the Milky Way, in order to build up the full galactic stellar halo. They then integrated the orbits of the satellites 12 billion years forward in time and subsequently created an artificial Gaia catalogue by observing the properties of the stars involved in the simulation. Observational errors were also added to the measurements based on expected precision levels (see Figure 2). They then applied the Friends of Friends algorithm in integral of motion space (E, L, L_z) in order to extract clusters. Given the measurements of the stars observed after 12 billion years, they were able to recover around 50% of the streams. Some problems that were faced is related to the clustering features: for an axisymmetric potential only E and L_z are fully conserved, while the total angular momentum L is conserved only partially. Assuming a (static) galactic potential also influences the calculation of E and may not accurately reflect the Milky Way. However, even when they performed the clustering using slightly different potentials, they were able to recover a similar number of streams.

Once the first data release of Gaia had been made, Helmi et al. [20] isolated a sample of halo stars by combining data from Gaia DR1 and RAVE. They identified significant substructure in velocity space using a velocity correlation function - measuring the proportion of pairs of stars having a significantly small difference in their velocity components compared to random sets. Their random sets were realized by shuffling the velocity components of their data set, and this correlation in velocity vectors is predicted to be an additional byproduct from merger events. They proceeded to identify overdensities in two versions of integral of motion space (E, L_z) and $(E, L_z, L_\perp)$ using a non-parametric density estimator with a Epanechnikov kernel, and used a maximum filter to find local maxima in densities. These overdensities were then compared to overdensities found in their randomized data set. The Watershed algorithm was used to determine the extent of members of a cluster. Altogether, this resulted in finding nine new substructures, named VelHel-1 to -9.

Koppelman et al. [21] isolated the retrograde halo (stars orbiting in the opposite direction compared to

the predominant) in order to map its substructures and their properties. Here, a combined dataset from the surveys Gaia, APOGEE, DR14, LAMOST and RAVE was used, containing 6D phase space data and chemical information. For mapping out substructures in the retrograde halo, they used HDBSCAN to look for fine-grained clusters using four parameters: E, L_z and eccentricity, which is a measure of how much the orbit deviates from a circle, together with metallicity [Fe/H]. This resulted in identifying an additional structure, Thamnos, together with mapping the properties of previously identified structures in the retrograde halo.

Borsato et al. [22] used DBSCAN to identify clusters in the Gaia DR2 radial velocity sample. Here, they deal with the measurement errors in the data by modelling each data point as a multivariate Gaussian distribution with the standard deviation of each feature reflecting the measurement error in the raw data products. They then run 200 Monte Carlo simulations on this data by sampling from the joint probability distribution representing the measurements of each star, and subsequently calculate the corresponding integrals of motion. After applying DBSCAN on each of the 200 subsets, clusters that have less than four stars in common with any other cluster are removed and considered noise. The remaining stars in the subsets were then clustered with optimised parameters. Statistical significance of a cluster was established by comparing the density of the observed clusters with the expected density in a theoretical model of the Milky Way. Finally, they use a colour-magnitude diagram to verify that the observed clusters have similar physical properties.

An overview of the most important previously found structures in the Milky Way halo can be seen in Figure 3 [23] (note that the L_z axis in this diagram is flipped compared to the standard used in this thesis). This work by Naidu et al. made a detailed analysis of the substructure in the stellar halo using metallicities and integrals of motion and was able to assign more than 95% of their sample to some substructure.

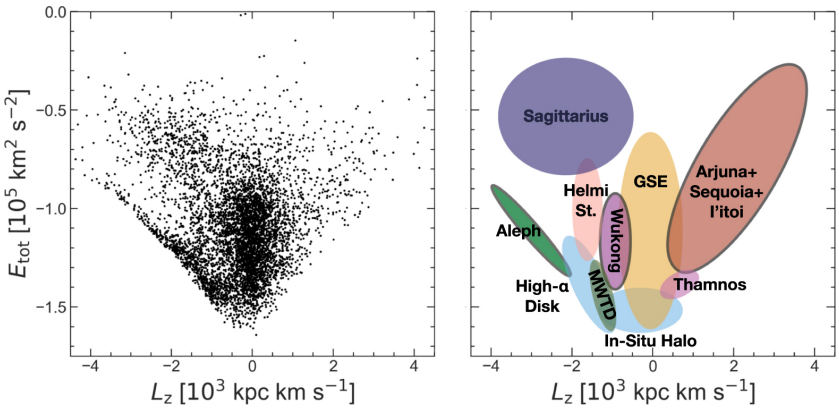


Figure 3: Map of the most important previously found structures in the Milky Way halo, according to [23].

To summarize the existing methods, astronomers have used either manual selection or various clustering algorithms, together with domain knowledge, in order to extract clusters in integral of motion space. While a great deal of structure has been detected this way, one of the main problems with manual selection is the difficulty to precisely motivate the observed structures. When applying a clustering algorithm, the choice of parameters is a problem. As the parameter choice will inherently dictate the outcome, it becomes impossible to prove that the outcome of the algorithm is an ideal clustering, as we cannot use physics to argue which parameters would be ideal for a data set. The result of this is having to heuristically select a strict set of parameters for the clustering, and there is a risk of missing less obvious but still significant structures. Furthermore, the space we are operating in has a non-uniform density, with more stars at lower energies and angular momentum (see Figure 2a). This further complicates parameter choices. Assessing the statistical significance becomes especially difficult in the case of manual selection, as the human expertise which it is based on is not necessarily quantifiable. Furthermore, many clustering algorithms work as a black box giving little insight and control of how the outcome was obtained. This makes the results difficult to interpret and clusters whose origin cannot be precisely explained invite a certain amount of scepticism. Lastly, the measurement errors of the data are often ignored, which may have unsettling effects, as the errors may be even more emphasised in a derived quantity. Furthermore, it may be the case that the uncertainties are so large that it affects the validity of the cluster. One way to handle the uncertainties is to use repeated stochastic experiments in order to control the robustness of a clustering, as done in [22]. While this is promising in principle, it is computationally expensive, which becomes especially troublesome when the amount of data increases. Therefore, we will aim for a more analytic way to model the uncertainties.

3 Methods

One of the primary goals of this work has been developing a data-driven algorithm, where the extracted structures are statistically based and self-explanatory in nature. We also want a method that detects more than the most obvious clusters, which ideally is able to scan through data set without missing any significant structure. In this section we explain the methods we have used to achieve this. In Section 3.1 we explain the quality cuts we have made on the data set, feature selection and data scaling before proceeding to explain the considerations of the clustering algorithm. We also explain the steps taken in order to prove the statistical significance of the clusters we find. This is the first objective of the thesis. The second objective is extracting substructure in velocity space for the established clusters, and the method for this is described in Section 3.2. Lastly, in Section 3.3 we derive a membership probability for a star belonging to a cluster by modelling the stars and clusters as probability density functions given their propagated measurement uncertainties.

We will use the following basic notation: x_i for a data point in our four-dimensional clustering space. X_i for a data point modelled as a probability density function given its observed values and propagated uncertainties, corresponding to the mean $\mu_i = x_i$ and covariance matrix Σ_i. N is the number of stars in the data set after applying quality cuts, and we will call this selection the halo set. C_i denotes a candidate cluster, which is a connected component in the halo set for which we want to evaluate the statistical significance. We denote the number of members of a candidate cluster N_{C_i}.

3.1 Clustering in Integral of Motion space

This section describes the methods for the principal part of the thesis, which is developing a clustering algorithm to detect significant structures in integral of motion space. Section 3.1.1 describes the quality cuts made on the Gaia RVS sample together with some basic descriptions of its characteristics. Section 3.1.2 describes how we make the feature selection and data scaling. Section 3.1.3 describes our choice of clustering algorithm and how we use it to find all candidate clusters, corresponding to connected components in the data set. Finally, in Section 3.1.4 we describe how we evaluate the statistical significance of all the candidate clusters we find.

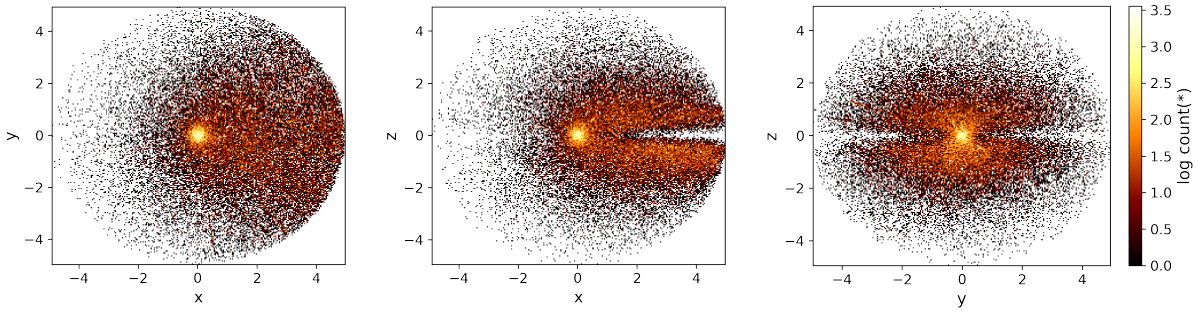
3.1.1 The data set

The data set we use in this work is the Gaia eDR3 radial velocity sample. Before we apply our clustering algorithm, we need to ensure that the data we enter is of the most premium quality possible. We place the following quality cuts on data fields in the Gaia catalogue: `parallax_over_error`> 5, `ruwe`< 1.4 and $d < 5$ kpc, where the distance is calculated according to Equation 2 after correcting the parallaxes for a zero-point offset of 0.017 mas. These cuts pertain to the reliability of measurements in the Gaia catalogue and have been done according to recommendations in previous work, e.g. Koppelman et al. [18]. We note that we have 76229 sources with $|V - V_{LSR}| > 180$ km/s, and we will use this selection as later described in Section 3.1.4.

We extract halo stars by demanding $|V - V_{LSR}| > 210$ km/s, a common condition which selects stars in the halo based on their large velocity magnitude compared to the local standard of rest. This cut is not too conservative, and will allow for some contamination from the thick disk [17, 18]. Additionally, the energy as calculated by Equation 5 and the galactic potential we define may result in an energy larger than zero (meaning that the star would have exceeded the escape velocity). We remove the small number of stars this regards from the data set. The resulting data set contains $N = 48880$ sources and we will refer to this selection as the halo set. The halo set is visualized by its Cartesian coordinates in Figure 4. Most sources are in the near vicinity of the sun (the origin), with a higher density of observations towards the positive x-axis (the galactic centre). Part of this is interpreted as contamination from the disc. We see a gap in observations around $z = 0$. This is due to the Gaia catalogue containing less stars with radial velocities in the Galactic bulge and disk, in combination with our selection procedure for the halo [24].

3.1.2 Feature selection and scaling

The first consideration at hand is what features to use in the clustering step. Energy E and angular momentum in z-direction L_z are given, as these are true integrals of motion and better preserved in a merger event. The total angular momentum L (and similarly the perpendicular angular momentum $L_\perp$) are quasi-preserved and thus also carry information, even though this signal will be more noisy than the one contained in E and L_z. In addition, it can be seen that the clusters in general are not expected to have a spherical shape, but rather follow a slightly outwards bent shape in $E - L_z$ space - look for example at Figure 2. This becomes a problem for many clustering algorithms, as the most common objective function is to maximize the density in a spherical region. Looking more closely at the way the structures bend in $E - L_z$ space, this domain is in fact bounded



(a) (b) (c)

Figure 4: The halo set as visualized in Cartesian coordinates with origin centered around the sun.

by physical laws and for each value for E there is a maximum possible L_z. This funnel-like shape is due to energy having a quadratic dependency on the total velocity $|v|$ while L_z is linearly depending on v_x and v_y. Astronomers call this slightly bent shape and orbital parameter the circularity of a star, and it can be calculated as a function of E and L_z. We will denote the circularity as η. In addition to the total angular momentum L, we choose η to be a fourth input feature, as this conveys additional important information about the shape of the clusters, and will effectively make a distance-based clustering algorithm prioritize a tendency towards the slightly bent curves of constant circularity. The circularity for each data point has been calculated according to

$$\eta = L_z / L_z^{max}(E) \tag{7}$$

where $L_z^{max}(E)$ is the largest possible L_z for a given energy (corresponding to a circular orbit) [25].

Each feature is linearly scaled to the range of [-1, 1] (except for the circularity, which is already in this range). We use a fixed reference range for this min-max-scaling in order to fix the relationship between the unscaled and scaled data coordinate system, which is necessary for being able to compare towards random reference data sets as we will do in Section 3.1.4. Our equal range in scaling also implies that each of the four features will be considered equally important in a distance-based clustering algorithm. In order to increase the weight of a particular feature, it can be scaled to a larger range than the other features. In this case, L_z and E could be seen as being given slightly more weight than L, as the circularity is directly derived from the two former, and also positively correlated with L_z. In addition, as the total angular momentum depends partially on the L_z-component, these two features are correlated. This indirect emphasis on L_z also is supported by the knowledge that the clusters are expected to have a larger spread in E and L than in L_z [2]. The halo set visualized in all combinations of clustering features can be seen in Figure 5.

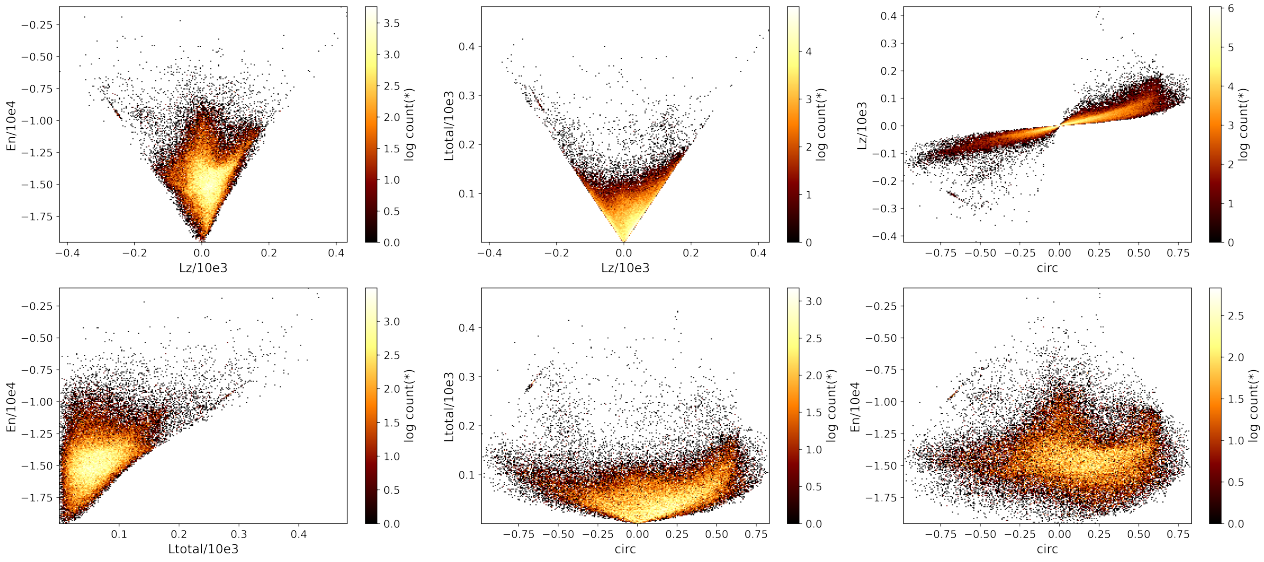


Figure 5: The Gaia eDR3 RVS halo sample after applied quality cuts, visualized for our four clustering features.

3.1.3 Selection of clustering algorithm

Next, we must choose a clustering algorithm that best fits our purposes. While there is a range of options to choose from, what we want is maximum control over the process combined with an exhaustive vacuum cleaning of information in the data set. This is especially important as we are dealing with unsupervised machine learning, so there is simply no ground truth to verify our findings with, apart from possibly results from simulations.

Our clustering algorithm will have to deal with a potentially large amount of noise in search for significant clusters, which immediately rules out traditional clustering algorithms such as K-means or Expectation Maximization with Gaussian Mixture Models [7], as they assume that all data points should be assigned to some cluster. Options that also can handle noise would be the Friends-of-Friends algorithm, DBSCAN, and HDBSCAN [26–28]. The main problem with the former two is that they require specifying some static parameters, for example the smallest distance between data points of the same cluster. As our halo set consists of non-uniform density regions, using static parameters for this will only work well on some parts of the data space. HDBSCAN is able to extract variable density clusters, but in addition to having a bit of a black box tendency in the application it also requires specifying some parameters. As we want to make as little assumptions as possible about the properties of the clusters, as well as desiring full control over the clustering process, we look for a simpler option.

Hierarchical clustering is a group of algorithms that offers many of the requirements that we are demanding [7]. More specifically, agglomerative clustering is a bottom up approach which does not require any parameters. For each iteration of the algorithm, it merges the two clusters with the smallest distance between each other, with each data point initially being considered a singleton cluster. The method aligns well with our desire to approach our clustering from a perspective of probability theory, since under the hypothesis that the most likely clusters are the groups of data points with the smallest distance between each other, the algorithm provides a hierarchy of the most likely clusters in an exhaustive manner.

Hierarchical clustering can be based on various ways to measure the distance between two clusters. Average linkage measures the average distance between the data points of the clusters, and Ward linkage measures the distance such that the within-cluster variance is minimized. However, both of these assume isotropic cluster shapes, and in our case we would have to apply some advanced transformations to the data space before we could expect spherical clusters. The single linkage method, however, considers the distance between two clusters to be the shortest distance between any two data points belonging to the first and second cluster respectively. This way it allows for any cluster shape and is also computationally more efficient than other linkage methods, as it is only necessary to compute the distance matrix of the data points once, giving the algorithm a computational and memory complexity in $O(N^2)$. In this work we use the standard Euclidean distance metric to compute the distances between the data points.

The single linkage algorithm is also closely related to graph theory, as it is equivalent to finding the minimum spanning tree in a graph [29]. The minimum spanning tree consists of the set of edges between nodes in a graph for which the total weight of the edges is minimized, and each node is included. In our case, the nodes are our data points and the edges the distance between them.

The single linkage algorithm outputs a so called linkage matrix, giving information about which data points or clusters have been merged at each step of the algorithm, and the merge cost or distance between the two clusters. These steps will indicate iteratively larger connected components [29]. This is a great thing, since it means that the algorithm will iteratively merge groups of data points into clusters that are most likely to belong to the same structure, and as the merge cost increases, it is less and less likely that the merging of two clusters is desirable. The algorithm performs $N-1$ steps - it will keep going until every data point has been assigned to a cluster, or each connected component has been connected into the minimum spanning tree. By this process we obtain $N-1$ potential clusters encoded in the linkage matrix returned by the algorithm, each given by the the connected components that were merged at the corresponding step of the algorithm. This is another great thing, as we can now reverse engineer the linkage matrix and examine *any* candidate cluster, or connected component, for a quality criterion.

The basic idea is, test all $N-1$ candidate clusters included in the linkage matrix and evaluate them by a cluster quality criterion. The clusters that examine statistical significance according to the selected criterion are accepted. This way or method will also be able to handle noise, as the data points that do not belong to any cluster displaying statistical significance will be discarded. The criterion for statistical significance is described in the next subsection.

3.1.4 Evaluating the statistical significance

The single linkage algorithm provides an efficient way to examine every possible cluster of the data set in an efficient manner, since the hierarchical way of building clusters bottom up provides a list of all connected components in the data set being connected by edges shorter than or equal to the nth smallest edge length at step n. In this way, it maximizes the probability of obtaining the most closely connected components.

In order to examine the quality of each candidate cluster included in the linkage matrix, we need a cluster evaluation metric. As mentioned, a challenge is that our 4D clustering space has a higher density of stars with low energy and angular momentum than regions with higher energy - see the funnel-like shape in Figure 2. This makes it less straightforward to evaluate and compare clusters in different regions with a standard cluster evaluation metric such as the Silhouette Coefficient, as clusters in different regions will have different characteristics. Hence, we would like a method which takes the expected density of a region into account, combined with a metric which evaluates the quality of the clusters. In order to assess the expected density of a region, we need a reference frame for what a galaxy halo without structure would look like. In order to do this, we use the given data set but calculate the integrals of motion using random permutations of the v_y and v_z components. This way we obtain an artificial data set which has similar properties to the observed data, but where the correlations in the velocity components and hence structure in the integrals of motion space is broken up. The same method to generate a random reference data set was used in [20]. This method provides one of the most realistic non-structured distributions of stars in a galaxy halo that we can currently realistically achieve, since assuming a theoretical model of a galaxy and generating a halo based on this may lead to subtle issues related to the assumptions of the theoretical model. To generate each artificial data set we use a sample which is slightly larger than the selection we have used for our halo set, in order to obtain enough halo-qualifying stars in each artificial set. We therefore scramble the velocity components of all stars with $|V - V_{LSR}| > 180$ km/s (so the halo, but also a little bit more - 76229 sources in total). We then recompute the integrals of motion of the resulting data points, before randomly selecting N stars out of the set of artificial stars ending up with $|V - V_{LSR}| > 210$ km/s. We generate $N_{art} = 100$ artificial data sets as reference. An example of these artificial data sets used is displayed in Figure 6, where an arbitrarily chosen realization is visualized. Comparing to Figure 5 we see that the two data sets are very similar in their characteristics, but that the by eye visible substructure in the halo set has been diluted.

Next, we examine each candidate cluster which is included in the linkage matrix resulting from applying the single linkage algorithm to our halo set. We want to statistically either prove or reject that the candidate cluster that we are examining is significant, or with other words, not just a random collection of stars close to each other. We can assess this by first computing the expected density of stars in a region, and compare the difference between the observed and the expected count, while also taking the statistical error of these quantities into account.

We determine the expected number of stars in a region by defining an elliptical boundary around each

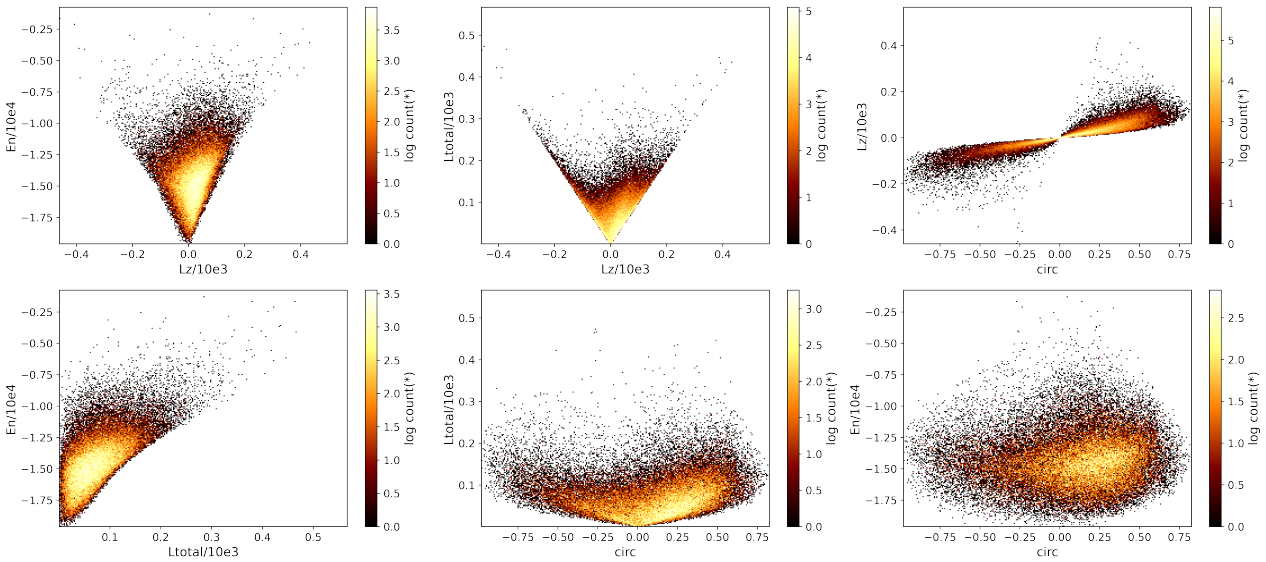


Figure 6: Artificial dataset number two out of $N_{art} = 100$, visualized in all subspaces of our clustering features.

candidate cluster C_i. This is achieved by applying principal component analysis (PCA) to the set of data points $\{x_{C_i}\}$ belonging to C_i. PCA approximates the shape of the cluster by computing the set of eigenvectors $\{e\} = [\vec{e_1}\ \vec{e_2}\ \vec{e_3}\ \vec{e_4}]^T$ and corresponding eigenvalues $\lambda = [\lambda_1, \lambda_2, \lambda_3, \lambda_4]^T$ of the covariance matrix of C_i [30]. These give the direction and magnitude of the principal components of the data distribution of the cluster, where first principal component corresponds to the direction of maximum variance in the distribution, together with its magnitude. Each principal component is orthogonal to each other. If we transform the data points of C_i to PCA-space P_i by mapping them according to

$$p_j = (x_{C_i} - \mu_{C_i})^T e \tag{8}$$

we obtain a mean-centred and rotated version of the data where the axes of the coordinate system correspond to each principal component. Here μ_{C_i} is the mean of all data points in C_i. We can relate this to the standard equation of an ellipse, since the data points in P_i are centered around the origin and the direction of maximum variance is aligned with each coordinate axis. The standard equation of an n-dimensional ellipsoid can be seen below, where a_i denotes the length of each axis [31].

$$\sum_{i=1}^{n} \frac{x_i^2}{a_i^2} = 1 \tag{9}$$

Since the variance along each principal component of our cluster is given by the eigenvalues, we can define the length of each axis of the ellipse based on this, selecting

$$a_i = 2\sqrt{\lambda_i} \tag{10}$$

corresponding to two standard deviations of the distribution along a_i, or covering approximately 95% of the data points along this axis. We can also consider a larger or smaller value for the length of the axis, but selecting two standard deviations provides a tightly fitting boundary around the data while neither being too strict nor including too much empty space. Now we can compute the expected count in the region of a candidate cluster C_i by analysing its PCA transformation, and mapping the data of the artificial data set to the PCA-space defined by C_i. Then, the expected count is the number of data points satisfying

$$\frac{p_1^2}{(2\sqrt{\lambda_1})^2} + \frac{p_2^2}{(2\sqrt{\lambda_2})^2} + \frac{p_3^2}{(2\sqrt{\lambda_3})^2} + \frac{p_4^2}{(2\sqrt{\lambda_4})^2} \leq 1 \tag{11}$$

where p denotes a data point from the artificial set having been mapped to PCA-space according to Equation 8. We do not have to apply the PCA-transformation to all stars in the artificial data set for each C_i, it is enough to inspect the near vicinity of each cluster, so we reduce the amount of mappings necessary by only inspecting a rectangular neighbourhood around C_i.

We can easily compute the standard deviation $\sigma_{C_i}^{art}$ of the counts that we find in each artificial data set in the region of a candidate cluster C_i. As we have generated 100 artificial halos, $\sigma_{C_i}^{art}$ is the standard deviation of the of the counts that we find in a region for each of the artificial halos, and should not be confused with any standard deviation obtained from our (real) halo set. The former is equivalent to the statistical error on the region count of the artificial data sets, $e_i^{art} = \sigma_{C_i}^{art}$. If we treat the observed data as having Poissonian properties, the statistical error on the observed cluster count becomes $e_i^{real} = \sqrt{N_{C_i}}$, as the variance is equal to the mean in a Poisson distribution [32]. The statistical significance of a cluster can then be obtained by the difference between the observed and expected count, divided over the sum of the squared errors. If we choose the significance level to be 3σ, this becomes

$$N_{C_i} - N_{C_i}^{art} > 3\sigma_i \tag{12}$$

$$\sigma_i = \sqrt{(e_i^{real})^2 + (e_i^{art})^2} \tag{13}$$

or similarly,

$$\frac{N_{C_i} - N_{C_i}^{art}}{\sqrt{N_{C_i} + (\sigma_{C_i}^{art})^2}} > 3 \tag{14}$$

This can be interpreted as how many standard deviations the real and the artificial counts differ from each other. It can be noted that $N_{C_i}^{art}$ is the average count in the artificial data sets.

Hence, all candidate clusters that satisfy Equation 12 are accepted as significant structures. As the linkage matrix contains hierarchically overlapping components, we need to make a decision on how to treat overlapping significant structures. The linkage matrix is likely to contain signal starting from the core of a significant cluster, and each after following merge of the algorithm. We therefore select the largest cluster which is significant when extracting cluster labels for the data set. In the worst case scenario we obtain some noise in the clusters by using this method, as a dense core together with a less dense surrounding might still result in a statistically significant overdensity compared to a semi-uniform background. However, we can detect and discard this potential noise by investigating the properties of each cluster further - this is described in the next subsection. Another way to approach the overlapping subclusters problem would be to stop merging structures when Equation 14 reaches its maximum value, but there are issues with this method, as there is no guarantee that this maximum will occur at the "real" cluster boundary.

Furthermore, to save computation time, we only investigate candidate clusters which contain less than 20% of all data points. This is motivated by that each stream in the local stellar halo is predicted to contain less than 5% of the stars [13], and considering this prediction investigating clusters smaller than 20% of the full data set is on the safe side. We also demand at least five members in a candidate cluster to investigate it further, but due to our criterion for statistical significance, the smallest number of members possible in an accepted cluster is 12. This is since we assign any expected normalized count of a region $N_{C_i}^{art}$ smaller than one the value one, and same goes for the standard deviation $\sigma_{C_i}^{art}$. Therefore $N_{C_i} = 12$ is the smallest integer which can satisfy Equation 14 with the smallest possible values for the expected count plugged in: $(N_{C_i} - 1)/\sqrt{N_{C_i} + 1^2} > 3$. The parameter choices covered in this paragraph can be changed according to the desired needs, for example the significance level in Equation 12.

The result of the methodology described above is a flat label assignment to the data set, where each unique label denotes the largest significant cluster that a data point belongs to, and data points not belonging to any significant structure are labelled as noise.

3.2 Extracting subgroups in velocity space

Velocity space, or more specifically, the v_ϕ, v_R and v_z components of a cluster can give us more clues about its validity and properties. In velocity space we also expect stars of the same cluster to be clumped together, but possibly broken up into many subgroups. These could be in the shape of elongated strings or dense clumps, as a result of tidal effect and phase mixing. Since the velocity components indicate the direction of movement of the stars, while not necessarily being in the same phase of the orbit, we are able to see more versatile patterns in velocity space compared to the constant integrals of motion. Hence, we want to extract substructure in each cluster that we have determined to be significant, which could represent groups originating from the same merger event but in different phases of their orbits.

We approach this by applying another round of clustering on each cluster which has been established as statistically significant. Thankfully each data set representing a cluster in velocity space is far smaller and less complex than our original halo set where we first needed to sort out what is significant and what is noise. An example of what we can expect in velocity space can be seen in Figure 7. Here wee see the same simulation by Helmi et al. as in Figure 2, with the distribution of the clusters depicted in velocity space. The original clusters can be seen in the top panel, with the clusters recovered in their simulation in the bottom. As the figure shows, some clusters consist of a single string, or bratwurst shaped structure, while others consist of distinct subgroups with for example opposite signs in v_R. Hence, our goal with this method is to extract labels for these distinct subgroups. Some considerations to take into account is that our clusters may contain some noise due to the way that we extract flat cluster labels, and that the clusters may have various shapes, sizes, and number of subgroups in velocity space. Much like in integral of motion space, we are not exactly sure of the shape and characteristics of the subgroups in this space. Hence, we need a clustering algorithm that is able to extract the most likely subgroups in the data while handling some possible noise.

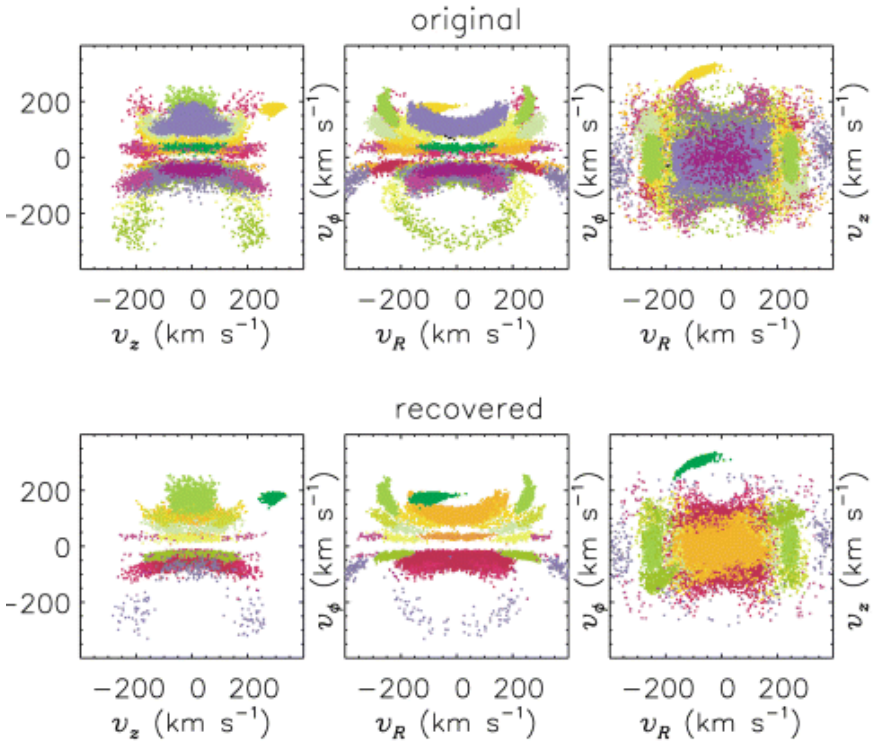


Figure 7: Halo star distribution in velocity space according to the simulation by Helmi et al. [2], corresponding to the clusters in integral of motion space in Figure 2.

We select the HDBSCAN algorithm to perform the clustering in velocity space, as it is both able to extract clusters of various shapes and sizes and sift out possible noise. HDBSCAN stands for Hierarchical Density-Based Spatial Clustering of Applications with Noise, and extends the DBSCAN algorithm by providing a hierarchical approach to extracting clusters [28]. HDBSCAN is closely related to the DBSCAN algorithm, but while the latter relies on fixed parameters for the neighbourhood radius and minimum number of neighbours to distinguish between clusters and noise, HDBSCAN is a form of robust single linkage algorithm which extracts stable clusters from the cluster hierarchy tree with an Excess of Mass algorithm. The reason we find it suitable in velocity space

Table 2: Parameters used for applying HDBSCAN in velocity space.

Parameter	Value	Function
min_cluster_size	$\min(10, \sqrt{N_C})$	Smallest number of members in a cluster
min_samples	$\min(3, \sqrt[4]{N_C})$	Smallest neighbourhood for a core point
allow_single_cluster	True	Singleton cluster allowed as output

but not the best choice in integral of motion space is mainly related to its black box character, decreasing the explainability of the extracted clusters, and having to specify parameters. However, in velocity space we have already established the statistical significance of each cluster and mainly want an indication of possible subgroups.

HDBSCAN works as follows. First, it defines the distance between two data points a and b as their mutual reachability distance. This is defined as the maximum of the Euclidean distance between the datapoints, the distance to the kth neighbour of a, and the distance to the kth neighbour of b. This way, outliers are penalized since their mutual reachability distance will be high. A minimum spanning tree (corresponding to applying single linkage) can then be formed based on the distance matrix defined by the mutual reachability. It then proceeds to analyse the resulting clustering hierarchy tree. The parameter `min_cluster_size` denotes the smallest group of data points that the algorithm will accept as a cluster in this context. Furthermore, a data point is a core point if it has at least `min_samples` in its neighbourhood given some specific distance threshold, and is otherwise either a border point belonging to a cluster, or noise. Increasing this parameter will result in the algorithm being more likely to classify outliers as noise. When analysing the cluster hierarchy, the algorithm distinguishes between boundary points falling out of a cluster and a parent cluster splitting into two, based on the `min_cluster_size`. The stability of each candidate cluster is evaluated over varying values for distance threshold and integrating the result. The clusters that stay the most stable when varying the distance threshold are chosen as the best clusters.

We apply HBDSCAN directly to the v_R, v_ϕ and v_z components of each cluster, without scaling, as the range for these values are already of the same magnitude. The parameter choices we make are listed in Table 2. We choose 10 as smallest cluster size and 3 as value for `min_samples`, as the former is a sufficient number of stars to be an interesting subgroup for astronomers, and we do not want to be too strict on noise classification with the latter. We also allow for a single cluster to be returned as output in addition to noise, as this is not allowed by default in HDBSCAN. Since we have some clusters that are as small as 12 members, while still possibly containing distinct components, we allow `min_cluster_size` to be $\sqrt{N_C}$ and `min_samples` $\sqrt{\sqrt{N_C}}$ in case we have a small cluster. The value for `min_cluster_size` in this case reflects the Poissonian error on N_C, and the value for `min_samples` the Poissonian error on the `min_cluster_size`.

The output of this is an additional column in our halo catalogue, indicating subgroup membership. The algorithm also labels data points that do not qualify as either a core or border point, being particularly far from the established clusters, as possible noise.

3.3 Assigning cluster membership probability

Once we have established significant clusters, we would like to obtain an estimate of the probability that a star belongs to a cluster given the, often significant, measurement errors in the raw data. Should we not already have taken these uncertainties into account earlier, when applying our clustering algorithm to our data sample? Not necessarily. The reason for this is that each measurement uncertainty can be seen as a statistical dispersion expressed in terms of a standard deviation around the mean [19]. Therefore, maximum likelihood estimation is applicable, and the most likely values are in fact the observed values. Therefore it would be possible to take the measurement errors into account when doing the clustering step, but it would add greatly to the complexity, while not necessarily providing much more accurate results.

Since we can interpret each raw measurement as a having a Gaussian distribution, this means that each quantity derived based on the raw measurements and their uncertainties also will have some theoretical probability distribution. Describing the nature of the error in a derived quantity is called error propagation and we will use this to model each star as a probability density function in our clustering space. Based on this, we can also model a cluster as the probability density function of all contributing members. We can then assess the membership probability based on how well the probability density of a star fits to the cluster distribution. The details of how we implement this is described below.

3.3.1 Error propagation

In order to be able to indicate the uncertainty for each feature in integral of motion space, we need to propagate the errors in the raw data. Ideally this would be done by deriving exact analytical expressions for the distribution of errors in the derived space, but unfortunately this is only feasible for simple expressions [33]. For non-linear transformations, an often used method to propagate errors is using a first-order Taylor expansion about the point $x = \mu_x$. Then, the covariance matrix Σ_y of the transformed variable $y = f(x)$ can be obtained based on the covariance matrix Σ_x and its Jacobian J_x, according to [16, 34]

$$\Sigma_y = J_x \Sigma_x J_x^T \tag{15}$$

The Jacobian J_x is a matrix defining all first order partial derivatives of the covariance matrix. The above equation can be applied independently of how many input and output variables the transformation consists of.

We propagate the errors for our clustering features E, L, L_z and η. For each star this gives us the covariance matrix of its four features Σ_i, which we use together with the previously computed values for our data points x_i, the latter being equivalent to the mean μ_i around which the Taylor expansion was made.

Even though this is an approximation and it assumes that the error distribution on the dependent variable is Gaussian (which is not the case for non-linear transformations), it is a good approximation when the error in x is small [34]. It also enables us to model the stars as four-dimensional Gaussians instead of more complicated distributions, which becomes extremely useful when proceeding to analyse the membership probabilities.

3.3.2 Clusters as probability density functions

Based on the propagated uncertainties and their covariances, we can now model each star as a probability density function in the form of a multivariate Gaussian X, with mean μ and covariance matrix Σ [35].

$$X \sim \mathcal{N}(\mu, \Sigma) \tag{16}$$

$$\mu = \begin{bmatrix} E & L & L_z & \eta \end{bmatrix}^T \tag{17}$$

$$\Sigma = \begin{pmatrix} \sigma_E^2 & \sigma_{(E,L)} & \sigma_{(E,L_z)} & \sigma_{(E,\eta)} \\ \sigma_{(E,L)} & \sigma_L^2 & \sigma_{(L,L_z)} & \sigma_{(L,\eta)} \\ \sigma_{(E,L_z)} & \sigma_{(L,L_z)} & \sigma_{L_z}^2 & \sigma_{(L_z,\eta)} \\ \sigma_{(E,\eta)} & \sigma_{(L,\eta)} & \sigma_{(L_z,\eta)} & \sigma_\eta^2 \end{pmatrix} \tag{18}$$

The probability density of a multivariate Gaussian evaluated at point x is determined by

$$\mathcal{N}(x \mid \mu, \Sigma) = (2\pi)^{d/2} |\Sigma|^{-1/2} e^{(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))} \tag{19}$$

We can analytically describe each cluster as a Gaussian multivariate distribution based on its member stars and their uncertainties. While Expectaton Maximization for Gaussian Mixture Models (EM-GMM) is a well known method to estimate the parameters of a Gaussian distribution given a set of data points [7], it is also possible to perform this expectation maximisation for the case where each data point is associated with an uncertainty. This is called extreme deconvolution [36] and extends traditional GMM by modelling each observed data point w_i as a noisy projection of their true values v_i: $w_i = v_i$ + noise, where the noise is drawn from the known covariance matrix Σ_i associated with the uncertainty of w_i. The extreme deconvolution algorithm maximizes the objective function ϕ, being the log-likelihood of each data point w_i belonging to the model given the model parameters θ (in this case being the cluster mean μ_C and covariance matrix Σ_C). Fitting a single Gaussian component per cluster, the objective function becomes

$$\phi = \sum_i ln(p(w_i \mid \theta)) = \sum_i ln(\mathcal{N}(\ w_i \mid \mu_C,\ \Sigma_C)) \tag{20}$$

Evaluating the probability density of w_i against the cluster distribution yields [36]

$$p(w_i \mid \theta) = \mathcal{N}(w_i \mid \mu_C,\ \Sigma_C + \Sigma_{w_i}) \tag{21}$$

Applying an iterative expectation maximization algorithm using the objective function ϕ will converge to the Gaussian multivariate distribution $\mathcal{N}(\mu_C, \Sigma_C)$ of each cluster C given the uncertainties in the underlying data [36].

3.3.3 Evaluating membership probability

As we now have an expression for probability density function of each cluster, we can obtain estimates for how well each participating data point fits to this distribution by looking at its distance to the cluster mean in combination with the covariance of both distributions. More specifically, we obtain the likelihood of each X_i belonging to the cluster distribution $\mathcal{N}(\mu_C, \Sigma_C)$ by Equation 21, which can be expressed as

$$p(x_i \mid \theta) = \mathcal{N}(x_i \mid \mu_C,\ \Sigma_C + \Sigma_{x_i}) \tag{22}$$

for each observed data point x_i belonging to cluster C. This is equivalent to the probability density of $\mathcal{N}(\mu_C, \Sigma_C)$ evaluated at X_i. The likelihood represents the relative probability that the cluster distribution takes the value X_i, and integrating over the full distribution gives the value 1. While the relative likelihood provides a value which enables comparison of how well various data points fit the cluster distribution, it does not give an immediately interpretable indication of membership probability. The value of the likelihood at a certain Euclidean distance is also dependent on the shape of the distribution. A wide distribution gives a lower likelihood at a constant Euclidean distance than a highly concentrated one - this is evident for example when comparing a one-dimensional normal distribution with a large standard deviation to a normal distribution with a small standard deviation. Therefore we would like to relate the likelihood to a more independent measure of membership probability. Fortunately, given a specific dimensionality, a Gaussian probability density function will have constant likelihood at ellipsoids of constant Mahalanobis distance D [37, 38]. This quantity measures distance in terms of variance along the axis between the data point x and mean of the distribution μ, and is defined as

$$D = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)} \tag{23}$$

Where Σ^{-1} is the inverse of the covariance matrix describing the shape of the distribution. The Mahalanobis distance can be thought of as the distance to the mean, in terms of standard deviations, after the distribution has been normalized to spherical covariance. In 1D, the Mahalanobis distance is equivalent to the number of standard deviations a data point is from the mean. We can compute the Mahalanobis distance between the probability density X_i and the cluster distribution given the likelihood obtained from Equation 22, and noting that the likelihood of a specific n-dimensional multivariate normal distribution is uniquely determined by the Mahalanobis distance. Relating to Equation 19 and Equation 23, a Gaussian probability density function can be expressed in terms of Mahalanobis distance as

$$\mathcal{N}(x \mid \mu, \Sigma) = (2\pi)^{n/2} |\Sigma|^{-1/2} e^{(-\frac{1}{2} D^2)} \tag{24}$$

The squared Mahalanobis distance can then be solved out as

$$D^2 = -2(ln(\mathcal{N}(x \mid \mu, \Sigma)) + ln((2\pi)^{n/2}) + \sqrt{ln|\Sigma|}) \tag{25}$$

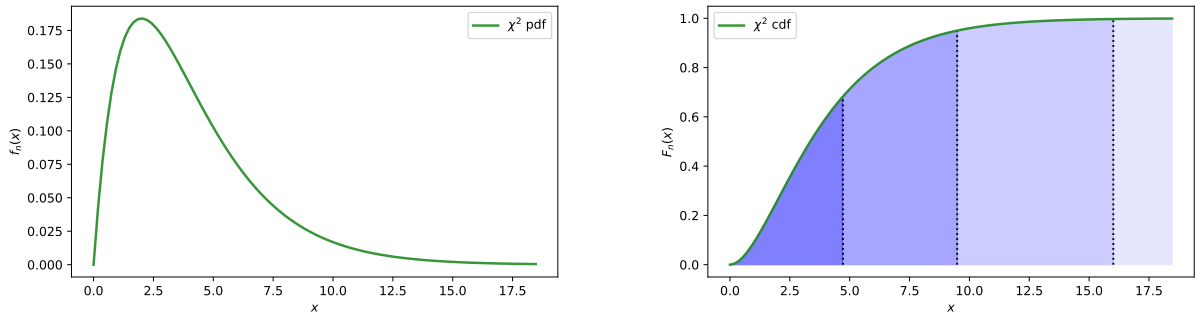
where the likelihood for the data point in question, $\mathcal{N}(x \mid \mu, \Sigma)$, is the value obtained by Equation 22.

Furthermore, for a spherical Gaussian distribution, the sum of squares of its n independent and identically distributed random variables follows a pre-determined distribution: the chi-square distribution with n degrees of freedom χ^2_n. The chi-square distribution with $n = 4$ is visualized in Figure 8. We see the probability density function $f_n(x)$ in 8a, and its cumulative distribution function $F_n(x)$ in 8b. As the Mahalanobis distance represents the distance to the mean of a distribution after the distribution has been normalized to spherical covariance, the squared Mahalanobis distance D^2 follows a chi-square distribution.

We can use the cumulative chi-square distribution to indirectly integrate over our cluster probability density, and determine the proportion of the distribution that is within a certain Mahalanobis distance, and how much falls outside. For each star, we indicate the proportion of the cluster distribution that falls further away than the Mahalanobis distance of the star according to

$$p(x \geq D^2) = 1 - \int_0^{D^2} f_n(x)\, dx = 1 - F_n(D^2) \tag{26}$$

Here the number of degrees of freedom is $n = 4$, as our data points are four-dimensional. Based on this we can choose to discard e.g. the 5% furthest away data points by choosing $1 - F_n(D^2) < 0.05$ - treating them as outliers - and accepting the rest of the data points as the 95% most likely cluster members. As can be seen in Figure 8b, this corresponds to the datapoints with a squared Mahalanobis distance of 9.5 or larger.



(a) Probability density function

(b) Cumulative distribution function

Figure 8: Probability density of the chi-square distribution with four degrees of freedom $f_n(x)$ (a) and its cumulative distribution function $F_n(x)$ (b). This chi-square distribution in this case describes the theoretical distribution of data points of a spherical four-dimensional Gaussian distribution as a function of their squared distance to the mean. The blue shaded regions represent common confidence intervals, corresponding to squared Mahalanobis distances covering 68%, 95%, 99.7% and 99.9% of the distribution. It can be noted that 95% of the distribution falls within a squared Mahalanobis distance of approximately 9.5.

3.4 Experiment

Here we summarize the methods described in the sections above and how they were used to obtain the results of Section 4. We impose quality cuts on the Gaia radial velocity sample and select halo stars according to $V - V_{LSR} > 210$, giving $N = 48880$ sources. We apply single linkage clustering in four dimensions: E, L_z, L and η (circularity), with features scaled to a range of $[-1, 1]$. We generate 100 artificial data sets simulating a halo without structure by scrambling the velocity components of the quality filtered radial velocity sample and recalculating the four clustering features. We then investigate each candidate cluster (or connected component) included in the linkage matrix returned by the single linkage algorithm, and compute the expected density for each cluster. The expected density is defined as the normalized count in the artificial data sets in the same region as the candidate cluster. We consider an elliptical cluster boundary defined by the principal components of each candidate cluster with axis lengths equalling two standard deviations of spread. If the density in this region is more than three standard deviations away from the mean expected density of the artificial data sets, according to Equation 12, we accept this as a significant cluster.

We then extract subgroups in velocity space (v_R, v_ϕ, v_z) by applying HDBSCAN separately on each cluster, with the parameters listed in Table 2. The result of this is another column indicating the labels for our substructures, where possible noise is also flagged.

Finally, we obtain an indication of membership probability for each star in a cluster by modelling the star as a multivariate Gaussian given their linearly propagated measurement uncertainties. By modelling the uncertainty in each star as a probability density function like this we can fit a multivariate Gaussian representing the cluster using an extreme deconvolution algorithm [36]. For each star, we then calculate how much of the probability density function of the cluster is within (or outside) the Mahalanobis distance of the probability density of the star by relating it to the cumulative chi-square distribution.

We use Python and the following libraries to implement our method in code: Vaex, for efficient handling of the data set, data exploration and error propagation [39]. Scipy, for implementation of the single linkage algorithm and chi-square distribution [40]. HDBSCAN for extracting substructure in velocity space [41]. The XDGMM Python wrapper for computing membership probabilities [42, 43]. We also use NumPy and Matplotlib for utility functions [44, 45].

Table 3: Summary statistics of significant clusters extracted in Gaia eDR3.

Statistic	Value
Clusters	419
Stars assigned to some cluster	26677
Members (range)	$[12, 9987]$
Members (average and standard deviation)	63.8 ± 496.6
Members (median)	24
Members (mode)	16

4 Results

Here we present the results obtained by applying the methods described in Section 3. The cluster and subgroup labels and the associated membership probabilities are available as a catalogue provided as additional material.

4.1 Integral of motion space

We extract 419 significant clusters in integral of motion space. Summary statistics for these are presented in Table 3. 55% of the stars, 26677 out of 48880, have been assigned to some significant structure. The number of members in a cluster ranges between 12 and 9987 with mean 63.8, while it is possible that the largest cluster in fact comprises multiple streams so close to each other that the algorithm is unable to distinguish between them. We see a histogram of the number of members in each cluster in Figure 9. As we only have six clusters larger than 200 members we truncate the histogram at the seventh largest cluster, having 182 members. We note that the vast majority of clusters (85%) have less than 50 members, with the median being 24 and the mode 16.

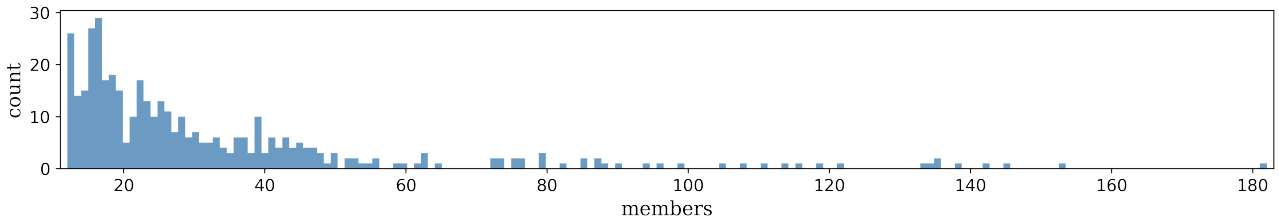


Figure 9: Histogram of the number of members in the clusters. The histogram is truncated for readability, and we have six clusters that exceed the range of the histogram, with 182, 225, 249, 379, 425, 1970, and 9987 members respectively.

The clusters are plotted for all combinations of our clustering features in Figure 10. Clusters can be found all over the data space. As structures are overlapping and it becomes quite a clutter to visualize all clusters in one figure, we plot the clusters over 15 subpanels in Figure 11. Here we see 27 or 28 clusters per subpanel in $E - L_z$ space. We plot the same clusters in $L_z - L_\perp$ space in Figure 12. We do not use the perpendicular angular momentum $L_\perp$ as a clustering feature, but some structures are known to have a distinct shape in this space, such as the Helmi stream [46]. The Helmi stream can be seen as the red brown cluster in the third panel in the second row around $L_\perp = 2000$ and $L_z = 1000$. The algorithm picks up on the Helmi stream as a single cluster, while it is possible that it consists of two substructures, as illustrated by the small gap between the left and right part of the cluster [18]. The same clusters are plotted in velocity space in Figure 13, where the shape of the structure we see aligns well with what is expected according to simulations [2, 47]. In the next section we look more closely into what subgroups exist for each cluster in velocity space. Finally, in Figure 14 we plot the distribution of stars not in assigned to any cluster. It is interesting to assess the level of smoothness in this distribution, but purely by the naked eye it is not quite possible to determine whether there may exist undetected clumps.

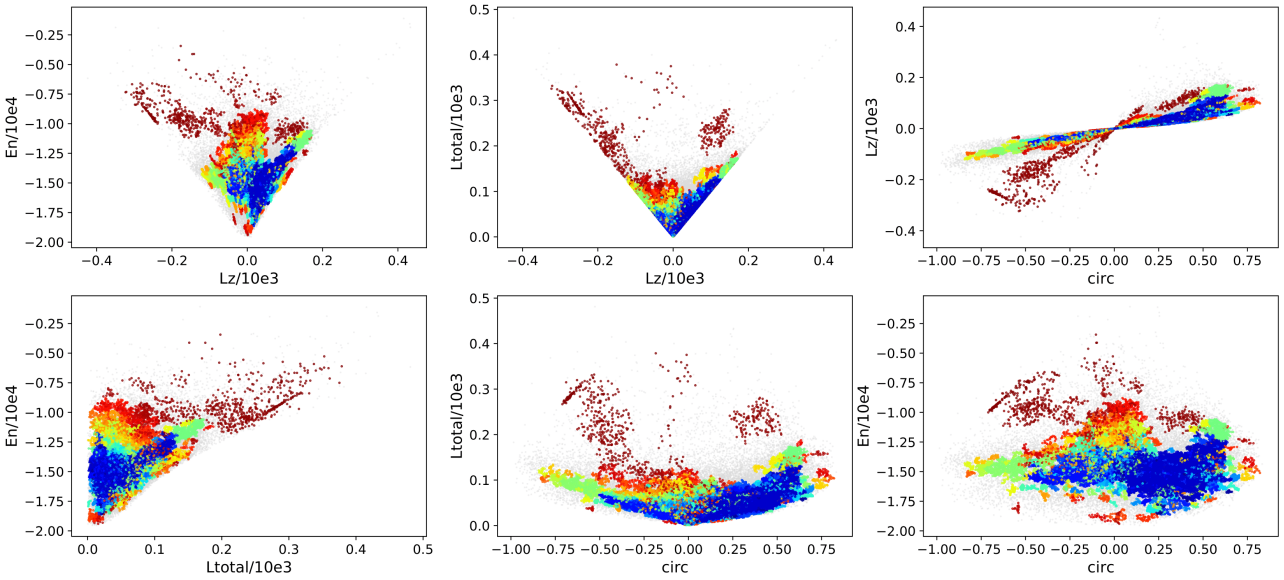


Figure 10: Statistically significant clusters plotted for each subspace. Please note that as we obtain 419 significant clusters, the colourmap above presents some clusters in the same or almost the same nuance.

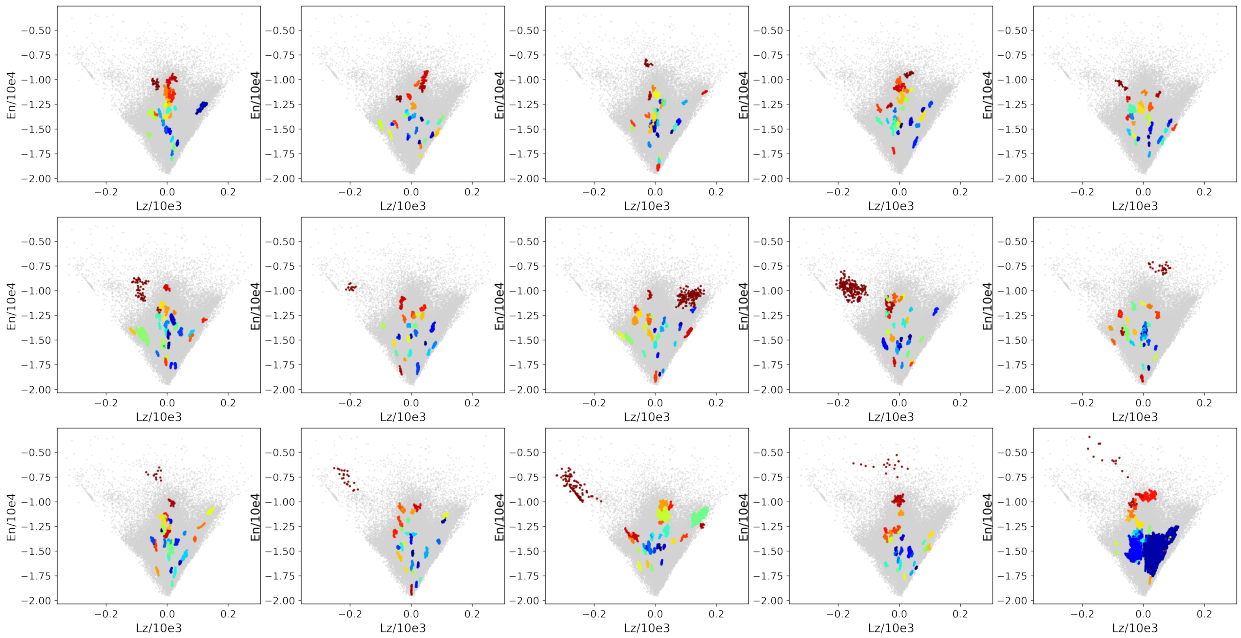


Figure 11: Statistically significant clusters in $E - L_z$ space, with 27-28 clusters per subpanel.

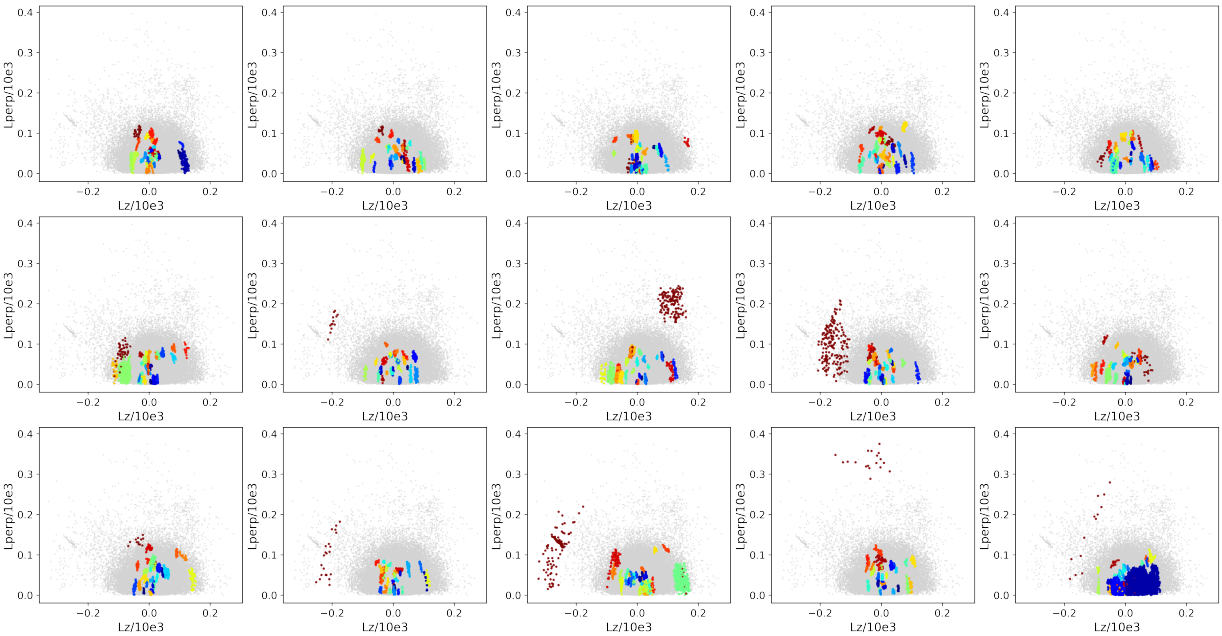


Figure 12: Statistically significant clusters in $L_z - L_\perp$ space, with 27-28 clusters per subpanel.

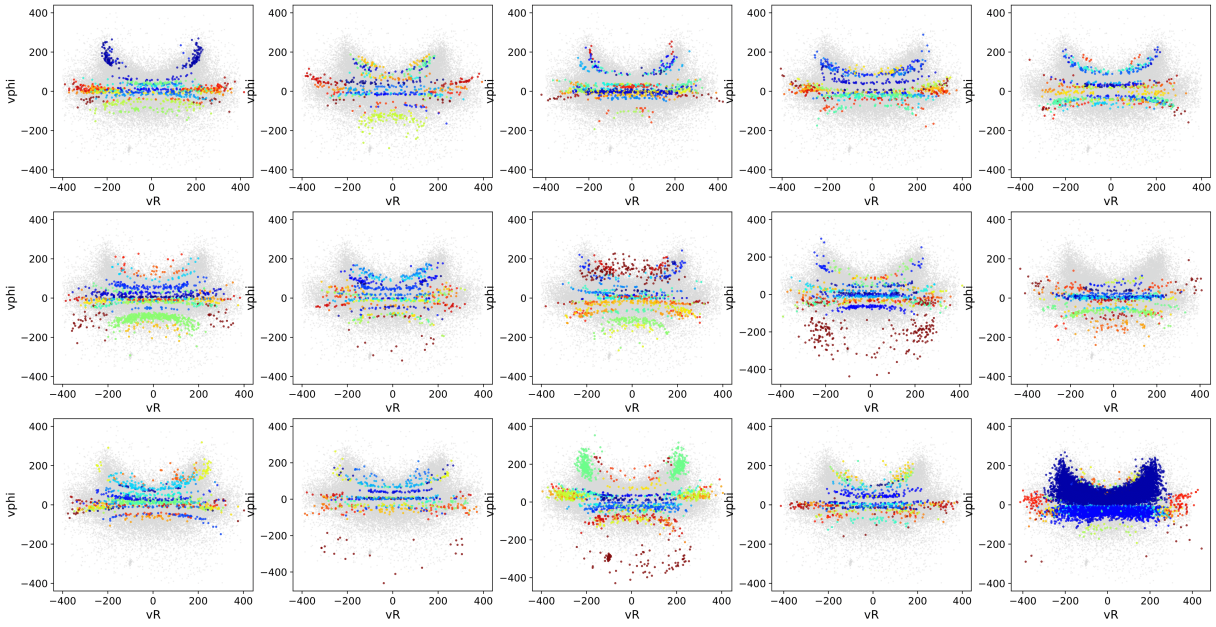


Figure 13: Statistically significant clusters in velocity space, with 27-28 clusters per subpanel.

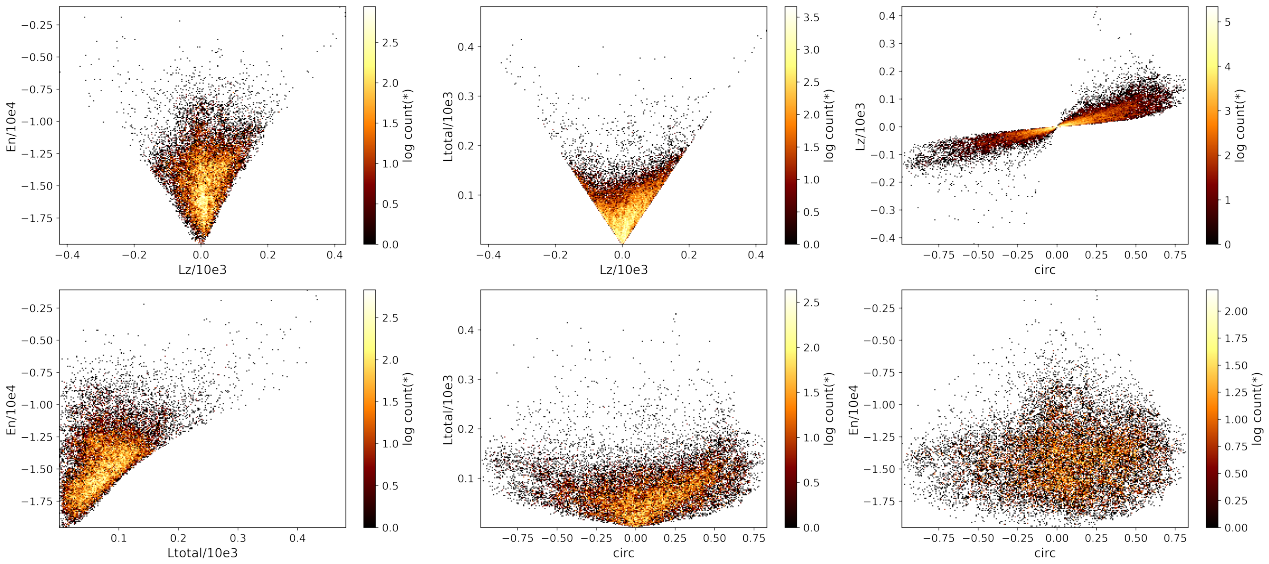


Figure 14: The distribution of stars not assigned to any cluster plotted for each subspace.

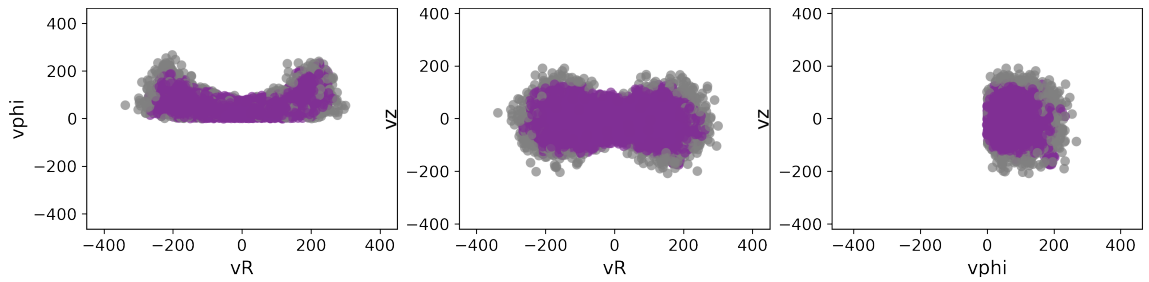
Table 4: Summary statistics on subgroups extracted in velocity space.

Statistic	Value
Total number of subgroups	799
Number of subgroups per cluster	$[1, 4]$
Average number of subgroups per cluster	1.9
Number of clusters with one subgroups	132
Number of clusters with two subgroups	207
Number of clusters with three subgroups	67
Number of clusters with four subgroups	13

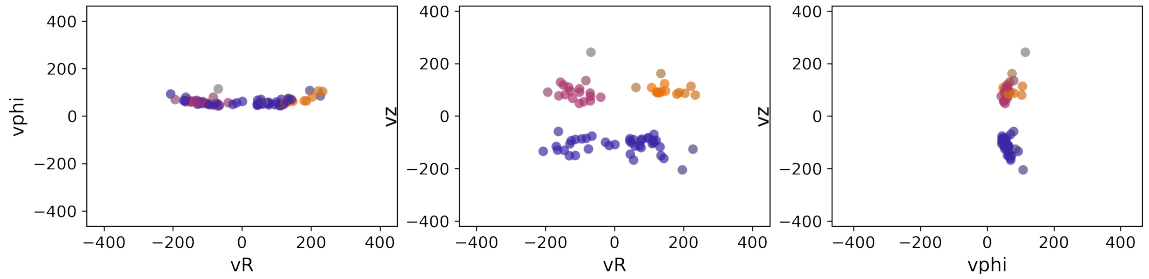
4.2 Velocity space

Here we present the subgroups that HDBSCAN separates our clusters into in velocity space. Summary statistics on this is presented in Table 4. We obtain 799 subgroups in velocity space, meaning that each cluster is split into 1.9 subgroups on average. We find between one and four subgroups for each cluster, two being the most frequent with 207 subgroups, and four the least common with 13 occurrences. As we have 419 clusters in integral of motion space, we do not display plots of the subgroups in velocity space for all of them, but select five that display some representative properties. These are displayed in Figure 15. The subpanels depict the following: cluster number 29 is the largest cluster that we extract in integral of motion space and this is determined to be a single group in velocity space. Possible noise in the outskirts of the distribution is displayed in grey. Cluster number 80 is one of the clusters split into three subgroups. Cluster number 274 is the second largest cluster containing four subgroups. Cluster 382 displays two particularly distinct clumps in $v_R - v_\phi$. Finally, cluster number 413 is a single dense component in v_ϕ-v_z but displays an interesting circular structure in v_R-v_z.

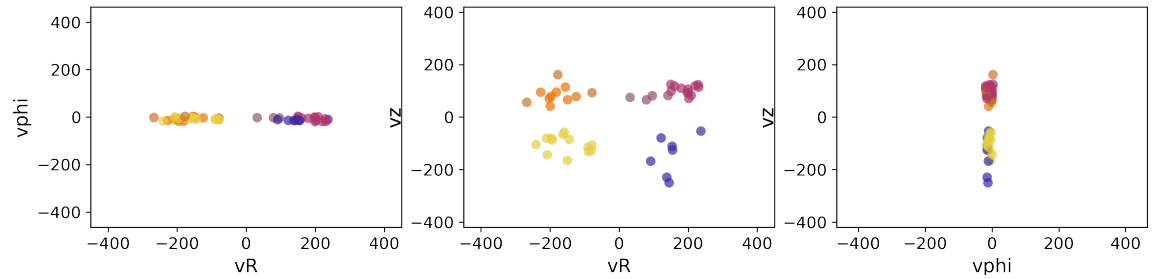
Furthermore we compute the dispersion in the principal components after applying PCA to the Cartesian velocity components of each cluster and subgroup. A small dispersion in these velocity components could indicate a particularly narrow and interesting stream. Here we only consider a volume of 2 kpc from the sun as otherwise large gradients in the velocities might inflate the dispersions. We compute the dispersion for clusters or subgroups with more than three members. Figure 16 displays histograms of these standard deviations for each cluster and subgroup within 2 kpc having at least four members. We clearly see lower dispersions within the subgroups, which is to be expected, as the signal that can be picked up in the latter case will be diluted if the cluster contains more than one subgroup in velocity space. We can see that there is a remarkable number of subgroups with an extremely small dispersion in the second principal component: 17 with a dispersion less than 1 km/s and 100 where the dispersion is smaller than 5 km/s. These groups are surely of interest when looking for streams with especially clean orbits.



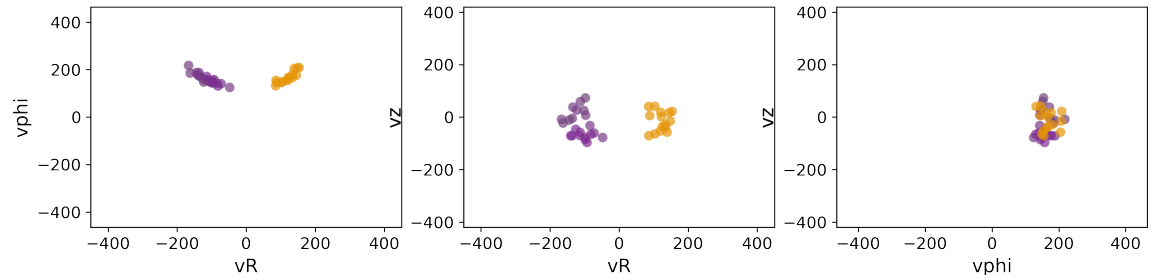
(a) Cluster number 29 (9987 members, one subgroup).



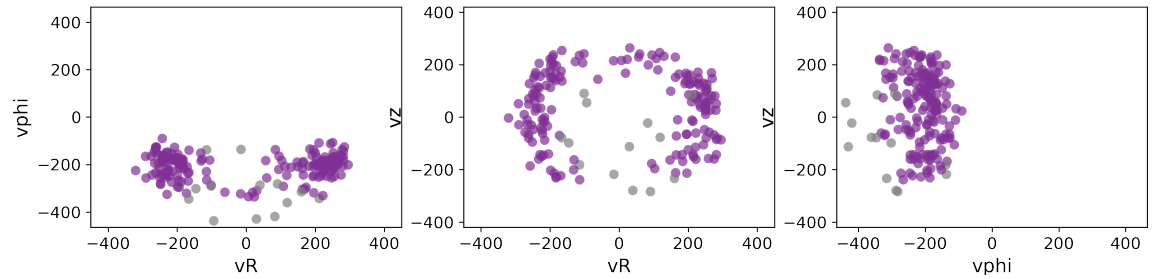
(b) Cluster number 80 (73 members, three subgroups).



(c) Cluster number 274 (47 members, four subgroups).

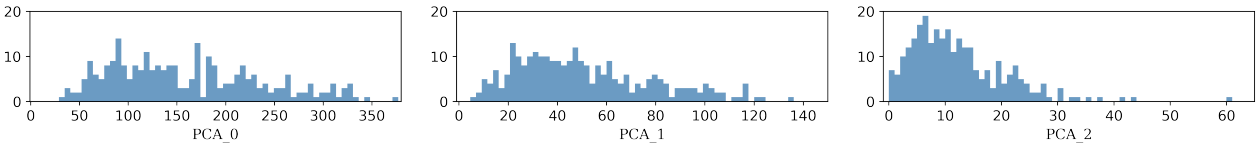


(d) Cluster number 382 (37 members, two subgroups).

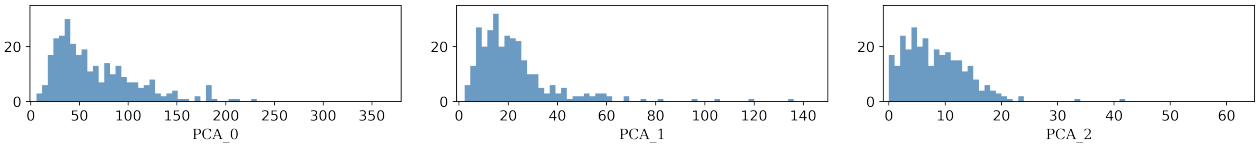


(e) Cluster number 413 (182 members, one subgroup).

Figure 15: Subgroups in velocity space for a selection of clusters. Possible noise is displayed in grey.



(a) Clusters



(b) Subgroups

Figure 16: Histograms of the standard deviation (km/s) along each principal component after applying PCA to the Cartesian velocity components of (a) each cluster and (b) each subgroup in velocity space. Here we only consider the stars within 2 kpc and compute the standard deviation if a cluster or subgroup has at least four members within this distance.

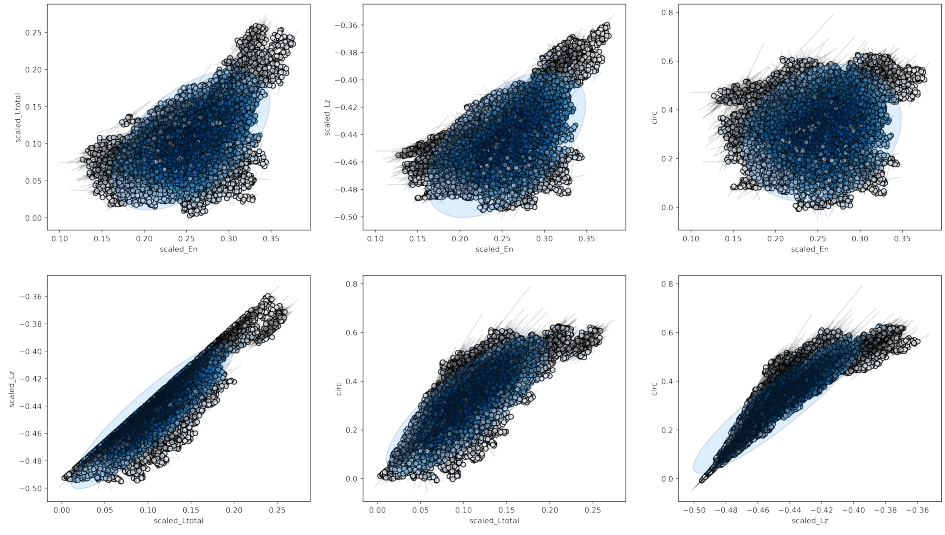
4.3 Membership probability

In Figure 17 we see the same clusters as presented in Figure 15, now illustrated by their uncertainties in all subspaces. The probability density of each data point is modelled by an ellipse with axis lengths equalling one standard deviation of spread, with the mean marked as a circle with black borders. The cluster distribution as fitted by the XDGMM algorithm is marked in light blue, for a region covering two standard deviations of spread along each principal axis. As can be seen, the uncertainties in each individual star are highly correlated in the subspaces, to the extent that the length of the minor axis is almost zero, forming lines in the plots instead of ellipses. The intensity of the color of a circle indicates the value obtained for membership probability, with darker blue indicating higher membership probability and lighter blue lower membership probability. As we can see, the high probability data points are mostly close to the mean of the cluster distribution, having smaller uncertainties, while lower probability data points lie towards the outer regions of the confidence ellipses.

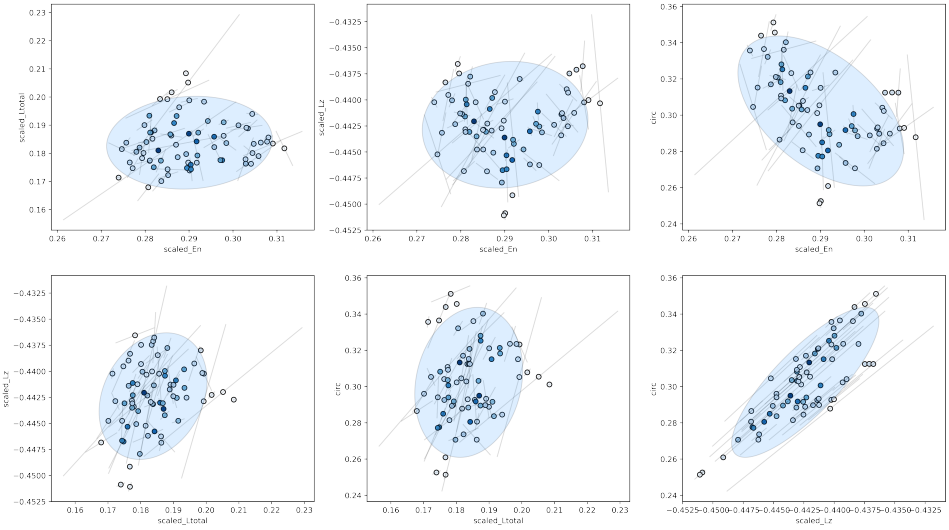
We can choose a significance level when deciding how strict we want to be when trying to detect outliers, according to the methods we describe in Section 3.3. If we choose the significance level to be 0.05, or choosing to consider the 5% furthest away data points as outliers, we discard 709 stars for cluster 29, 9 for cluster 80, 3 for cluster 274, 3 for cluster 382 and 13 for cluster 413. As this corresponds to $7\%, 12\%, 6\%, 8\%$ and 7% of the cluster members respectively for each cluster, we can see that these numbers are somewhat larger than the expected 5%. This may hint at that our clusters cannot be perfectly approximated with a multivariate Gaussian distribution. Please note that the outliers here were computed by consulting the chi-square distribution with four degrees of freedom and should not be confused with that we choose to visualize the cluster distribution with ellipses covering two standard deviations of spread along each axis in Figure 17.

We can evaluate the viability of the membership probability method by comparing the values obtained to the outliers as determined by HDBSCAN in Section 4.2. Computing the Pearson correlation coefficient between a data point being assigned to a subgroup in velocity space and its membership probability gives $r = 0.0957$, indicating a negligible to weak correlation. Similarly, we can choose to consider stars with a membership probability less than 0.05 as an outlier. Out of the 2397 stars labelled as outliers in velocity space, the membership probability also labels these as outliers in 19% of the 2397 cases. In velocity space we have 24280 stars in total assigned to some subgroup and here the membership probability gives a value higher than 0.05 for 86% of these 24280 stars. We obtain 3799 outliers in total for a membership probability threshold of 0.05.

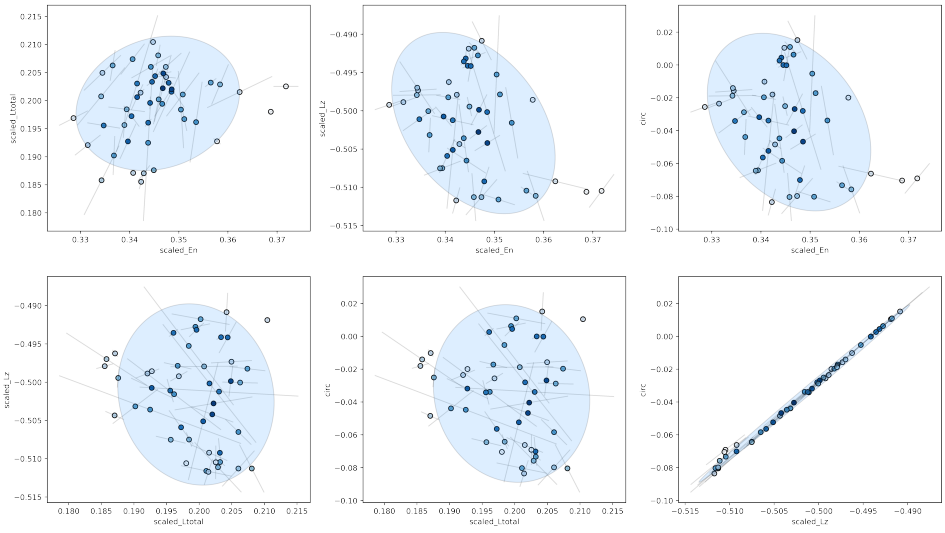
The results of evaluating the membership probability is a value in the range $[0, 1]$. It is in fact possible to evaluate the membership probability of a star belonging to any neighbouring cluster, if this analysis is wanted. We have not included this analysis in the thesis, instead the additional material of the thesis provides the membership probability only for the stars that have already been assigned to a cluster in Section 4.1.



(a) Cluster number 29

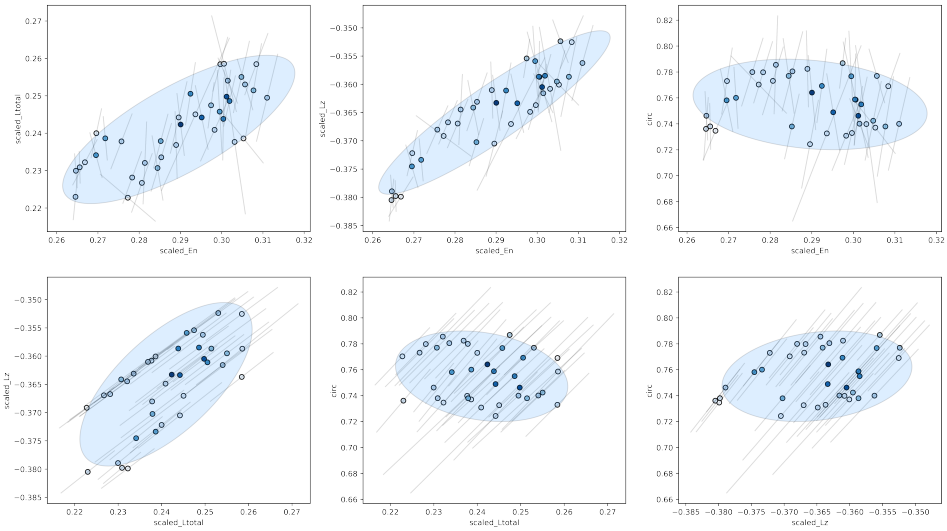


(b) Cluster number 80

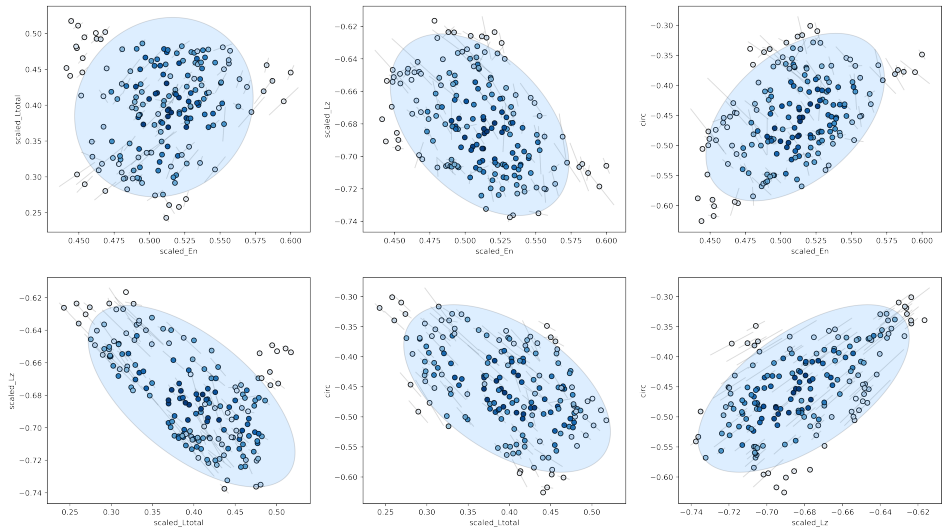


(c) Cluster number 274

Figure 17: Probability density distribution for a selection of clusters (light blue region) and member stars. Darker blue indicates a higher membership probability.



(d) Cluster number 382



(e) Cluster number 413

Figure 17: Probability density distribution for a selection of clusters (light blue region) and member stars. Darker blue indicates a higher membership probability.

5 Discussion

Here we discuss the interpretation of the results, the correctness aspects of the methods, and the considerations of the algorithm from a computational perspective.

5.1 Interpretation of results

Our results assign more than half of the stars in our data set to some significant cluster (see Table 3). This large a fraction is promising but not unrealistic according to predictions on substructure in the halo and other previous work [23]. While the clusters extracted are proven to be statistically significant overdensitites, they do not necessarily have a one-to-one correspondence with individual stellar streams. To begin with, accreted objects that are close to each other or possibly overlapping in our clustering space run a risk of being labelled as the same cluster. Due to always selecting the largest significant structure when making the flat cluster assignment, the algorithm will currently be unable to distinguish between separate objects that are located sufficiently close to each other, as the union of these will also display statistical significance according to our criterion. Ideally we would analyse the merging hierarchy of these clusters in order to see whether the cluster may actually consist of multiple different structures, but it is not certain that the merging hierarchy would be able to give unambiguous results. We can also analyse other properties to map the nature of the extracted clusters: location in velocity space, metallicities, and their HR-diagram. This could possibly enable identification of several streams being labelled as one.

Another consideration is that it is in fact possible that the full outer halo, and at least part of the inner, is built up by mergers [20]. If this is true, the use of our artificial reference halo may pose a paradox in the sense that while the small-scale structure is broken up in the artificial set, the average density for the full data space is still identical to our halo set. This may make it difficult for the algorithm to detect massive mergers with a large dispersion in the integrals of motion, as we require observing a remarkable difference in density compared to same region in the artificial halo. If the progenitor covers a large region, with perhaps multiple local overdensities, the algorithm may not be able to display statistical significance for the full region of the progenitor, and will instead indicate the local overdensities. It is highly likely that a notable part of our clusters represent this scenario, and that they are in fact local overdensities belonging to the same progenitor.

Furthermore, our distance-based algorithm may suffer from the fact that the data space is non-uniform: there are more stars in the lower energy regions of the V-shape we observe in $E - L_z$ space than for higher energies (see Figure 5). This, in combination with the algorithm finding local overdensities as argued in the previous paragraph, are likely the main reasons to why the algorithm is unable to extract for example the well established dwarf galaxy Gaia-Enceladus [48] (see Figure 3) as a single cluster. Gaia-Enceladus covers both the more high density region below in the V-shape and extends upwards towards less dense regions. The algorithm will perform merging steps purely based on the Euclidean distances between the data points, and since these distances will be smaller in higher density regions, the merging process may prioritize linking together components in this region. One way to correct for the non-uniform data space might be to normalize the data set by the average density of the region. However, this poses challenges with regards to preserving the correctness of the data space and not destroying the signal that we want to detect. Future work could investigate association between the smaller clusters that we think originate from Gaia-Enceladus, and how they relate to each other in the merging hierarchy.

Looking at the clusters we find, some of them are just plain (non-disrupted) globular clusters, which are well-mapped structures of the Milky Way. While these are not interesting in terms of finding streams, they display a strong clustering in both configuration and integral of motion space, and it is natural that our method picks up on these. While many of the other clusters detected display properties that align well with what is expected from streams, further analysis is needed in order to map their properties, which clusters correspond to previously identified structures, and whether some new structures may have been identified. Comparing Figure 3 and Figure 10, we can draw the conclusion that the extracted structures cover at least the same regions as previously established ones. By a preliminary manual comparison, some of the clusters can be linked to Gaia-Enceladus, the Helmi stream, Thamnos, Sagittarius and Sequoia [21, 46, 48–50].

There may be a slight bias in the data in the sense that it only covers the local stellar halo, and signal is not detectable if it is not strong enough in the solar neighbourhood, even though data points that are currently labelled as noise may be part of more distant streams for which we currently do not have enough data. Furthermore, brighter and more nearby stars will have more accurate measurements, while stars originating from regions with extremely high-density, such as the cores of globular clusters, will have less accurate measurements.

At large distances we also only have measurements from bright stars. Furthermore, some of our clusters are likely to be a result of contamination from the disk. The halo sample could potentially be further refined by combining our velocity based selection with metallicity information, as part of the halo is more metal poor than the disk [51].

The completeness of the data is also a discussion point when looking at the results from Section 4.2, especially when looking at the subgroups formed in velocity space. For some string-like structures, it possible that the split is a result of missing data in between subgroups, and that this is in fact the same portion of the stream, or subgroup in velocity space. As an example we can look it Figure 15b, where it might be possible that the pink and orange subgroup should actually be joined by some missing data around $v_R = 0$. Another thing to investigate in velocity space is how the results would change when varying the parameters, for example the velocity dispersions or the number of subgroups. Still, an advantage of HDBSCAN is its robustness in extracting clusters, so varying the parameters slightly should not result in drastically different results.

As previously mentioned, it is possible to treat the membership probability as an indication of outliers that could be filtered away as potential noise in a cluster. A way to assess the outcome of this method is cross-matching it with the outliers as determined by HDBSCAN in velocity space. The outliers do not match to a broad extent as illustrated by a mere 19% agreement when comparing the outliers found by the membership probability (using a significance level of 0.05) to the ones found by HDBSCAN. This indicates that we should not necessarily filter out stars falling below some threshold without further consideration. One aspect is that despite having an indication for the membership probability, there is still no ground truth for what the real cluster members are. The membership probability is also merely an indication of the how each data point falls with respect to the distribution of all stars assigned to the same cluster, and just because a star falls in the outskirts of this distribution it does not mean that it is not a true member. Similarly, for some other cluster we may need to use a much higher threshold in order to obtain the true members. Due to this speculative nature, I recommend using the membership probability primarily as an optional tool when investigating the properties of the extracted clusters as they are - for example comparing to outliers in the metallicity distribution. Overall, we should interpret the membership probability as a relative indication and which should be used in combination with analysis of other properties.

We also note that each cluster Gaussian is fitted based on all contributing members and having a lot of noise in a single cluster may scew the fit of the cluster Gaussian in an unfavourable way. This noise could be the result of a cluster with a very dense core, having had some random neighbours added to it by the merging process. To combat unrepresentative cluster distributions we may want to consider using methods to obtain more robust estimates of the Mahalanobis distances, for example using the minimum covariance determinant method [52]. It is also not known whether a multivariate normal density is a good representation of a cluster at all, as the single linkage algorithm can produce various non-ellipsoidal cluster shapes. However, ellipsoids are quite flexible geometries, and it can be argued that they are better than rectangular regions or dividing the data space into bins.

5.2 Correctness

An advantage of this method is the exhaustive approach - under the assumptions that we make, and from a probabilistic point of view, no significant structure goes undetected. However, there are always inherently uncertain parameters. The uncertainty in the underlying data remains a problem for any distance-based clustering algorithm, despite our efforts to indicate a membership probability after applying our clustering algorithm. We could solve this by instead of measuring the statistical significance as the difference in count between a candidate cluster and the same region in the artificial data sets ($N_{Ci} - N_{C_i}^{art}$), we could measure the difference in probability density spanning the same region. However, this method would require much higher computational complexity, as we would need to scan variable regions all over the data space, and subsequently obtain a measure to compare the total probability density of each region. The latter could theoretically be done by computing sums of integrals over each probability density included in a region, but as this is very computationally heavy to do with high precision in 4D, I do not recommend it at the moment of writing. However, it can be noted that this work does in fact take the uncertainty in N_{C_i} into account when determining the statistical significance of cluster C_i (see Equation 12), which makes not taking the measurement uncertainties into account in the clustering step acceptable.

Related to modelling the cluster boundary, we select the axes of this ellipsoid to have a length equalling two standard deviations of spread in the direction of each axis. This is equivalent to covering 95% of the data along each corresponding principal axis (in a one-dimensional case). As we model our requirement of statistical significance according to $N_{C_i} - N_{C_i}^{art} > 3\sigma_i$, without mapping and counting the cluster members to PCA-space, we may want to correct for the region actually not covering 100% of the distribution of the data points in C_i. In

fact, it can be argued that we again should consult the chi-square distribution with four degrees of freedom in order to obtain axis lengths that cover a sufficient amount of the distribution in 4D. We could also formulate our criterion for statistical significance as $r * N_{C_i} - N_{C_i}^{art} > 3\sigma_i$ where r is the fraction of the distribution that theoretically is covered by our current choice of axes lengths. Alternatively, we could map also the cluster members to its PCA space and count how many members fall within the defined ellipsoid. On the other hand, the cluster boundary defined may also work in the opposite direction, providing a fit which produces somewhat overestimated counts in the artificial data sets instead of underestimated. This could happen in case that the ellipsoid turns out to produce regions with unnecessary empty space within the cluster distribution. The definition of the cluster boundary is a choice to be made and it would also be possible to observe how the results change if the length of the axes of the cluster boundary ellipsoid is varied.

The single linkage algorithm can also be sensitive to noise, and a single data point can form a bridge between two structures that should actually be separate. The effect of this could be decreased by using the mutual reachability distance to form the linkage matrix, similarly to HDBSCAN [28]. Using mutual reachability has proved to combat noise well in many applications of the single linkage algorithm. However, if there exists two separate components in our halo set, only linked together by some random noise, it is possible that our method for determining statistical significance is already able to handle this. It would be likely that the union of these structures do not display statistical significance, while treating them separately will. The extent of this effect requires further investigation.

Furthermore, the parameter choices for extracting structures in velocity space are somewhat heuristic, but as it is difficult to define exactly what characteristics in subgroups the astronomers are interested in, the results form an acceptable indication of the subgroups that can be found. The clustering in this space is also aided by that the data set encompassing a single cluster is far less complex than the full halo set, and that the stars in a cluster already have been established to belong to a significant overdensity.

We would also like to assess the suitability of modelling the uncertainties in our stars as Gaussian probability densities. In fact, a non-linear transformation on a Gaussian measurement uncertainty will be guaranteed to not have a Gaussian distribution. However, linear error propagation has been shown to provide good approximations of the real distributions when the errors are small [34], and the uncertainties in our halo set have been subject to quite strict quality cuts.

5.3 Efficiency and Scalability

The method we develop is computationally quite efficient, with the single linkage algorithm having a computational and memory complexity of $O(N^2)$. Due to the similarity between the single linkage algorithm and graph theory the computational complexity of the method can be reduced to $O(N\ log(N))$ by using for example Borůvka's algorithm for finding the minimum spanning tree [53]. The memory complexity could be reduced by computing the distance between all data points according to a tree-structure or treating the clustering space as n-orthotopes (or four-dimensional rectangles) for which you would only have to calculate the distance matrices within each rectangle together with its neighbouring volumes. This should reduce the memory complexity with a factor of $O(\sqrt{N})$.

The heaviest step of the algorithm is evaluating the statistical significance of each candidate cluster C_i, as it involves repeated PCA analysis. We map all members (falling within a sufficiently large rectangular region around C_i) in the 100 artificial halo sets to the PCA space defined by C_i, and count the number of members within the ellipsoid we define. Still, this step is fully parallelizable, and lasts "only" a couple of hours running on 20 cores for our problem size of $N = 48880$ halo stars and 4888000 stars in the artificial halo sets.

Analysing the efficiency of our methods in Section 3.2 and 3.3, HDBScan and XDGMM are both efficient algorithms, and applying them separately on each cluster is inherently fast as the input size is small.

5.4 Future work

The Gaia radial velocity sample which we use in this work is actually only a small subset of the full data set, and there is much more data available, but with a missing value for the radial velocity. We have speculated in developing a method to infer clusters in the 5D data sample, but for the scope of this thesis this falls into future work. We could for example take the verified clusters found in the 6D sample, and impute (or infer) values in radial velocity in stars from the 5D sample. We could then assess the fit to an already established cluster. One could also impose other constraints, such as a match in metallicity, available velocity components and HR-diagram. However, imputing the radial velocities is a very difficult problem and there would be some highly speculative elements in this endeavour.

Future work should also examine the nature of the analytic distribution of the error in integral of motion space, to confirm that the peak of the probability density function representing a star and its measurement errors in integral of motion space in fact lies at the observed value, and has not been shifted by the non-linear transformations required to calculate the integrals of motion from the raw measurements. Currently we assume that the peak of the probability density function representing a star corresponds to the observed integral of motion value, but if this is not the case, we should shift the value of each data point we cluster on by a similar vector in order to obtain the most accurate results.

Furthermore, it will be a large task to map out the dynamics and properties of the clusters we extract in integral of motion space, together with their relationship to previously established structures. We also want to analyse their metallicity and age estimates. This is in fact currently undergoing work, and we expect to show further results related to this in a few months from the current time of writing.

5.5 Conclusion

The thesis takes some important steps towards a more data-driven method of detecting substructure in integral of motion space of the Milky Way halo, and is able to assign 55% of our halo set of $N = 48800$ stars to some statistically significant cluster. A large portion of the detected structures display promising stream-like properties. The algorithm finds 419 clusters in total, with the median in cluster size being 24. It is likely that some clusters represent local overdensities of the same orbiting progenitor, for example in the case of Gaia-Enceladus. One of the main challenges for our distance-based clustering algorithm is the non-uniform data space, with less stars at higher energies. While the results look promising, further investigation into the properties of the extracted clusters is needed. Future work encompasses mapping of their dynamics and characteristics, together with their relationship to previously discovered structures.

Acknowledgements

I would like to direct a warm thank you to my supervisors. To Michael Biehl from the computer science department: Your serious expertise in machine learning and scientific computing provided consistently relevant input on the process, and being one of my favourite teachers at the CS program it was a pleasure to get to work with you. To Amina Helmi from the astronomy department: Your serious involvement in this work has been crucial for the quality of the results, and getting to work with such an excellent scientist and leader has been inspiring. Also a thank you to the Dynamics of Galaxies research group at the Kapteyn Astronomical Institute, it was an honour to participate with you during the course of the thesis. A heartfelt thank you to Helmer Koppelman for coming up with the initial project idea and providing important mentoring on the astronomy aspects of the problem, as well as on other professional and private aspects of life. Thank you to my parents for always giving me the opportunity to follow my dreams and get the best out of myself.

References

[1] Volker Springel, Simon DM White, Adrian Jenkins, Carlos S Frenk, Naoki Yoshida, Liang Gao, Julio Navarro, Robert Thacker, Darren Croton, John Helly, et al. Simulations of the formation, evolution and clustering of galaxies and quasars. *nature*, 435(7042):629–636, 2005.

[2] Amina Helmi and P Tim de Zeeuw. Mapping the substructure in the galactic halo with the next generation of astrometric satellites. *Monthly Notices of the Royal Astronomical Society*, 319(3):657–665, 2000.

[3] Timo Prusti, JHJ De Bruijne, Anthony GA Brown, A Vallenari, C Babusiaux, CAL Bailer-Jones, U Bastian, M Biermann, DW Evans, L Eyer, et al. The gaia mission. *Astronomy & Astrophysics*, 595:A1, 2016.

[4] F Van Leeuwen, JHJ De Bruijne, F Arenou, J Bakker, R Blomme, G Busso, C Cacciari, J Castañeda, A Cellino, M Clotet, et al. Gaia dr2 documentation. *Gaia DR2 documentation, European Space Agency; Gaia Data Processing and Analysis Consortium. Online at https://gea. esac. esa. int/archive/documentation/GDR2/*, 2018.

[5] Anthony GA Brown, A Vallenari, T Prusti, JHJ De Bruijne, C Babusiaux, M Biermann, Gaia Collaboration, et al. Gaia early data release 3: Summary of the contents and survey properties. *arXiv preprint arXiv:2012.01533*, 2020.

[6] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[7] Brian S Everitt, Sabine Landau, Morven Leese, and Daniel Stahl. Cluster analysis 5th ed, 2011.

[8] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[9] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[10] Amina Helmi. Streams, substructures and the early history of the milky way. *arXiv preprint arXiv:2002.04340*, 2020.

[11] Alexander L Muratov and Oleg Y Gnedin. Modeling the metallicity distribution of globular clusters. *The Astrophysical Journal*, 718(2):1266, 2010.

[12] Hannu Karttunen, Pekka Kröger, Heikki Oja, Markku Poutanen, and Karl Johan Donner. *Fundamental astronomy*. Springer, 2016.

[13] Andrew Gould. An upper limit on the granularity of the local stellar halo. *The Astrophysical Journal Letters*, 592(2):L63, 2003.

[14] O Bienaymé, AC Robin, and B Famaey. Quasi integral of motion for axisymmetric potentials. *Astronomy & Astrophysics*, 581:A123, 2015.

[15] Radosław Poleski. Transformation of the equatorial proper motion to the galactic system. *arXiv preprint arXiv:1306.2945*, 2013.

[16] Dean RH Johnson and David R Soderblom. Calculating galactic space velocities and their uncertainties, with an application to the ursa major group. *The Astronomical Journal*, 93:864–867, 1987.

[17] Helmer Koppelman, Amina Helmi, and Jovan Veljanoski. One large blob and many streams frosting the nearby stellar halo in gaia dr2. *The Astrophysical Journal Letters*, 860(1):L11, 2018.

[18] Helmer H Koppelman, Amina Helmi, Davide Massari, Sebastian Roelenga, and Ulrich Bastian. Characterization and history of the helmi streams with gaia dr2. *Astronomy & Astrophysics*, 625:A5, 2019.

[19] The European Space Agency. Gaia archive at esa v2.8. https://gea.esac.esa.int/archive/, 2018. Online; accessed on 28th of May 2020.

[20] Amina Helmi, Jovan Veljanoski, Maarten A Breddels, Hao Tian, and Laura V Sales. A box full of chocolates: The rich structure of the nearby stellar halo revealed by gaia and rave. *Astronomy & Astrophysics*, 598:A58, 2017.

[21] Helmer H Koppelman, Amina Helmi, Davide Massari, Adrian M Price-Whelan, and Tjitske K Starkenburg. Multiple retrograde substructures in the galactic halo: A shattered view of galactic history. *Astronomy & Astrophysics*, 631:L9, 2019.

[22] Nicholas W Borsato, Sarah L Martell, and Jeffrey D Simpson. Identifying stellar streams in gaia dr2 with data mining techniques. *Monthly Notices of the Royal Astronomical Society*, 492(1):1370–1384, 2020.

[23] Rohan P Naidu, Charlie Conroy, Ana Bonaca, Benjamin D Johnson, Yuan-Sen Ting, Nelson Caldwell, Dennis Zaritsky, and Phillip A Cargile. Evidence from the h3 survey that the stellar halo is entirely comprised of substructure. *The Astrophysical Journal*, 901(1):48, 2020.

[24] D Katz, P Sartoretti, M Cropper, P Panuzzo, GM Seabroke, Y Viala, K Benson, R Blomme, Gérard Jasniewicz, A Jean-Antoine, et al. Gaia data release 2-properties and validation of the radial velocities. *Astronomy & Astrophysics*, 622:A205, 2019.

[25] Andrew R Wetzel. On the orbits of infalling satellite haloes. *Monthly Notices of the Royal Astronomical Society*, 412(1):49–58, 2011.

[26] George Efstathiou, Carlos S Frenk, Simon DM White, and Marc Davis. Gravitational clustering from scale-free initial conditions. *Monthly Notices of the Royal Astronomical Society*, 235:715–748, 1988.

[27] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[28] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[29] John C Gower and Gavin JS Ross. Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 18(1):54–64, 1969.

[30] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[31] JA Wilson. Volume of n-dimensional ellipsoid. *Sciencia Acta Xaveriana*, 1(1):101–6, 2010.

[32] Thomas A. Severini. *Elements of Distribution Theory*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2005.

[33] R Luck and JW Stevens. A simple numerical procedure for estimating nonlinear uncertainty propagation. *ISA transactions*, 43(4):491–497, 2004.

[34] Joel Tellinghuisen. Statistical error propagation. *The Journal of Physical Chemistry A*, 105(15):3917–3921, 2001.

[35] Y. L. Tong. *The Multivariate Normal Distribution*. Springer New York, 1990.

[36] Jo Bovy, David W Hogg, Sam T Roweis, et al. Extreme deconvolution: Inferring complete distribution functions from noisy, heterogeneous and incomplete observations. *Annals of Applied Statistics*, 5(2B):1657–1677, 2011.

[37] Guillermo Gallego, Carlos Cuevas, Raul Mohedano, and Narciso Garcia. On the mahalanobis distance classification criterion for multidimensional normal distributions. *IEEE Transactions on Signal Processing*, 61(17):4387–4396, 2013.

[38] Thomas R Etherington. Mahalanobis distances and ecological niche modelling: correcting a chi-squared probability error. *PeerJ*, 7:e6678, 2019.

[39] Maarten A. Breddels and Jovan Veljanoski. Vaex: big data exploration in the era of gaia. *Astronomy Astrophysics*, 618:A13, Oct 2018.

[40] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[41] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11), mar 2017.

[42] Tom Holoien, Phil Marshall, and Wechsler Risa. Xdgmm release 1.1. https://github.com/tholoien/XDGMM, 2020.

[43] Thomas W. S. Holoien, Philip J. Marshall, and Risa H. Wechsler. EmpiriciSN: Re-sampling Observed Supernova/Host Galaxy Populations Using an XD Gaussian Mixture Model. , 153(6):249, June 2017.

[44] Charles R. Harris, K. Jarrod Millman, St'efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern'andez del R'ıo, Mark Wiebe, Pearu Peterson, Pierre G'erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[45] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3):90–95, 2007.

[46] Amina Helmi, Simon DM White, P Tim De Zeeuw, and HongSheng Zhao. Debris streams in the solar neighbourhood as relicts from the formation of the milky way. *Nature*, 402(6757):53–55, 1999.

[47] Facundo A Gómez, Amina Helmi, Anthony GA Brown, and Yang-Shyang Li. On the identification of merger debris in the gaia era. *Monthly Notices of the Royal Astronomical Society*, 408(2):935–946, 2010.

[48] Amina Helmi, Carine Babusiaux, Helmer H Koppelman, Davide Massari, Jovan Veljanoski, and Anthony GA Brown. The merger that led to the formation of the milky way's inner stellar halo and thick disk. *Nature*, 563(7729):85–88, 2018.

[49] Heidi Jo Newberg, Brian Yanny, Connie Rockosi, Eva K Grebel, Hans-Walter Rix, Jon Brinkmann, Istvan Csabai, Greg Hennessy, Robert B Hindsley, Rodrigo Ibata, et al. The ghost of sagittarius and lumps in the halo of the milky way. *The Astrophysical Journal*, 569(1):245, 2002.

[50] GC Myeong, NW Evans, Vasily Belokurov, JL Sanders, and SE Koposov. Discovery of new retrograde substructures: the shards of ω centauri? *Monthly Notices of the Royal Astronomical Society*, 478(4):5449–5459, 2018.

[51] Lorenzo Posti, Amina Helmi, Jovan Veljanoski, and Maarten A. Breddels. The dynamically selected stellar halo of the galaxy with gaia and the tilt of the velocity ellipsoid. *Astronomy Astrophysics*, 615:A70, Jul 2018.

[52] Hamid Ghorbani. Mahalanobis distance and its application for detecting multivariate outliers. *Facta Univ Ser Math Inform*, 34:583–95, 2019.

[53] Otakar Boruvka. O jistém problému minimálním. *Práce Mor. Prırodved. Spol. v Brne (Acta Societ. Scienc. Natur. Moravicae)*, 3(3):37–58, 1926.