PROF-CL. Internship. Technical report

Yuba Amoura

28 juin 2017

Résumé

We wanted, with this work, to study the precision of the profile parameters of the Euclid clusters , we started studying academic clusters in both 2D and 3D coordinates and found a precision of 0.13 *dex*, then we moved to the 3D data of the *MICE* Mock and found large non-cosmological concentrations, which was not the case with *Durham* Mock, then we compared them to the clusters detected by *Farrens*, after matching the two catalogs, to have an idea of the precision that we will reach in Euclid, we found a dispersion of around 0.3 *dex* in the best case. Finally we tried adding a background component and saw that it does not change the results.

Table des matières

1 Introduction

Within *EUCLID* mission, *PROF-CL* is the Processing Function (PF) whose mission is to find the best profile of the detected clusters, ideally re-centralize it and extract the backgrounds/foregrounds from it. This task is important for the improvement of the precision in cluster richness, in addition to possibility of improving the detection of the clusters, being able to adapt the aperture to the size of the cluster.

My work, among this PF, will be to give a first idea of the precision we can reach on the profile parameters, and checking the efficiency of the cluster finder : *Farrens*. To this end, I will try to find the best NFW profile fit for each cluster using a maximum likelihood method to retrieve its scale radius, and eventually its best center and background/foreground. I will do both Academic and Mock data work, starting with the first, in 3D and 2D 3.1 3.2. Then, moving to 3D real space data of *MICE* to study the clusters and the best way to fit them 4.

The largest part of my work was done on *Durham* Mock, and, as for *MICE*, we start studying 3D real space mocks and comparing to the academic case 5.1, then we study the 2D perfect membership case, where we know the exact angular positions of each galaxy in each cluster and compare it to the real space case 5.2, then the most important part will be to compare data from cluster finder *Farrens* to the real space 3D data to get the precision of the detections 6.2.

2 Analytical work

This section aims to explicit all the theoretical calculus I used during my internship, in each step of my work, the formula used in my code can be found here.

2.1 NFW profile in 3D

2.1.1 Normalized functions

The NFW density profile can be written :

$$\nu(r) = \frac{\nu_s}{(r/r_s)(1 + r/r_s)^2} \tag{1}$$

Where r_s is the scale radius, it happens that for the NFW it is also the radius of slope -2 in log-log r_{-2}, and ν_s is the scale density defined as $\nu(r = r_s)$.

The number of particles contained in a sphere of radius r is therefore

$$N(r) = \int_0^r \nu(x) x^2 dx d\Omega \tag{2}$$

Which gives for the NFW :

$$N(r) = \left[\ln(1 + r/r_s) - \frac{(r/r_s)}{(1 + r/r_s)}\right] 4\pi\nu_s r_s^3 \tag{3}$$

We define two normalized quantities :

The normalized number of particles.

$$\widetilde{N}(r/r_s) = \frac{N(r)}{N(r_s)} \tag{4}$$

The normalized number density

$$\widetilde{\nu}(r/r_s) = \frac{\nu(r)}{N(r_s)/4\pi r_s^3} \tag{5}$$

We can re-express these two functions as

$$\widetilde{N}(r/r_s) = \frac{1}{\ln(2) - 1/2} \times \left(\ln(1 + r/r_s) - \frac{r/r_s}{1 + r/r_s}\right) \tag{6}$$

$$\widetilde{\nu}(r/r_s) = \frac{1}{\ln(2) - 1/2} \times \frac{1}{(r/r_s)(1 + r/r_s)^2} \tag{7}$$

Where both functions depend only on r/r_s.

2.1.2 Probability

For a spherically symmetric profile the probability for a particle to be within radii r and r + dr can be written

$$dP(r|r_s) = \frac{r^2\nu(r)dr}{\int_{r_{\min}}^{r_{\max}} r^2\nu(r)dr} \tag{8}$$

The probability density is therefore

$$p(r|r_s) \hat{=} \frac{dP(r|r_s)}{dr} = \frac{r^2\nu(r)}{\int_{r_{\min}}^{r_{\max}} r^2\nu(r)dr} \tag{9}$$

Replacing by the normalized functions defined in the previous section

$$r^2\nu(r) = (r/r_s)^2\widetilde{\nu}(r/a) \times \frac{N(a)}{4\pi a} \tag{10}$$

And the denominator is

$$\int_{r_{\min}}^{r_{\max}} r^2\nu(r)dr = (N(r_{\max}) - N(r_{\min}))/4\pi = \frac{1}{4\pi} \times N(r_s) \times (\widetilde{N}(r_{\max}/r_s) - \widetilde{N}(r_{\min}/r_s)) \tag{11}$$

And finally

$$p(r|r_s) = \frac{(r/r_s)^2\widetilde{\nu}(r/r_s) \times N(r_s)}{4\pi a \times \frac{1}{4\pi} \times N(r_s) \times (\widetilde{N}(r_{\max}/r_s) - \widetilde{N}(r_{\min}/r_s))} \tag{12}$$

$$p(r|r_s) = \frac{(r/r_s)^2\widetilde{\nu}(r/r_s)}{r_s[\widetilde{N}(r_{\max}/r_s) - \widetilde{N}(r_{\min}/r_s)]} \tag{13}$$

2.1.3 Scale optimization principle

Consider we have a cluster, with a *fixed* known center, with a given number of galaxies in it, and we know precisely their position (in 3D), for each one we can calculate the distance from the center. We obtain a set of radii r_i. Let us assume the density profile of this cluster is an NFW.

Then we can write a likelihood function

$$L(r_s) = \prod_{r_i} p(r_i|r_s) \tag{14}$$

We maximize this function, and find the best r_s. We may notice that the function $p(r|r_s)$ depend on the minimum and maximum radii we take into account.

2.2 NFW profile in 2D

2.2.1 Normalized functions

When we observe the sky we see the projection of the clusters along the line of sight, so we need to project our 3D NFW through one axis to model the observations.

We define the normalized number of particles in a circle of radius R as follow :

$$\widetilde{N}(R/R_s) \;=\; \frac{N(R)}{N(R_s)} \tag{15}$$

And the normalized surface density function $\widetilde{\Sigma}(R/R_s)$

$$\widetilde{\Sigma}(R) \;=\; \frac{\Sigma(R)}{N(R_s)/\pi R_s^2} \tag{16}$$

One can prove that the normalized NFW surface density can be written

$$\widetilde{\Sigma}(X) \;=\; \frac{1 \;-\; arcC(1/X)/\sqrt{|X^2 \;-\; 1|}}{(X^2 \;-\; 1)} \;\times\; \frac{1}{2\ln(2) \;-\; 1} \tag{17}$$

Where $X \;=\; R/R_s$, $arcC(x) \;=\; \cos^{-1} h(x)$ for $X \;>\; 1$ and $arcC(x) \;=\; \cos^{-1}(x)$ for $X \;<\; 1$

The number of particles within a disk of radius R can be written

$$\widetilde{N}(X) \;=\; \left(\frac{arcC(1/X)}{\sqrt{|X^2 \;-\; 1|}} \;+\; \ln(X/2)\right) \frac{1}{2\ln(2) \;-\; 1} \tag{18}$$

Where we use the same conventions as for the surface density profile.

2.2.2 Probability

For a circular symmetric surface density profile, the probability for a particle to be between radii R and $R \;+\; dR$ can be written.

$$dP(R|R_s) \;=\; \frac{R\Sigma(R)dR}{\int_{R_{\min}}^{R_{\max}} R\Sigma(R)dR} \tag{19}$$

Playing the same game as in the previous section we obtain the surface probability density

$$p(R|R_s) \;=\; 2(R/R_s)\frac{\widetilde{\Sigma}(R/R_s)}{R_s[\widetilde{N}(R_{\max}/R_s) \;-\; \widetilde{N}(R_{\min}/R_s)]} \tag{20}$$

2.3 NFW 2D with background

When we detect a cluster with a poor redshift resolution, part of the signal comes from galaxies that are in the background and the foregrounds, we may think that considering a profile with an NFW + a background component will fit better the data.

Les us add a background with a constant surface density Σ_{bg}. The probability to find a particle (galaxy) within radii R and $R \;+\; dR$ becomes

$$dP(R|R_s) \;=\; \frac{[\Sigma(R) \;+\; \Sigma_{bg}]RdR}{\int_{R_{\min}}^{R_{\max}} \Sigma(R)RdR \;+\; (R_{\max}^2 \;-\; R_{\min}^2)\pi\Sigma_{bg}} \tag{21}$$

If we define

$$\widetilde{\Sigma}_{bg} \;=\; \frac{\Sigma_{bg}}{N(a)/\pi R_s^2} \tag{22}$$

Then we can re-express the probability density in terms of normalized functions as

$$p(R|R_s, \Sigma_{bg}) = \frac{2(R/R_s)(\widetilde{\Sigma}(R/R_s) + \widetilde{\Sigma}_{bg})}{R_s[\widetilde{N}(R_{\max}) - \widetilde{N}(R_{\min}) + ((R_{\max}/R_s)^2 - (R_{\min}/R_s)^2)\,\Sigma_{bg}]} \tag{23}$$

Then we maximize the associated likelihood to get both R_s and Σ_{bg}.

3 Academic results

3.1 Academic 3D

We start by a test on a perfect NFW profile, we Monte Carlo simulate N particles following a perfect NFW profile with a given scale radius r_s, then, by a maximum likelihood method, try to retrieve it. This is an important step to check the precision we get on a perfect model, and the limitations of the fit method.

Figure 1 shows the dispersion of the error on r_s versus the richness of the cluster. The error is proportional to $1/\sqrt{N}$ corresponding to poissonian error. For $N = 100$ we obtain an error of 0.13 *dex*

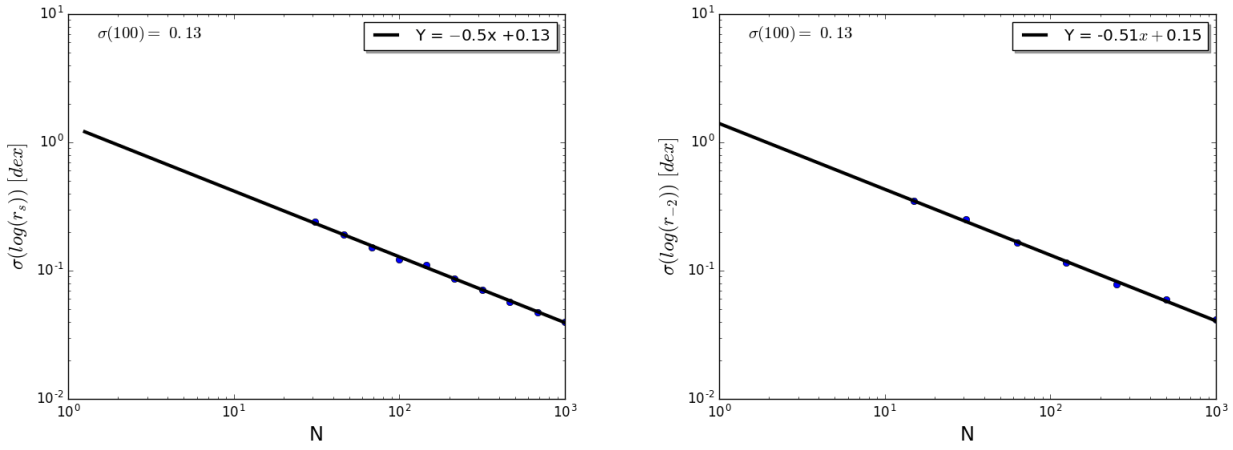

FIGURE 1 – The dispersion in r_s errors versus the number of galaxies in the cluster in log, with a linear fit to show the poissonian errors. Left for NFW in 3D, Right projected NFW in 2D

3.2 Academic 2D

We follow the same procedure with a projected NFW profile, corresponding to the situation we will have with Euclid.

In the projected NFW we obtain again an error of 0.13 *dex*.

Since this error is in the case of perfect NFW clusters, perfectly detected, it is a technical and statistical limitation of the method.

4 *MICE* Mock in 3D

Now we use Mock data clusters as targets. We start with the one from *MICE*, using the following file : $mice\_v2\_100deg2\_halo_complete\_h24\_corrected.fits$ containing the exact positions in euclidian coordinates, the halos masses, the cosmological redshifts and the angular positions.

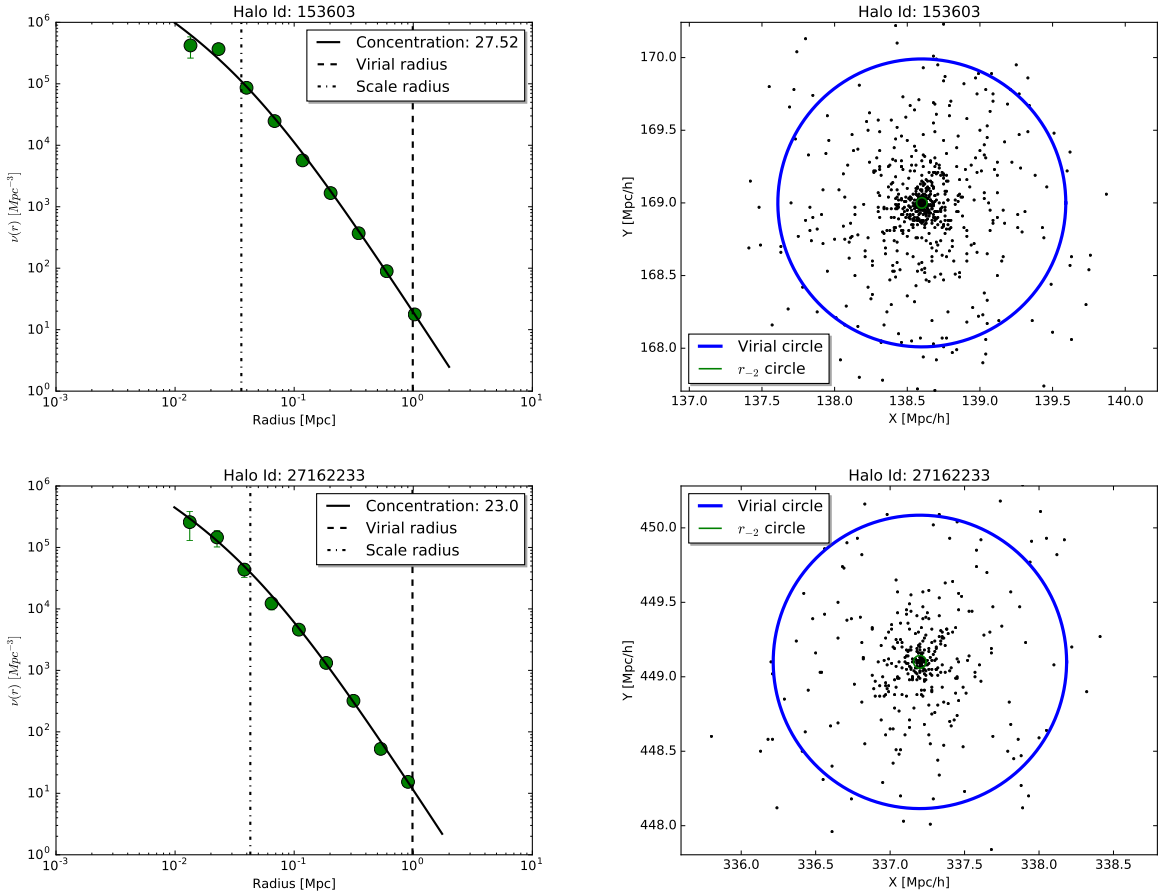

FIGURE 2 – Two examples of fits of the density profiles on the left and a view of the galaxies in X-Y axis with the virial and R_s circles.

We fit the clusters with the 3D method shown in 2.1.3, and show some results rich clusters (N $>$ 100).

We can notice the very large values of the concentrations, which is even more obvious when we do a statistical study of all halos above $M = 10^{13} M_\odot$ with at least 30 galaxies. figure 3 shows the concentration vs richness and log of the mass.

We see that the concentration distribution is Gaussian around $c = 24$. this is way too much for the usual cosmological models, and should be questioned, either our results or the mocks are not in adequacy with what we should get, we will see if we have the same issue for *Durham* clusters.

5 *Durham* mock

5.1 *Durham* 3D

The results of the previous section showed that the HOD Mock *MICE* have boosted, non physical concentrations, we can try to check with the semi analytical Mock from *Durham* whether we obtain more coherent concentrations.

As in the previous section we plot the fitted density profile for some clusters with more than 100 galaxies and a mass larger that $10^{13} M_\odot$ in figure 4.

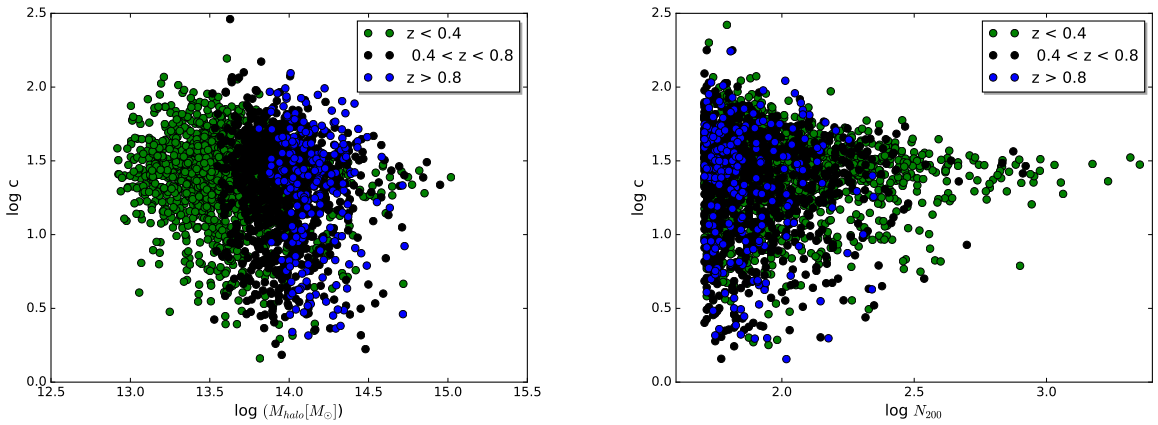

FIGURE 3 – log concentration of *MICE* halos versus, left : their masses, right : their richness, for three redshift bins

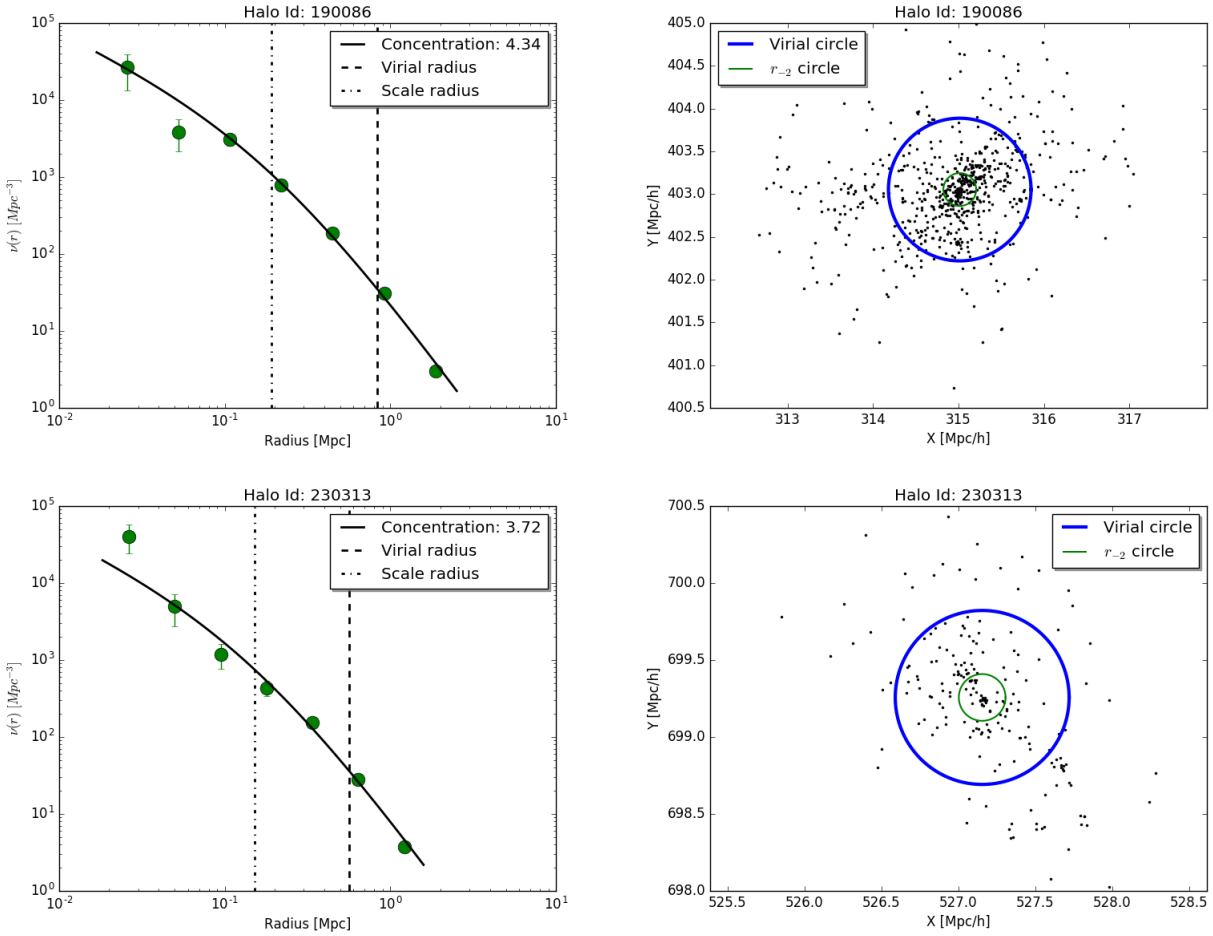

FIGURE 4 – Two examples of fits of the density profiles on the left and a view of the galaxies in X-Y axis with the virial and R_s circles.

The parameters used for the fit are :
— $r_{\min} = r_2$ where r_n is the n^{th} closest galaxy to the center.
— $r_{\max} = 0.8 * max(r_i)$ where $max(r_i)$ is the radius of the farthest galaxy from the center.
— minimization method : *Nelder-Mead* of the Python scipy.otimize.minimize() function.
One can also look at the concentration in function of the halo masses and the richness for three

redshift bands, which gives figure 5.

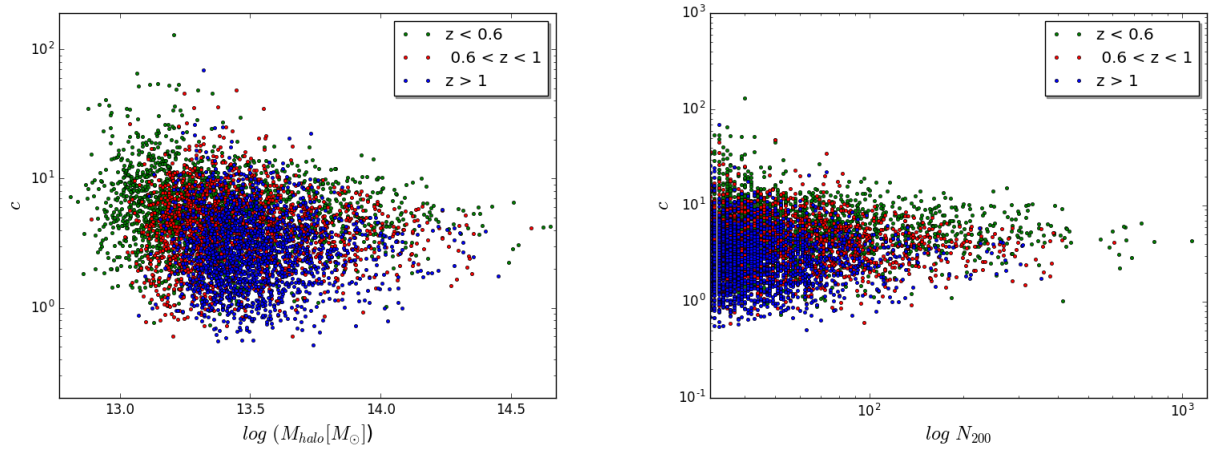

FIGURE 5 – log concentration of *Durham* halos versus, left : their masses, right : their richness. For three redshift bins

We can notice that the concentrations are more coherent with cosmological models which predict $c\ 4 \pm 1$.

5.2 Comparison 2D vs 3D

Now we try to fit the clusters in 2D using the angular positions of the galaxies and compare them to the scales we found with 3D data.

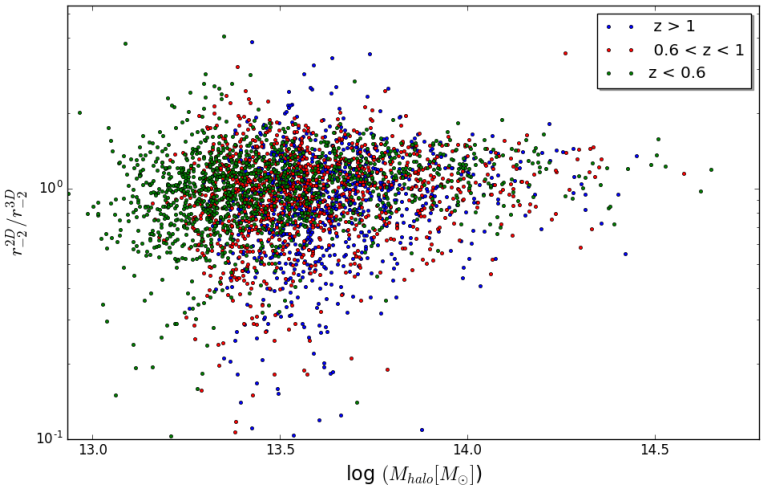

FIGURE 6 – Ratio between the 2D and 3D scale radius versus the cluster mass for three redshift bins

We get biases of +8% for $z < 0.6$, +5% for $0.6 < z < 1$ and 3% for $z > 1$, all due to projection effects. Our guess is that the biases are linked somehow to the way we are getting the scales, the minimization, or part of the algorithm.

Comparing 3D and 2D gives us the error that we have due to the projection of real space clusters to 2D clusters, with a perfect membership and no foregrounds/backgrounds. We should stress on the fact that this is a comparison between perfect clusters, we do not take into account any detection additional error such as backgrounds, foregrounds, cluster identification, incompleteness and impurity. We obtain an error of 0.14 *dex* as we can see in figure 7.

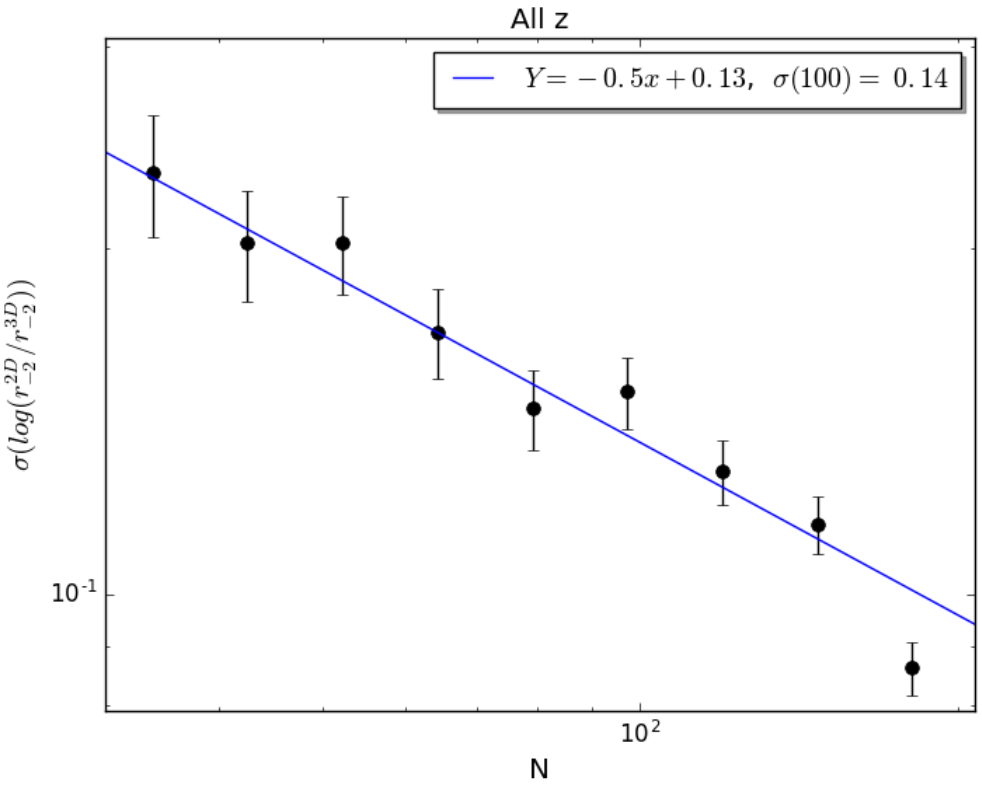

FIGURE 7 – Dispersion of the error on r_{-2}^{2D} with respect to r_{-2}^{3D}

6 Cluster Finder *Farrens*

So far, we have tried to find the best scales on Mocks with real data, we will try now to compare between the scales obtained using the Cluster Finder *Farrens* and the clusters of the *Durham* Mock. To do so, we first need to match the two catalogs.

6.1 Cluster matching

Matching the clusters appears to be a complicated task. The first thing we need to do is matching the galaxy names of the two catalogs of *Durham* mocks, one containing all precise informations on the simulation (3D positions, cluster mass, precise redshift, perfect galaxy membership...etc) *Euclid Deep complete.fits*. The other was the one available for the Cluster Finder Challenges, containing the informations that will provide Euclid. (angular positions, photometric redshifts and some spectrometric redshifts). *Euclid.deepv1.3*

Since we want to match clusters based on the second catalog, with the same galaxy names, with the first catalog, this first step is mandatory. We use the angular matching function of astropy [1], and we store a numpy array with the indexes of the galaxies in the *Euclid Deep complete.fits* corresponding to the ones in *Euclid.deepv1.3*. its length is the number of galaxies in *Euclid.deepv1.3* 9082168, stored in a txt file *idxmatch.txt*.

Now we have to face another issue, how to associate *Farrens* and *Durham* clusters, a quick view of *Farrens* tels us that it contains 556196 detected clusters. As a starting point we do not use an SNR cut, nor take into account the membership probability of each galaxy. We assume a perfect cluster membership catalog to avoid complicated issues with the matching with *Durham* clusters. On the other side, we make a cut on the halo mass at $> 10^{12} M_{\odot}$ motivated by the fact that lower

1. After correcting with a shift on the angular coordinates between the two catalogs

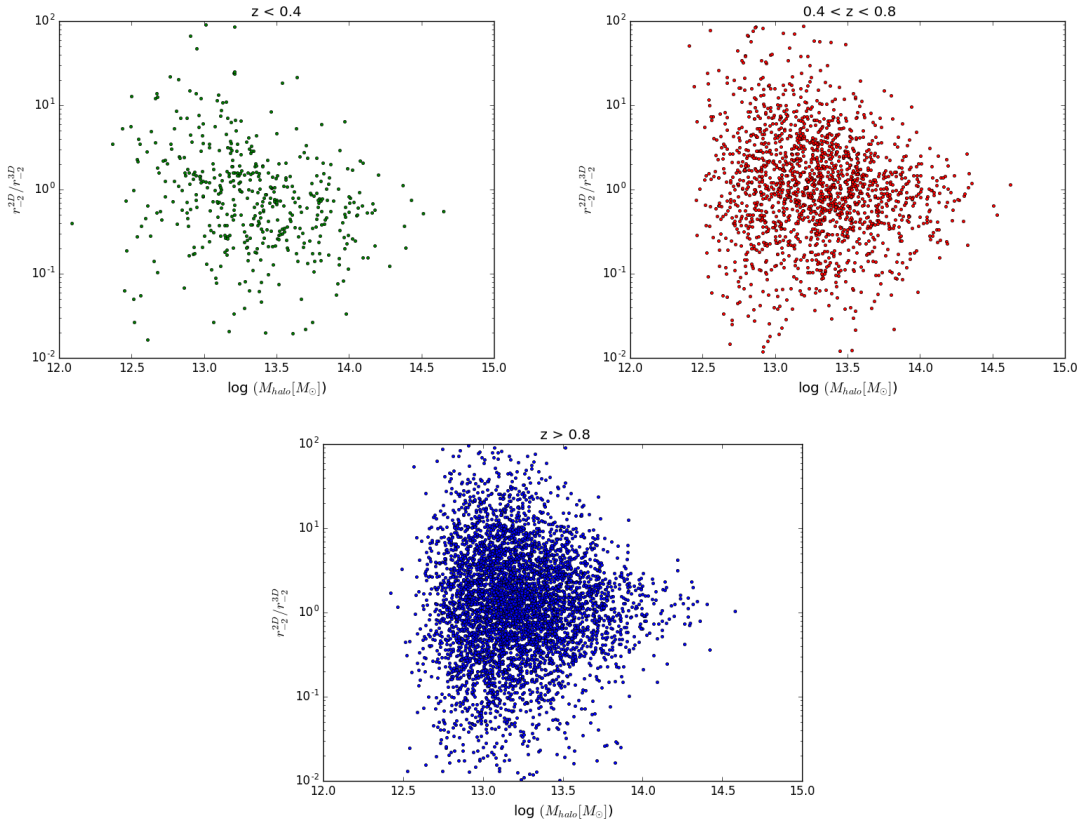

FIGURE 8 – Ratio between the 2D Farrens and 3D *Durham* scale radius in function of the cluster mass for three redshift bins.

mass clusters will not be detectable. We get 395800 clusters, less than *Farrens* ones, we will get even less if we restrict ourself to a given *richness* limit. Then the idea would be to find, for each *Farrens* Cluster, the best counterpart from *Durham*. The simplest criteria would be by number of galaxies shared, which present the drawback of giving more weight to rich clusters, the second simplest is to normalize the number of common galaxies by the size of the considered clusters, and this is the strategy we will adopt.

Another technical issue is the fact that we cannot simply calculate for each *Farrens* cluster, the number of common galaxies with each *Durham* one, we will have to loop over all of them, and it takes too much time. One way to solve this problem is select the best candidate in two steps, where the first one can be done vectorially, we could think of selecting first the clusters close in redshift, but the poor resolution makes it difficult, therefore the obvious criteria would be the angular positions, we select first the *Durham* clusters where the center is closer than the farthest galaxy of the *Farrens* cluster [2]. Then, for each selected cluster, we calculate the ratio between the shared and the total galaxies of the two clusters, and we select the larger. Without any richness restriction we obtain 339057 matched clusters and 56743 without candidate, however, after checking the data we can see that most of them are poor matches, sharing 1 galaxy, in fact, most *Durham* clusters have few galaxies, in fact, there are only 85000 *Durham* clusters with more than 5 galaxies and only 6000 with more than 30. And if we apply this cut, before making the match we get around 1550 good matches with $N > 30$ cut and around 1650 with $N > 5$. We store them in a numpy array of the same length where we put for each cluster, the *Farrens* index and all its relevant informations (*Durham* galaxy and halo id, halo mass, cosmological redshift, 3D and angular galaxy positions and radii/separation).

All this steps are explicitly written and explained in the Python program *matching.py*.

2. other limits such as the virial radius are possible, and maybe better

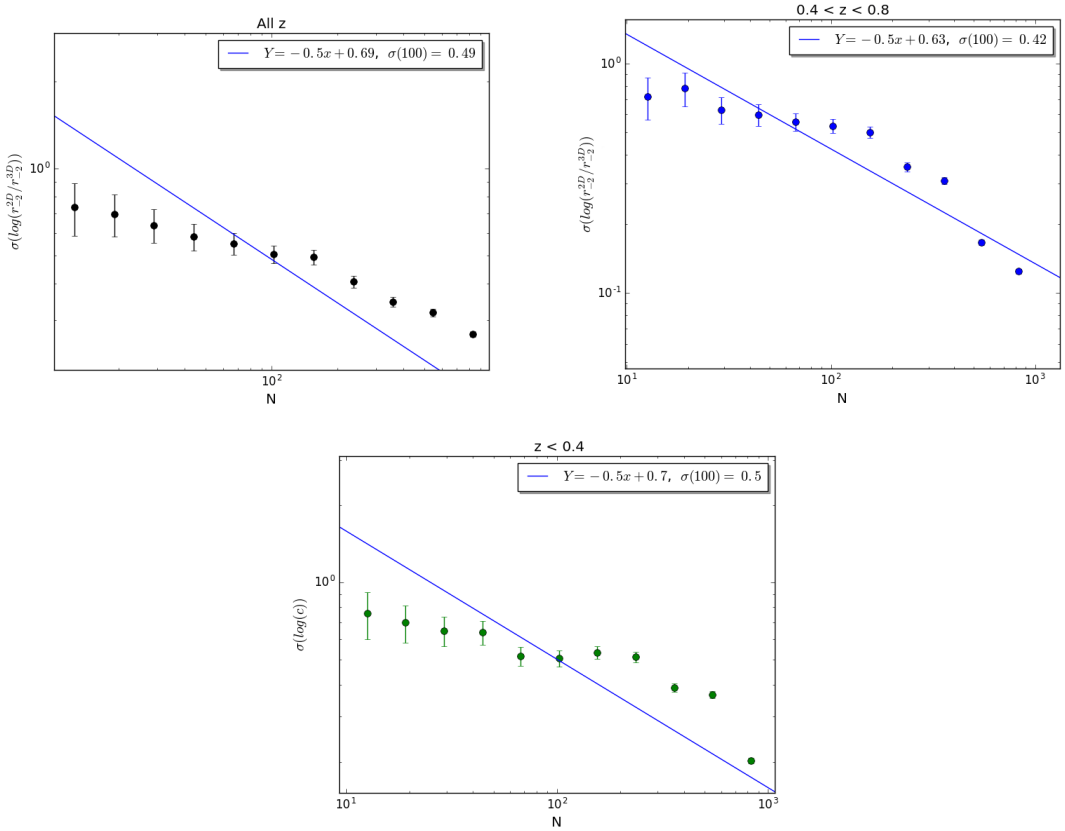

FIGURE 9 – Dispersion of the error on r_{-2}^{2D} with respect to r_{-2}^{3D}, the case where there is no richness restriction.

6.2 *Farrens* vs *Durham*3D

We have now everything to reach the initial goal, checking the precision of *Farrens* clusters with respect to the true Durham clusters, we have a matching between the two, and a method to compare the density profiles.

We start first by the match we did considering all *Durham* clusters richness, we obtain biases of 206%, 200% and 197% for respectively $z < 0.4$, $0.4 < z < 0.8$ and $z > 0.8$, in addition to the fact that, one third of scale radii obtained with *Farrens* clusters are 100 times larger or smaller than the 3D *Durham* scale radii, and are not taken into account for the calculation of the biases and the errors.

We complete these results by figures 8 and 9, representing the ratio of the two scale radii in function of the mass, for the three redshift bins and the dispersion of the error on r_{-2}^{far}. We get large errors of 0.42 dex, 0.5 dex and 0.49 dex for respectively $0.4 < z < 0.8$, $z < 0.4$ and all clusters, and are badly fitted by a poissonian behavior.

Now, if we use a matched catalog restricted to *Durham* clusters richer than 5 and 30 galaxies, we obtain the figures 10 and 11 respectively, for 4 different redshift bins. And get biases of -5%, -9% and -9% (10) for respectively $z < 0.6$, $0.6 < z < 1.2$ and $z > 1.2$. While it is -7%, -6% and -6% for the second case (11, with much better biases and very few outliers, the precision is also improved, particularly for $z > 1.2$ galaxies, where the dispersion follows a poissonian behavior with 0.28 *dex* for $N = 100$, much better than the previous results.

6.3 *Farrens* with background/foreground

The galaxy assignation is made even more difficult by the poor resolution in redshift that we have for most of them, we will have a non negligible component of galaxies that are in the background/foreground for each cluster, we could think of adding up a background component to our fit as explained in section 2.3.

However our results show that adding a background does not change our results in any way, if we fit a constant background, we obtain a very low value close to 0, and the scale radius obtained obtained with background is exactly the same without.

7 Conclusion

During this work, we had to write a first code allowing us to get an idea of the precision on the scales of the EUCLID clusters and the efficiency of the detection algorithms. We obtained that the minimum dispersion that can be reached, for an academic perfect NFW profile in 2D is 0.13 dex, this is the intrinsic limit of our method. Then, when we moved to *MICE* clusters, we figured out that they cannot be physical and should not be used in their current state, while the *Durham* clusters appear to be in adequacy with the cosmological models.

We showed, after that, that the best precision we can get on 2D perfect membership clusters, in comparison to 3D is 0.14 dex, this is in the case where we know exactly the membership of each cluster, without background and foreground. This error is due to projection effects and will always be there since we will have 2D data.

Now, the most important result we got is the comparison *Farrens* vs *Durham* 3D. In one side we had detected clusters as we will have with Euclid, and in the other the true 3D positions,

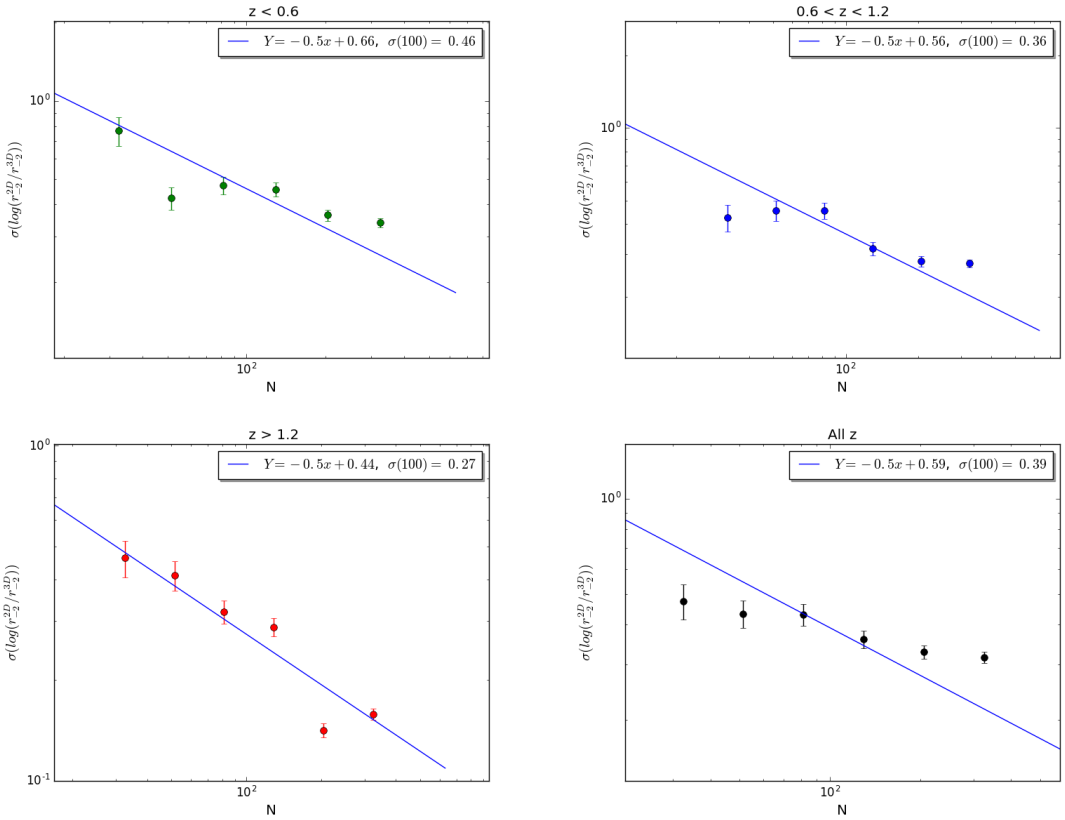

FIGURE 10 – Dispersion of the error on r_{-2}^{2D} with respect to r_{-2}^{3D} for $N > 5$ richness limit for *Durham* clusters.

unfortunately the results are not independent on how we match the catalogs, either in the method in itself or in the parameters of the method. This is the main restriction of such a study, knowing that, every quantified precision we can give will be subject to this issue, assuming we detect only rich ($N > 30$) clusters or all of them makes a difference, and probably that, doing another matching method will give completely different results. For instance, we obtain around 0.5 dex of error if we take into account all possible matches and around 0.35 if we take into account only $N > 30$ clusters. In addition to the matching method, outliers and bad fits impact also the results, by increasing the biases, not taking them into account changes the result, which implies to use an arbitrary cut... which impacts again the results. Finally, what we can say, despite all the previous remarks, is that we can reach a precision of around $0.35dex$ for 1600 matches of rich clusters and a small bias of around -10% if we exclude bad fits. Followed by the important remark that this results are just hints and depend on many different parameters.

And finally, maybe the most interesting result, is that adding a background does not improve, or change the maximum likelihood fit. Again, this is worth doing the same study with another program, model or parameters. In conclusion, in each step of our study we made a model/method/parameter choice which can, sometimes, be discussed and maybe worth changing.

We started by assuming a spherically symmetric NFW profile to fit the clusters, we could try non spherically symmetric models, with prolatness or other models. In the maximum likelihood algorithm we used, most of the time, a *Nelder-Mead* minimization from Python Scipy library, which appeared to be more efficient than any other minimization method in that library, we could get slightly different results with other minimization methods, this adds up to the other minimization parameters such as maximum and minimum radii to take into account in the fit, we tried many criteria to improve the results, probably the parameter that we changed the most to get to the conclusion that the most efficient choice is to take into account all the galaxies but the closest

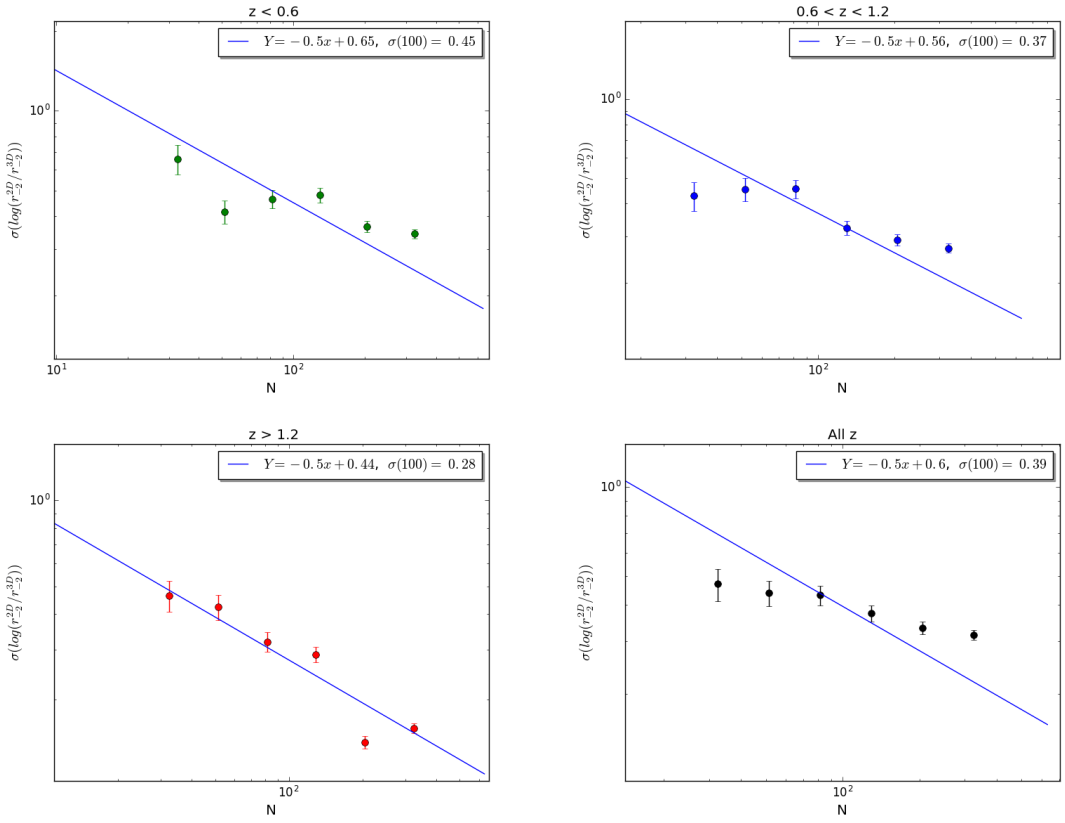

FIGURE 11 – Dispersion of the error on r_{-2}^{2D} with respect to r_{-2}^{3D} for $N > 30$ richness limit for *Durham* clusters.

to the center. This is also worth questioning it again with another fit method. Another way to improve the fit would be to try free center fits, which takes more time. And finally, we matched the *Farrens* and *Durham* catalogs with the simplest way possible, it is difficult to judge how efficient it was, the question we should ask first is how many matches we expect to have in the best case, because we found a small subset of all clusters matched, but if we take into account only rich *Durham* clusters ($N > 30$) we could match 25% of them.

The next steps would be ideally :

— Try prolate cluster models and other than NFW.
— Use other minimization methods and try to find an objective criteria for which method and parameters to chose.
— Try free center and adding background to the fits.
— Do the study with the *Bellegamba* catalog and check the differences with *Farrens*.
— Change the catalog matching.