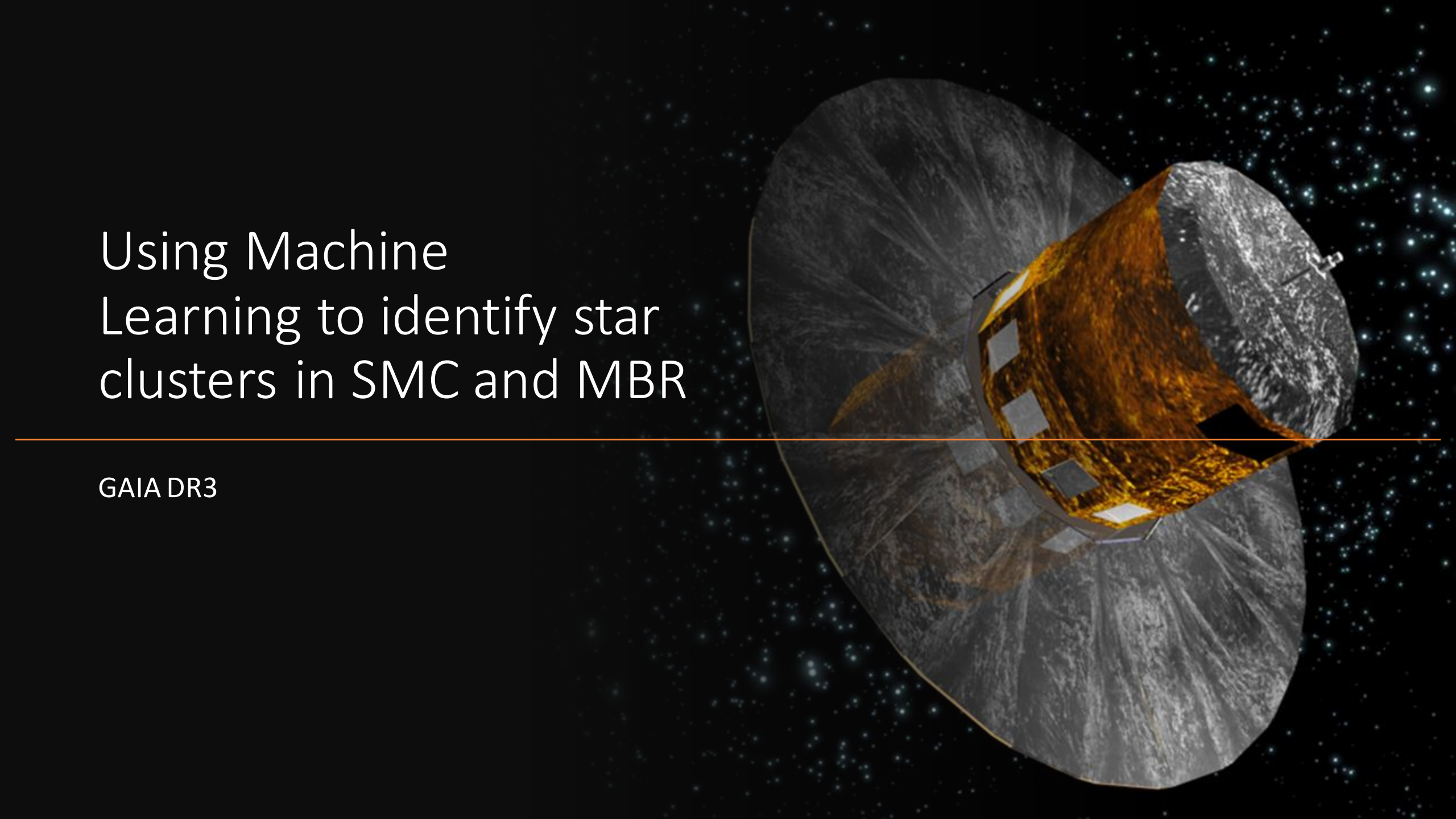
Using Machine Learning to identify star clusters in SMC and MBR

GAIA DR3

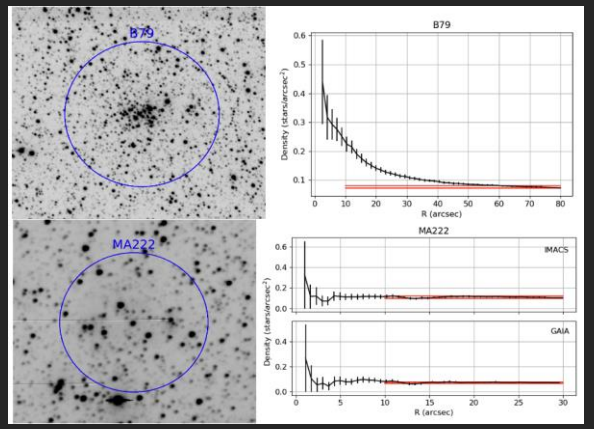


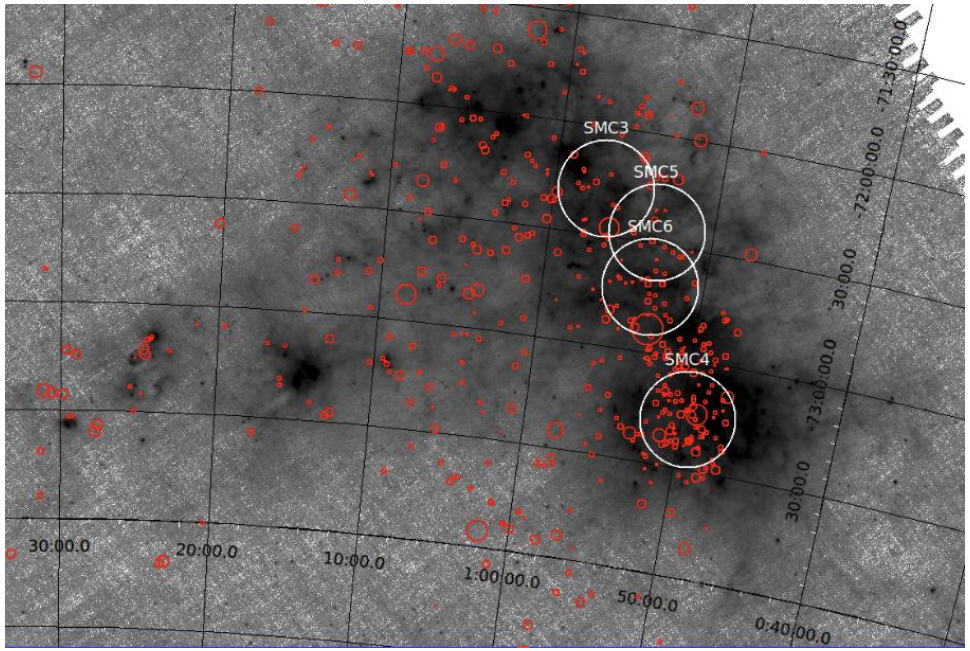
What we want to do and how

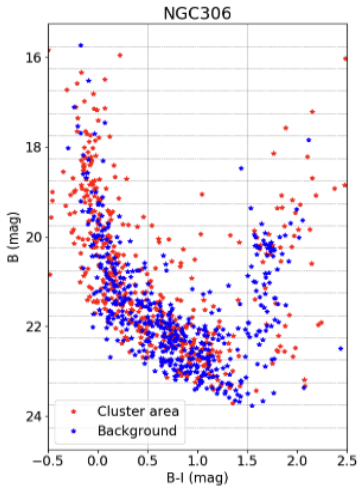
- **Automate** the **detection** of **star clusters** in SMC regions and MBR by using **Machine Learning** algorithms
- The **ML** method should have the following **characteristics**:
  - no need to specify the number of clusters beforehand
  - reproducible
  - well-defined steps to run it
  - includes the notion of noise (background)
- The ML algorithm we chose is **DBSCAN**. Our approach is presented in a paper (to be submitted to **MNRAS**) with regard to SMC cluster detection.
- Today we will extend the results to **MBR** as well

Fields under consideration, cluster classes

- There are **90 objects** in the Bica et al. (2020) catalogue with classifications C (cluster), CA (cluster-association) and CN (cluster-nebula), that lie in our SMC fields.
- For these 90 systems, we constructed **CMDs and radial profiles**.
- "**Quality class**" based on **3 criteria**: visual inspection, examination of the radial profiles, cluster CMD.







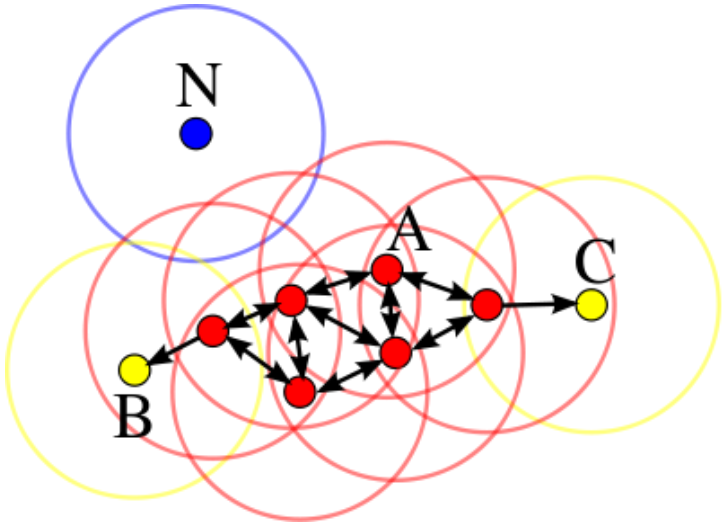
Class	characterization	Criteria satisfied
3	**Considered as certain clusters**	3/3
2	Probable clusters	2/3
1	Not probable clusters	1/3
0	Not confirmed as clusters	0/3

DBSCAN preliminaries

The algorithm requires two parameters: **ε (eps) and the minimum number of points required to form a dense region (minPts)**.

It starts with an arbitrary starting point that has not been visited. This point's ε-neighborhood is retrieved, and if it contains sufficiently many points (minPts), a cluster is started.

MinPts represents the minimum number of points within the eps distance, while the eps distance corresponds to the radius of a circle where MinPts are found



Here minPts=4 and eps is the radius of the circles:

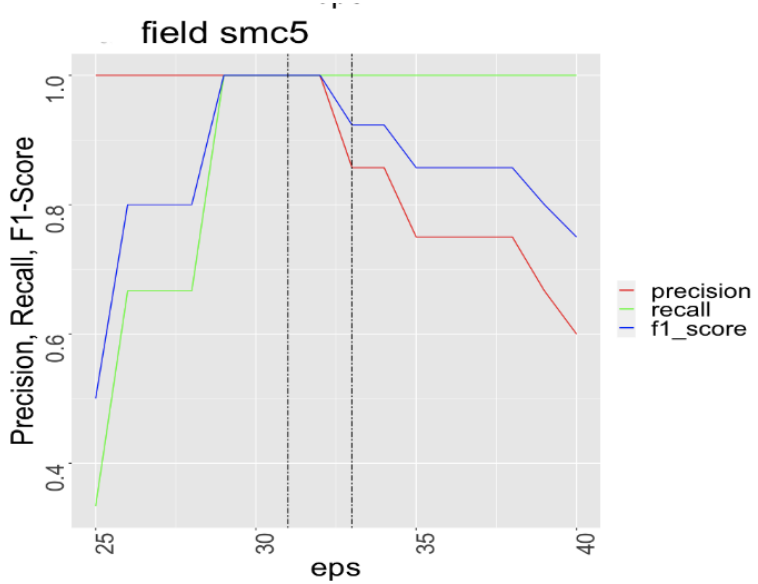
A: core point (in eps radius at least 4 points are contained)

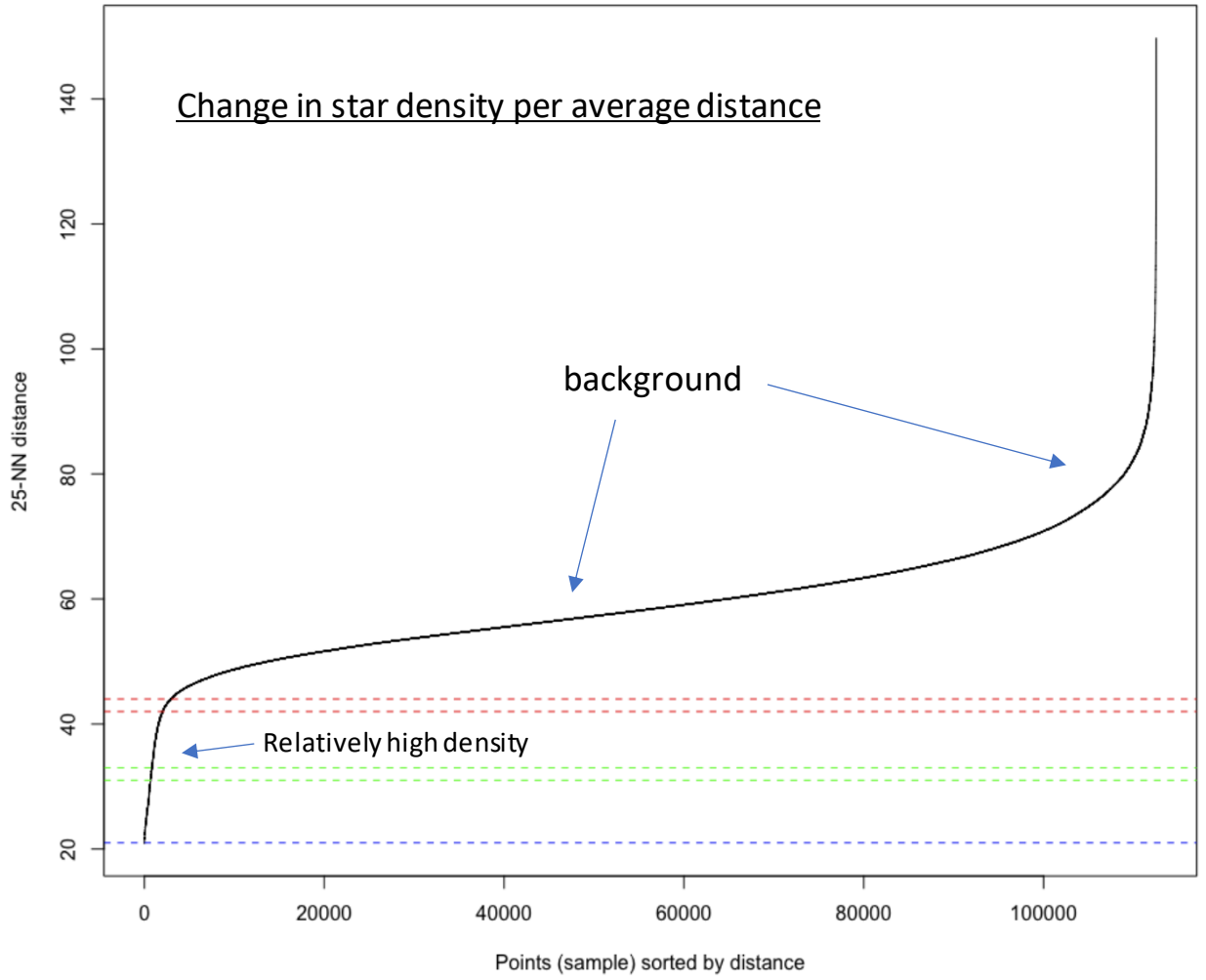
B,C: reachable points (from A)

N: background noise

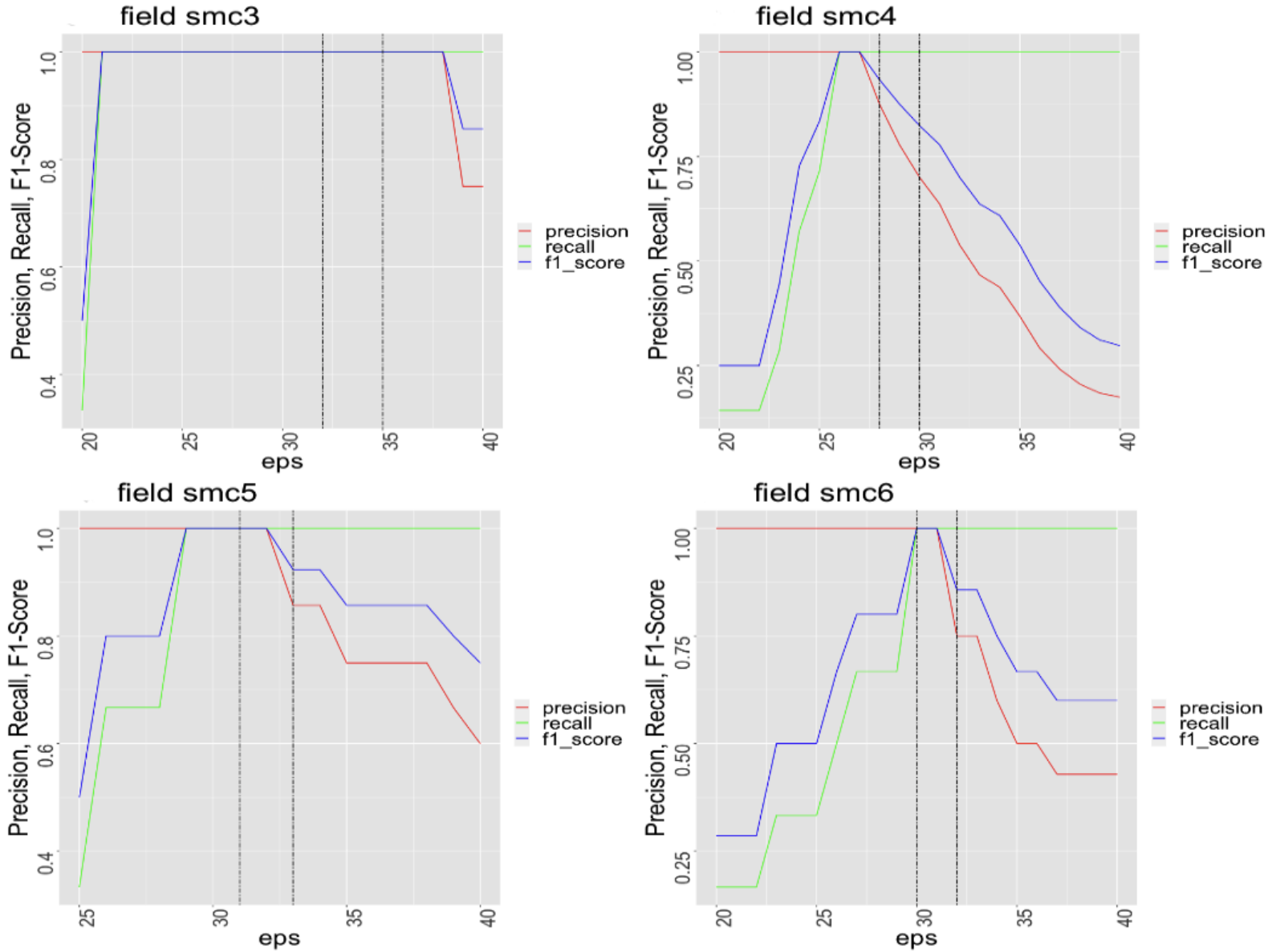
SMC5 Approach 1: "knee method"

- We adopted **MinPts=25** since we ran tests between 10 to 60, all were consistent.
- We need to specify **eps**.
- We use a kNN plot:
  y-axis : avg distance of 25 points from the selected one
  x-axis : number of points per avg distance
- The green horizontal lines in kNN plot (eps chosen) become the dotted vertical lines in P-R-F1 plot.





SMC fields: P,R,F-1 plots vs eps

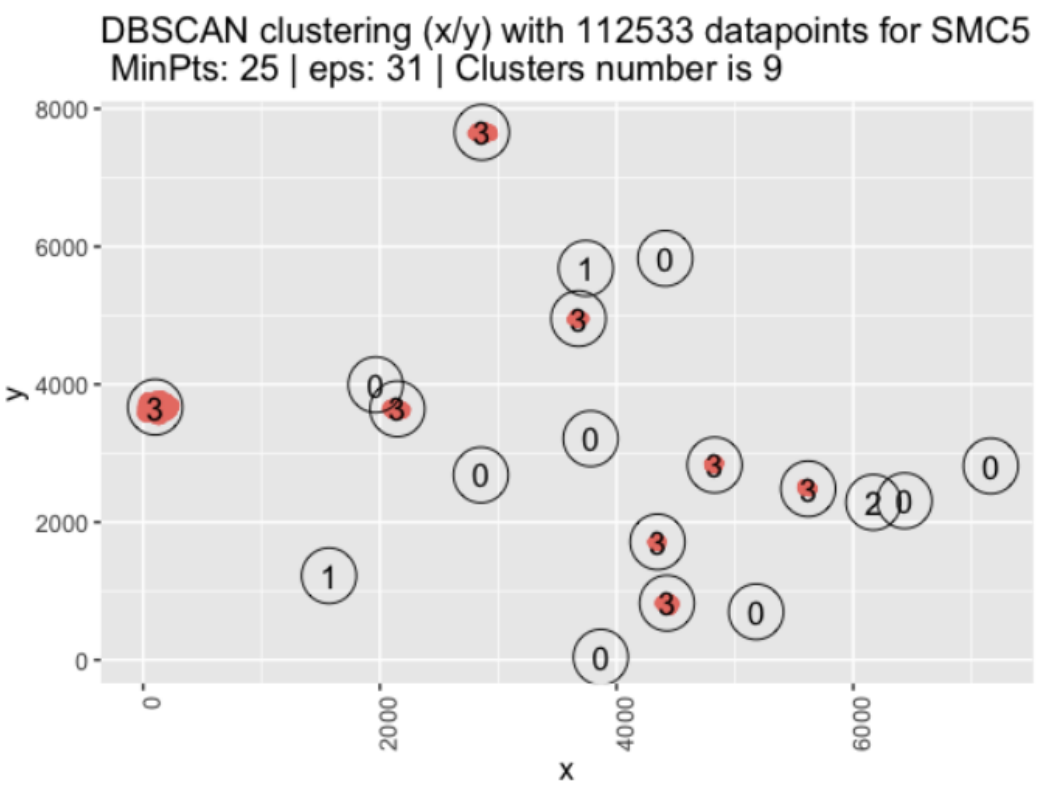


Eps selection seems to be an optimal one.

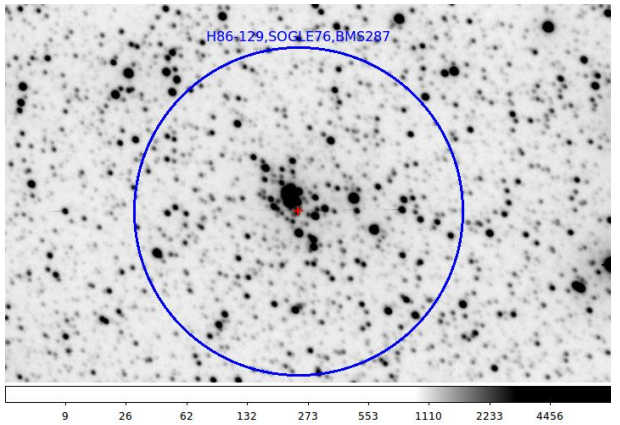
Recall (how many TPs are found regardless of FPs) is 100% for high eps values (green line).

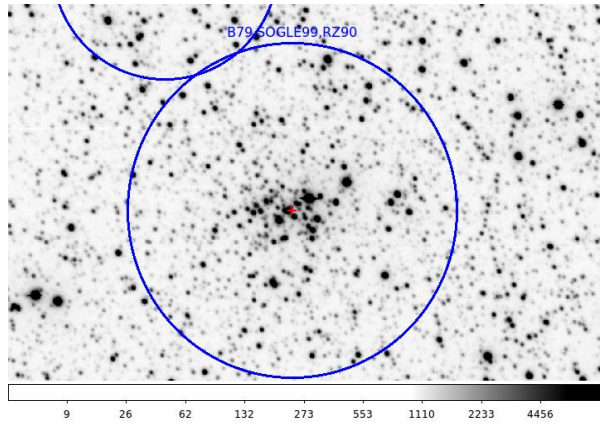
Is there a way to distinguish between FPs and TPs?
This gives us a hint to use another method, described next.

SMC5: How clusters look like in DBSCAN



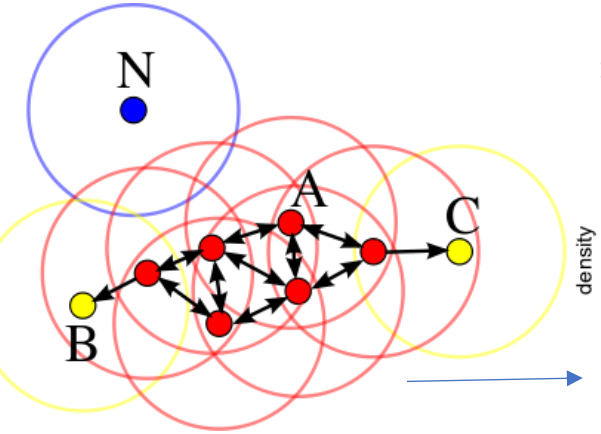
For the entire SMC5 starfield DBSCAN found the clusters in red, which are class 3 vs nothing in classes 0,1,2 using the optimal eps.



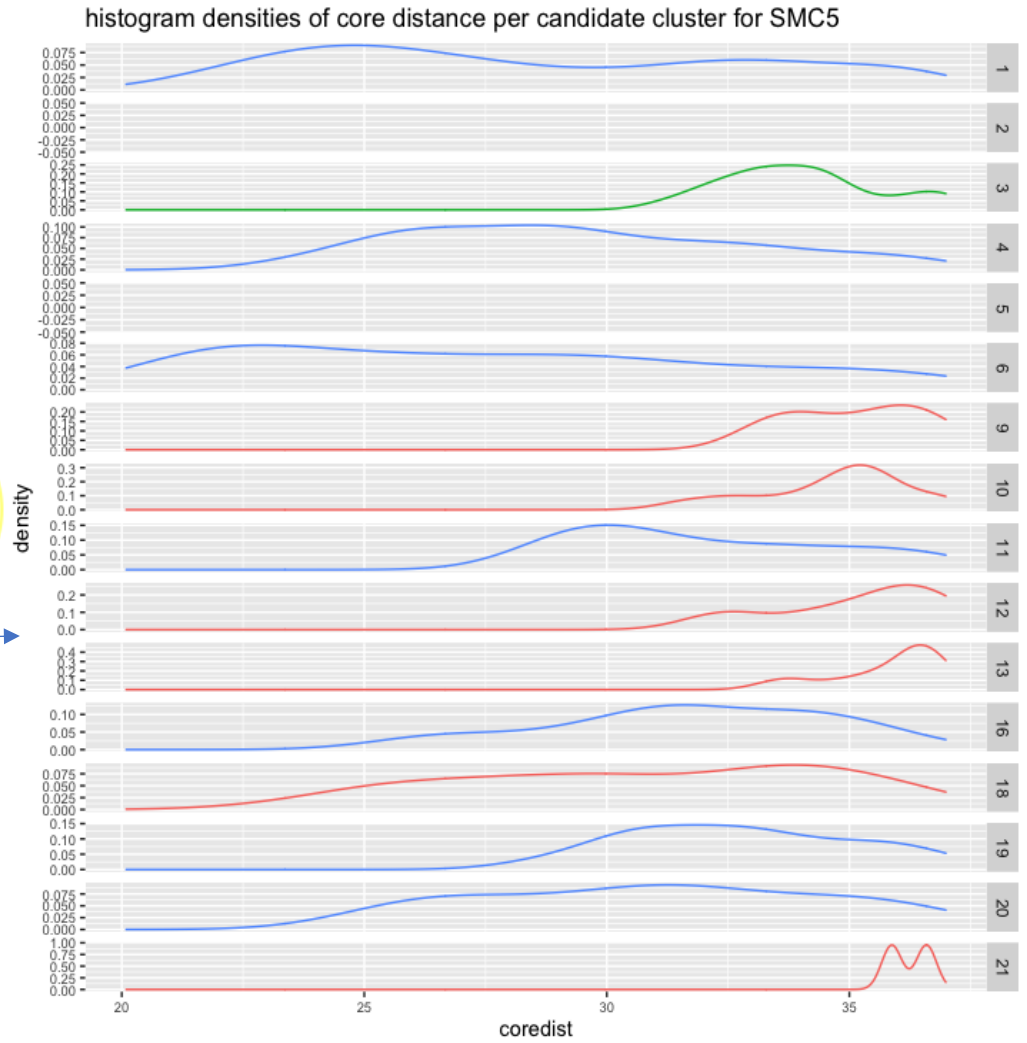


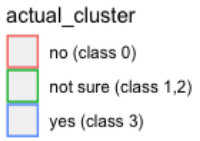
Individual star clusters from IMACS camera (Magellan Telescope) and DBSCAN

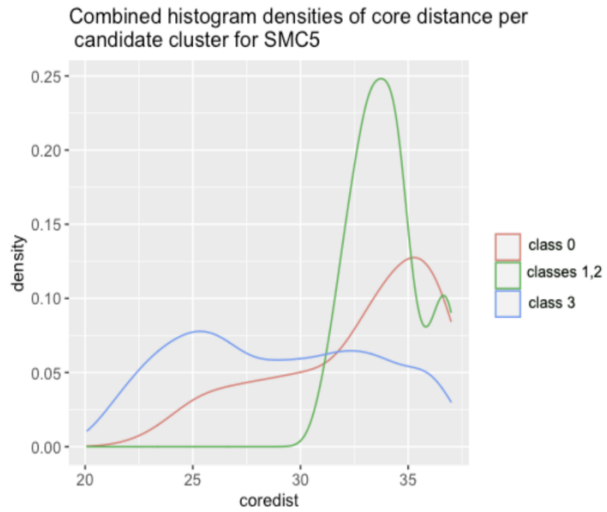
SMC5 approach 2: Density histogram method



Calculating all "core points" (red) distances and putting them in a density histogram plot.



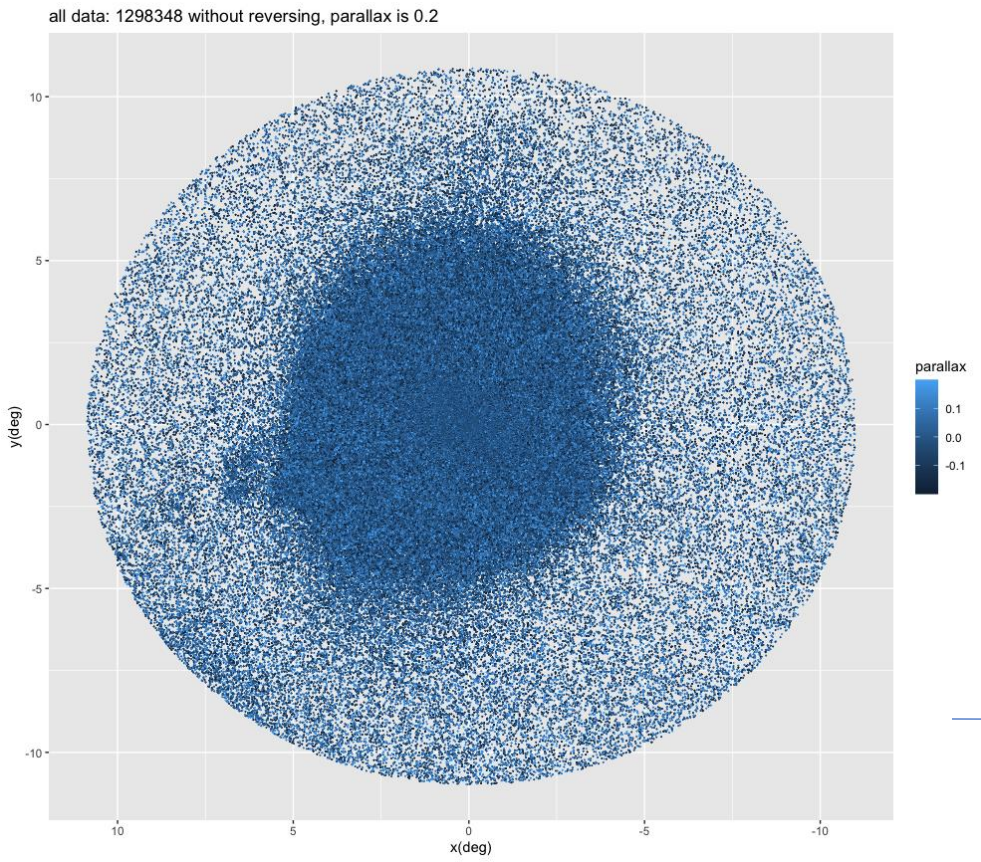




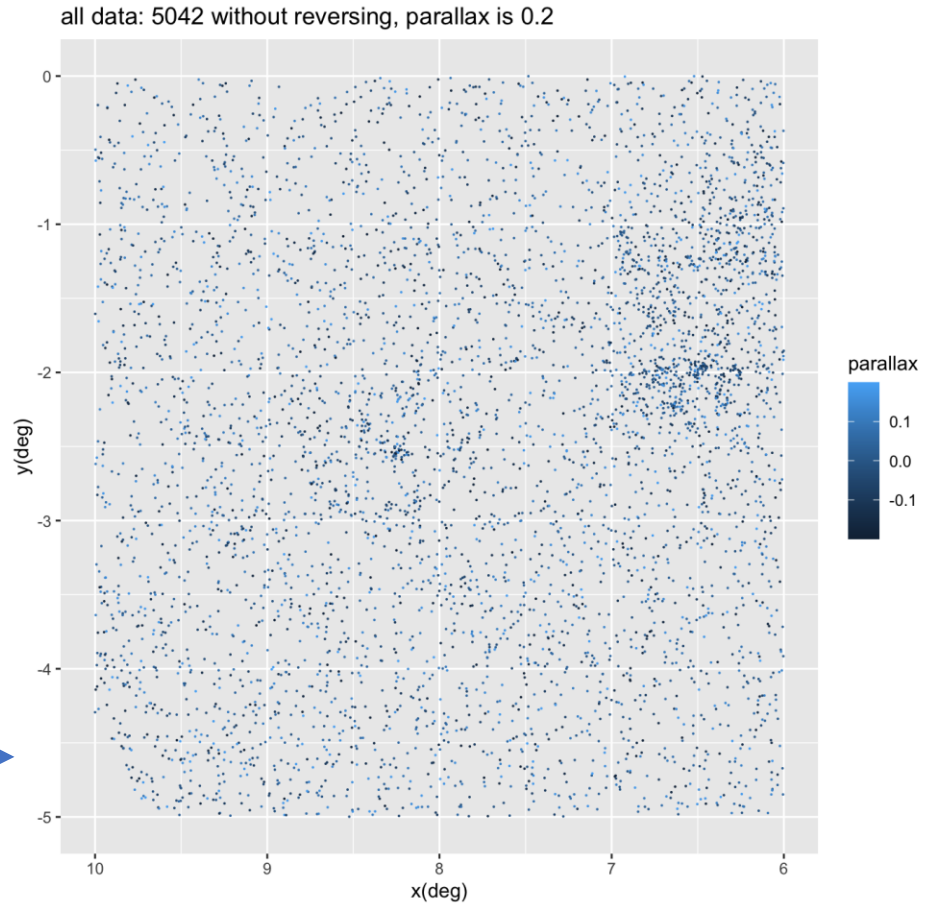
A better way to find the optimal **eps** is to put deliberately a high value. We get some FPs but also 100% of TPs and then distinguish them via their "core distance" density plot. **This method requires only one run.**

MBR: Available data

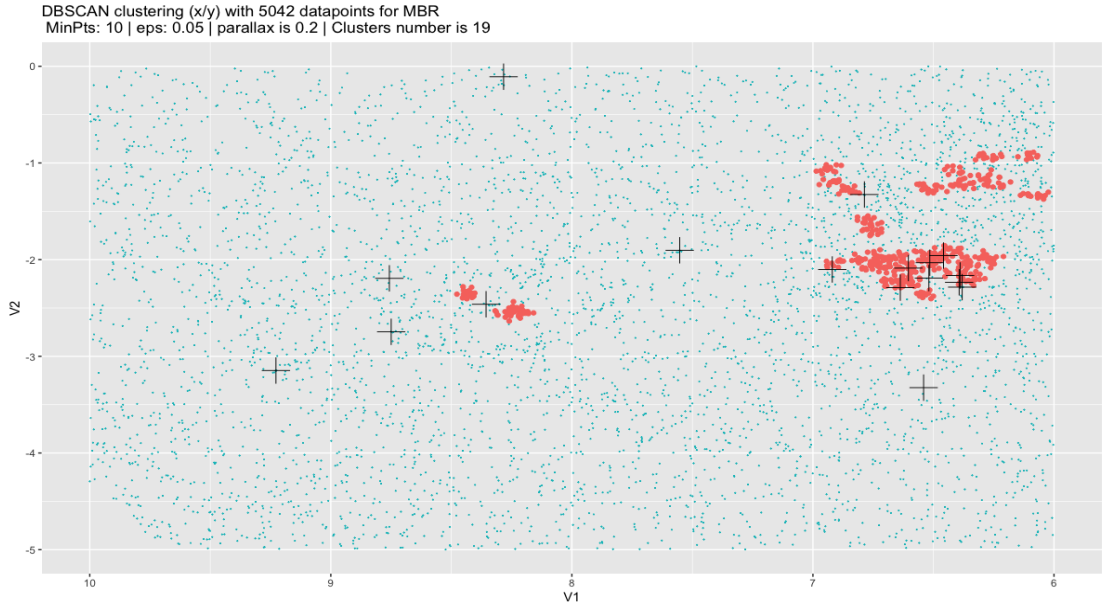
Decontaminating Milky Way stars by using parallax thresholds:

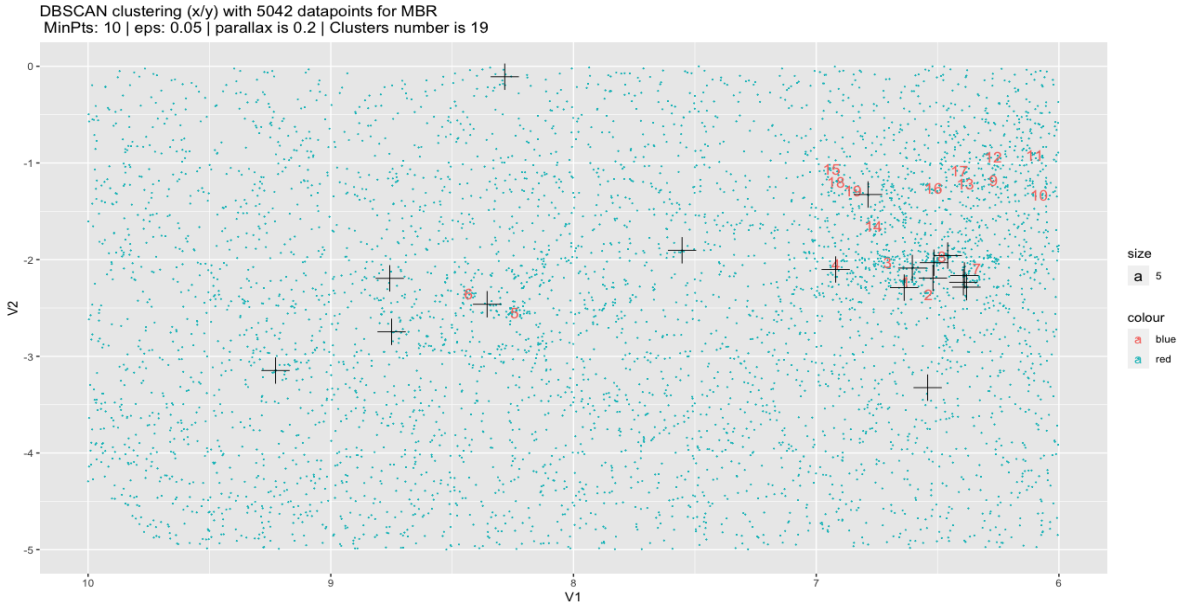


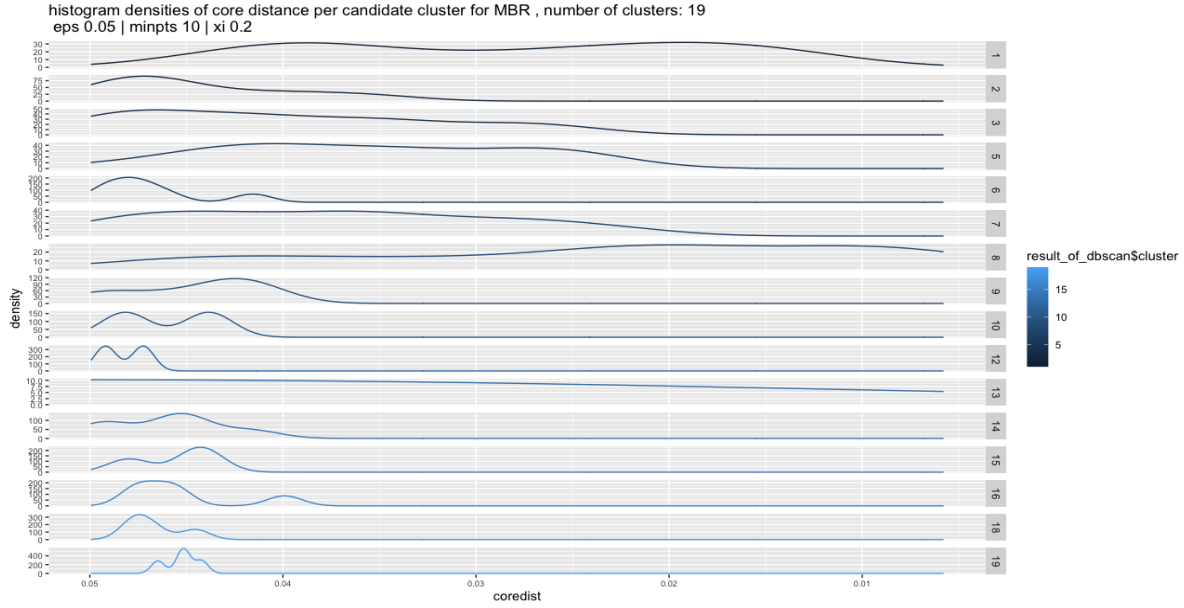
Zooming into MBR



MBR: Using density histogram method







Using the "core distance" method, we put a high eps value. We find the red clusters. Not all of them are actual, we use density histogram method to distinguish. Very clear distinction from the plot shape and AUC.

Previous cluster catalog is denoted with a cross for each cluster center.

(x-axis is inverted in the density plots)

Thank you!

Q & A