UNIVERSITÀ DEGLI STUDI DI TRIESTE

APPLIED DATA SCIENCE & ARTIFICIAL INTELLIGENC

High-Performance Unsupervised Machine-Learning in Numerical Simulations and Satellite imaging

Francesco Tomba
PhD student in Applied Data Science and Artificial Intelligence
University of Trieste

francesco.tomba@phd.units.it

Prof. Rodriguez Garcia Alejandro
Supervisor
UniTS

Prof. Tornatore Luca
Co-supervisor
INAF-OATS

Outline

- Introduction to the *Advanced Density Peak* (ADP) algorithm
- *ADP as an Halo Finder*: experimental results and technical challenges. From Python to MPI
- Using density based clustering for image segmentation

Advanced Density Peak (ADP)

Advanced Density Peak is a non-parametric density-based clustering technique, introduced by d'Errico, Facco, Laio, and Rodriguez in 2021 [1].

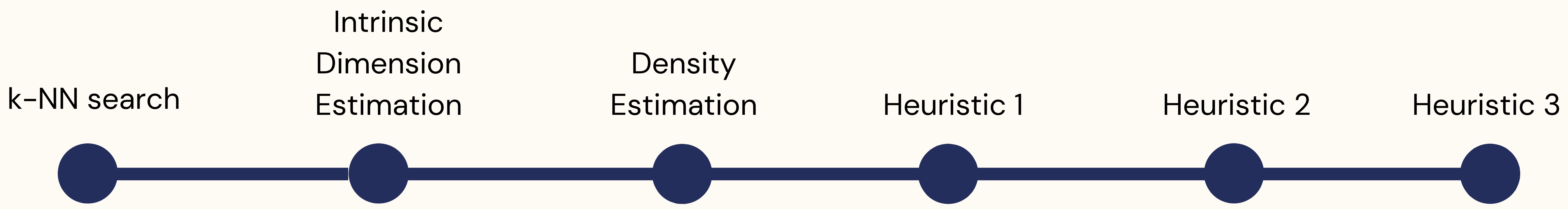
Density-based clustering interprets a dataset as the result of the sampling of a particular yet unknown distribution. Clusters are peaks in the density landscape

[1] M. d'Errico, E. Facco, A. Laio, and A. Rodriguez, "Automatic topography of high-dimensional data sets by non-parametric density peak clustering," *Information Sciences, vol. 560, pp. 476–492, 2021*

Advanced Density Peak (ADP)

Describes a procedure to automatically find peaks, valleys, and saddle points of the probability density landscape and provides a method to then group data points around maxima of the density

Data processing pipeline:



[1] M. d'Errico, E. Facco, A. Laio, and A. Rodriguez, "Automatic topography of high-dimensional data sets by non-parametric density peak clustering," *Information Sciences, vol. 560, pp. 476–492, 2021*

ADP Heuristics

H1: Cluster centers

The first heuristic aims to find cluster centers, namely maxima of the density. We introduce the quantity g

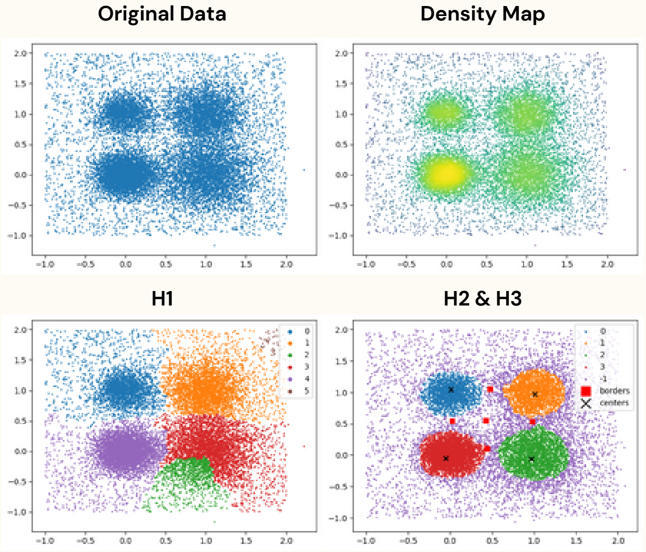
A point is a maximum if:

- it is the point with maximum g in its neighborhood
- it is not in the neighborhood of a point with higher g

Definition of the quantity g, and the error on the density estimate

$$g_i = \log(\rho_i) - \varepsilon_i$$

$$\varepsilon_i = \frac{\partial \log \rho}{\partial \rho}\epsilon_i = \frac{1}{\sqrt{k^*}}$$



H2: Density saddle points

A border point i between clusters c and c' has the following properties:

- in its neighborhood, there is a point j belonging to c'
- i is the closest neighbor of j belonging to c

A saddle point between two clusters is the border point that has the highest value of density.

H3: Cluster merging

The third heuristic tests the significance of the density peaks and merges clusters that can be considered as statistical fluctuations of higher peaks. A cluster c is merged to c' if the following condition holds.

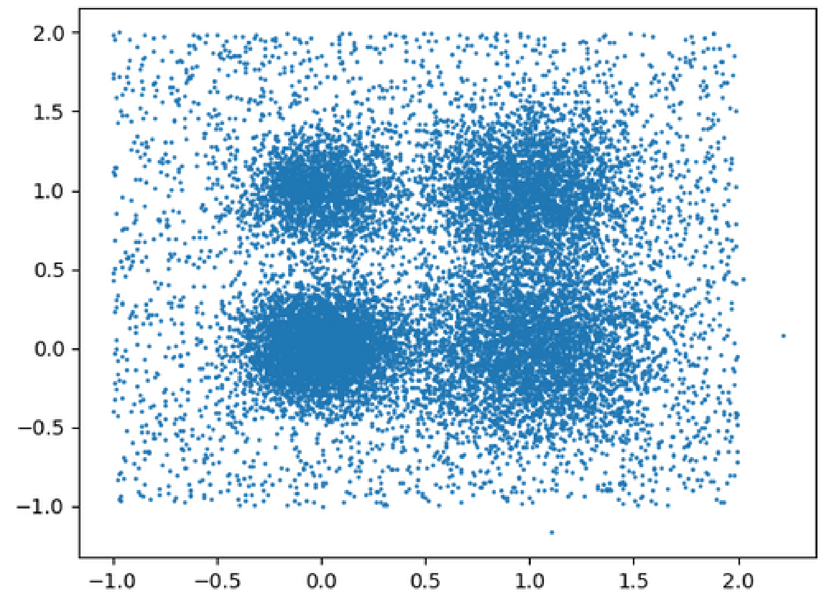
Cluster merging criterion

$$\log(\rho_c) - \log(\rho_{cc'}) < Z \cdot (\varepsilon_c + \varepsilon_{cc'})$$

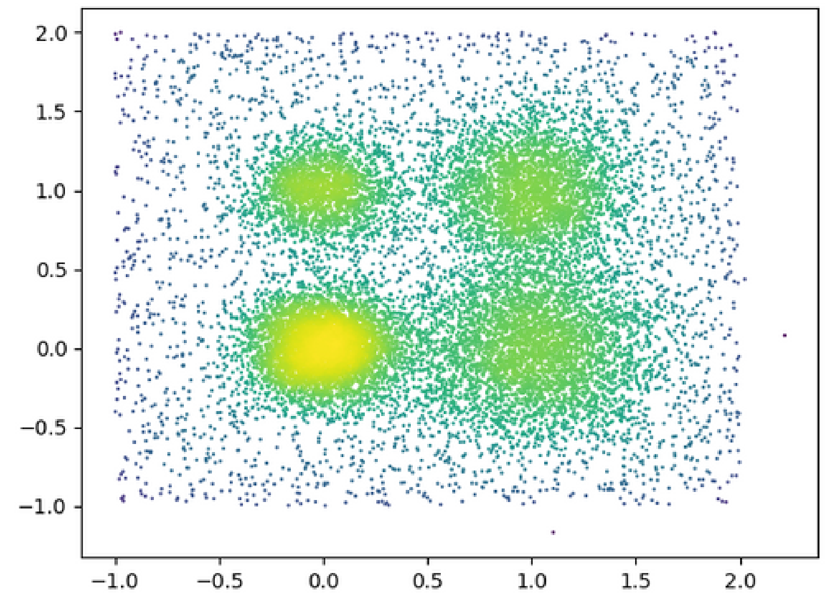
Where: $\rho_c \, \varepsilon_c$ are the density value and associated error at the peak and $\rho_{cc'} \, \varepsilon_{cc'}$ are the ones on the border

ADP Heuristics

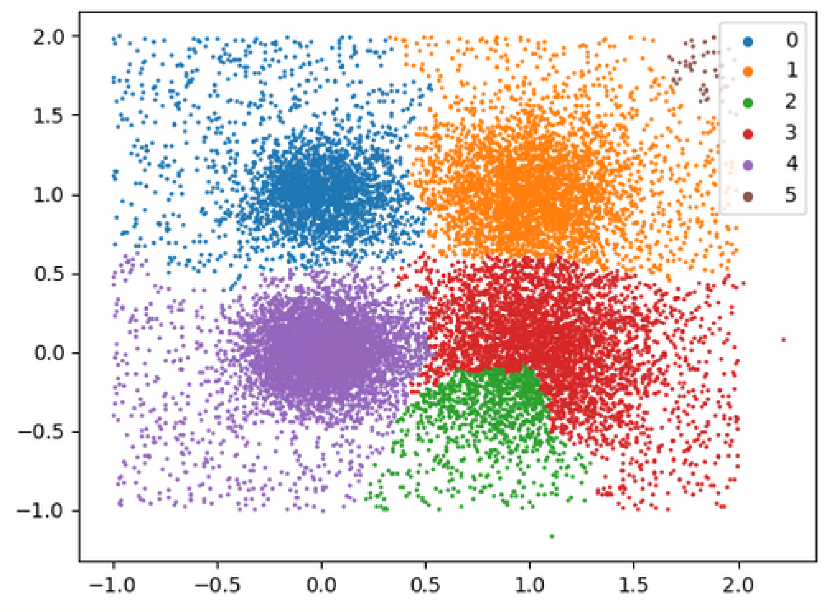
Original Data



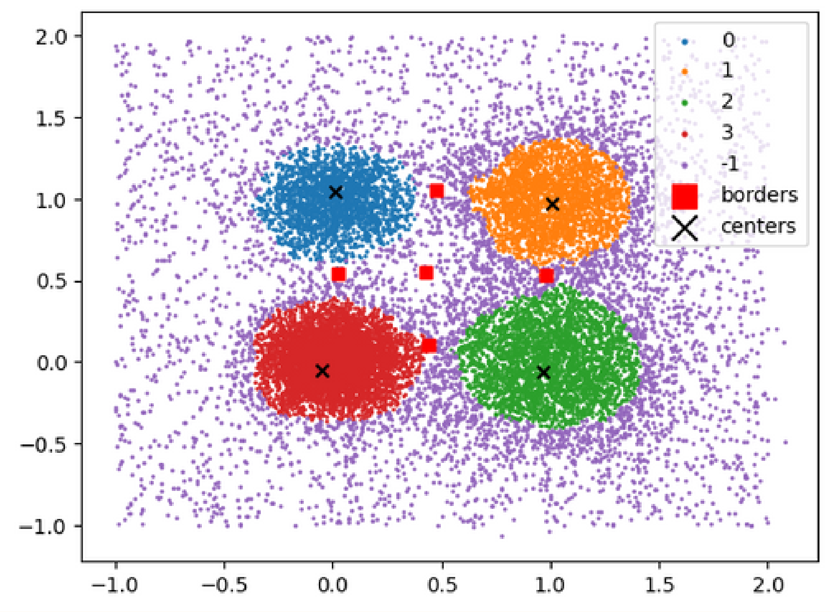
Density Map

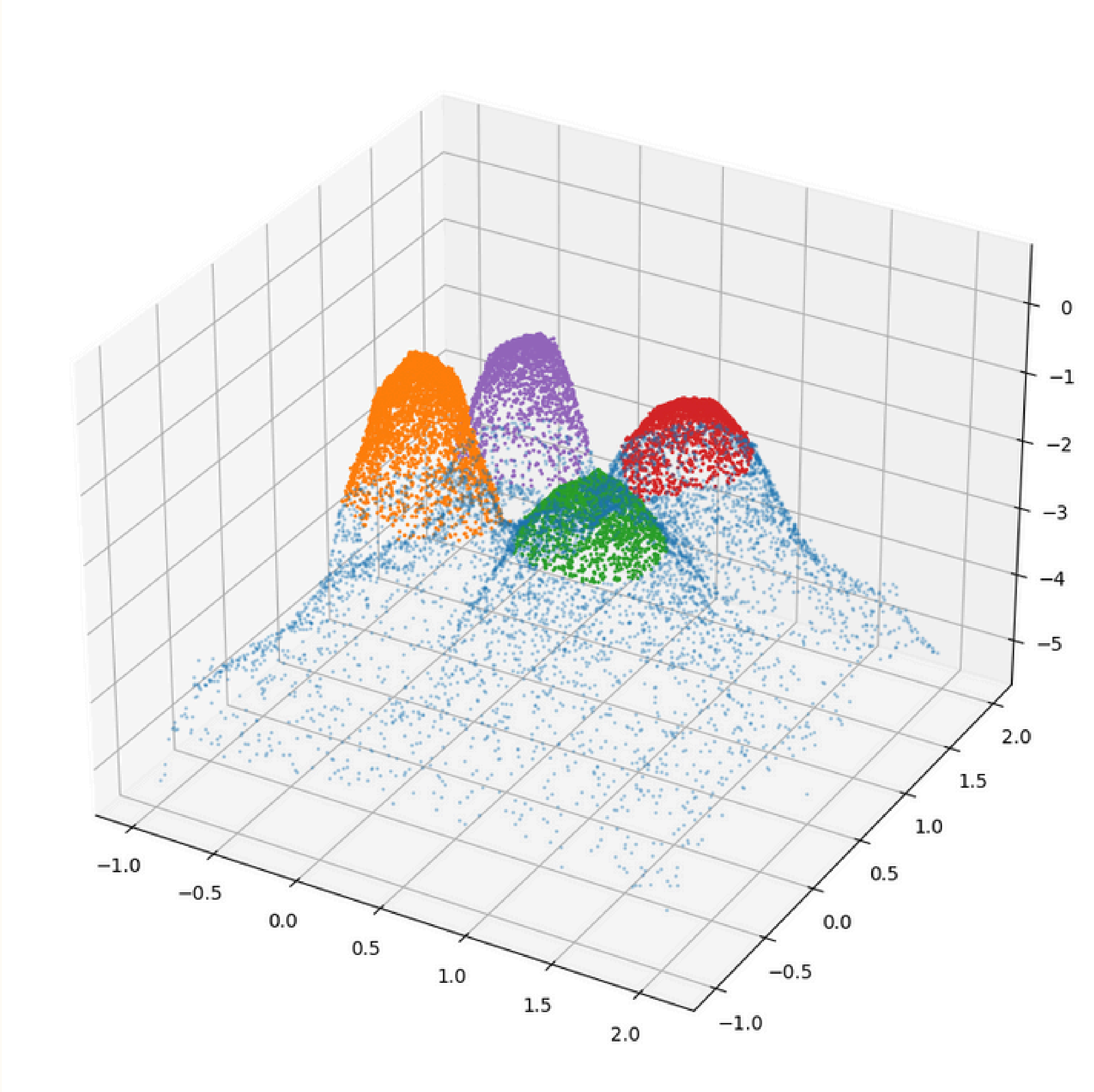


H1



H2 & H3





3D projection of the dataset in which z direction represents the density value

ADP as an Halo Finder

data sets description

ADP has been applied in finding substructures within *Friend of Friends (FoF)* groups, by applying substructure finders. These are namely galaxies and galaxy clusters.

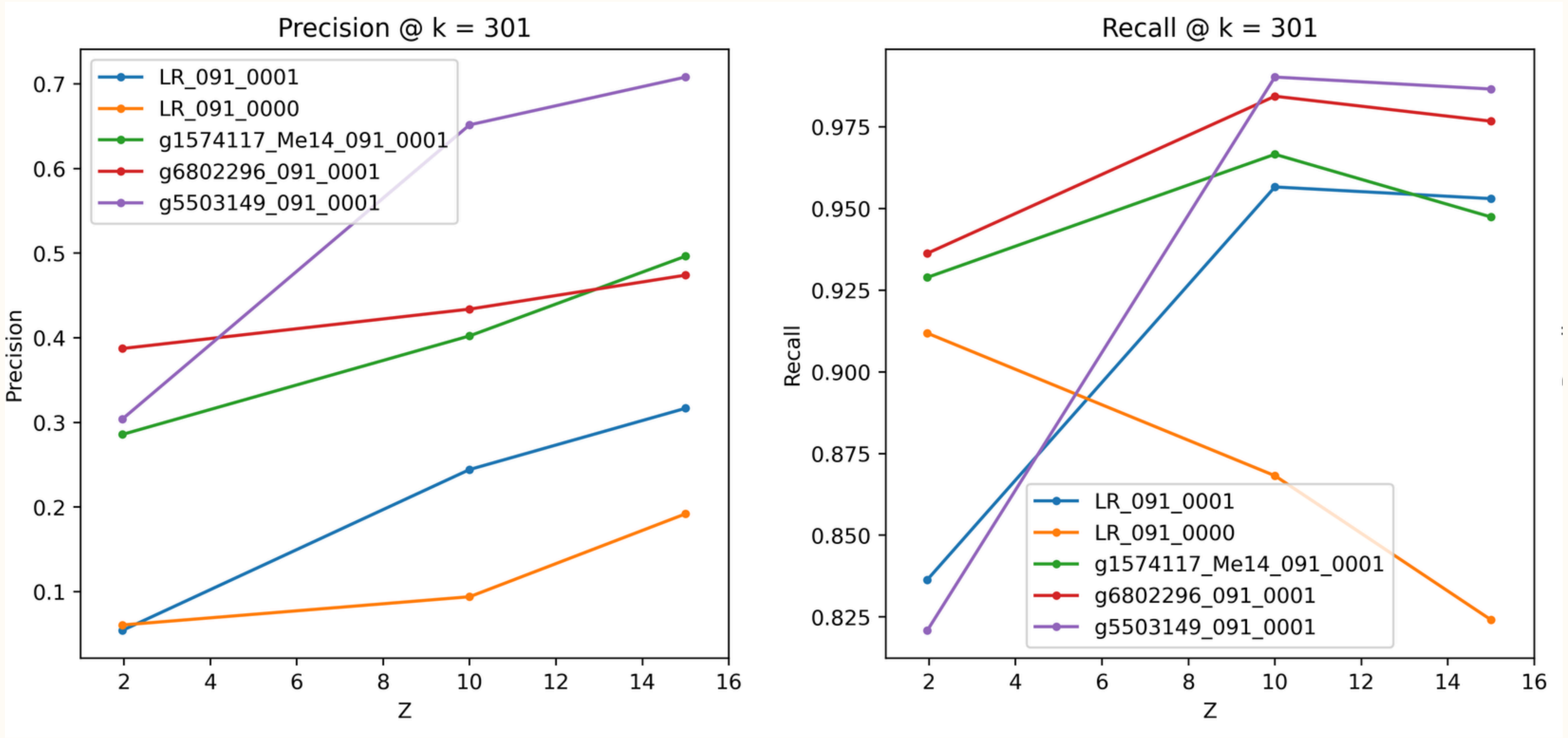
Particles are subdivided into four types namely gas, dark matter, stars, and black holes.

Each data point has 5 dimensions: the 3 spatial coordinates, x, y, z, and the kinetic and potential energy values.

Name	Number of points
LR_091_0001	1 842 842
LR_091_0000	3 673 512
g5503149_091_0001	5 846 506
g6802296_091_0001	6 068 944
g1574117_Me14_091_0001	6 613 708

The first two are the two largest FoF obj found in a low resolution of a massive galaxy cluster ($10^{15} M\odot$) The others are the second largest FoF groups from the simulation at higher resolution of smaller obj.

ADP as an Halo finder



Results compared against SUBFIND

Assignation by SUFIND as ground truth

Changing **k** : **little** or no **impact**
Changing **Z** : **huge** impact.

Recall values > precision ones:
false positives ~ true positives.

ADP splits up bigger structures

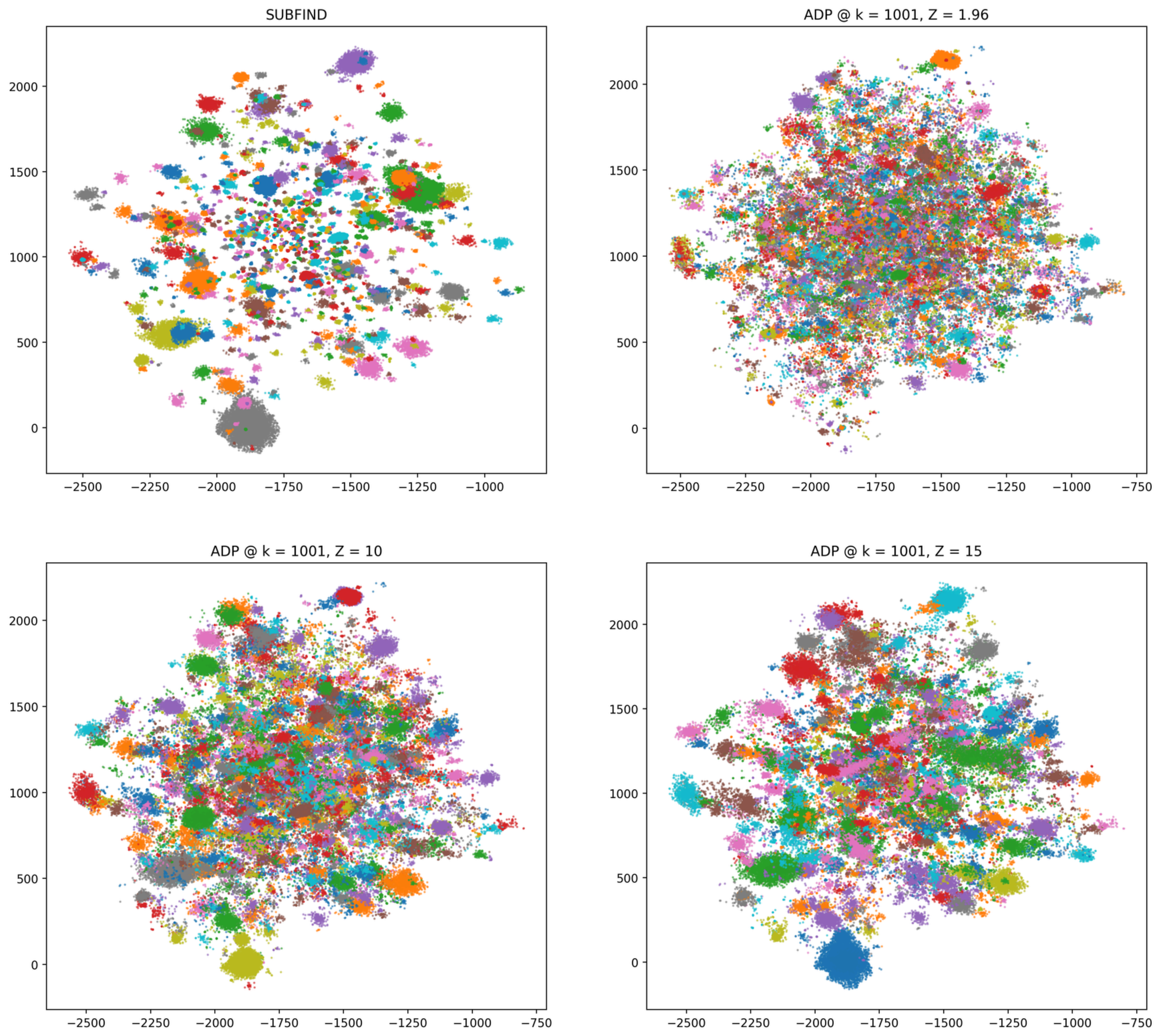
NOTE:
Positive and negative signals are computed by comparing the assignment of each couple of points

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN}$$

Experimental Results

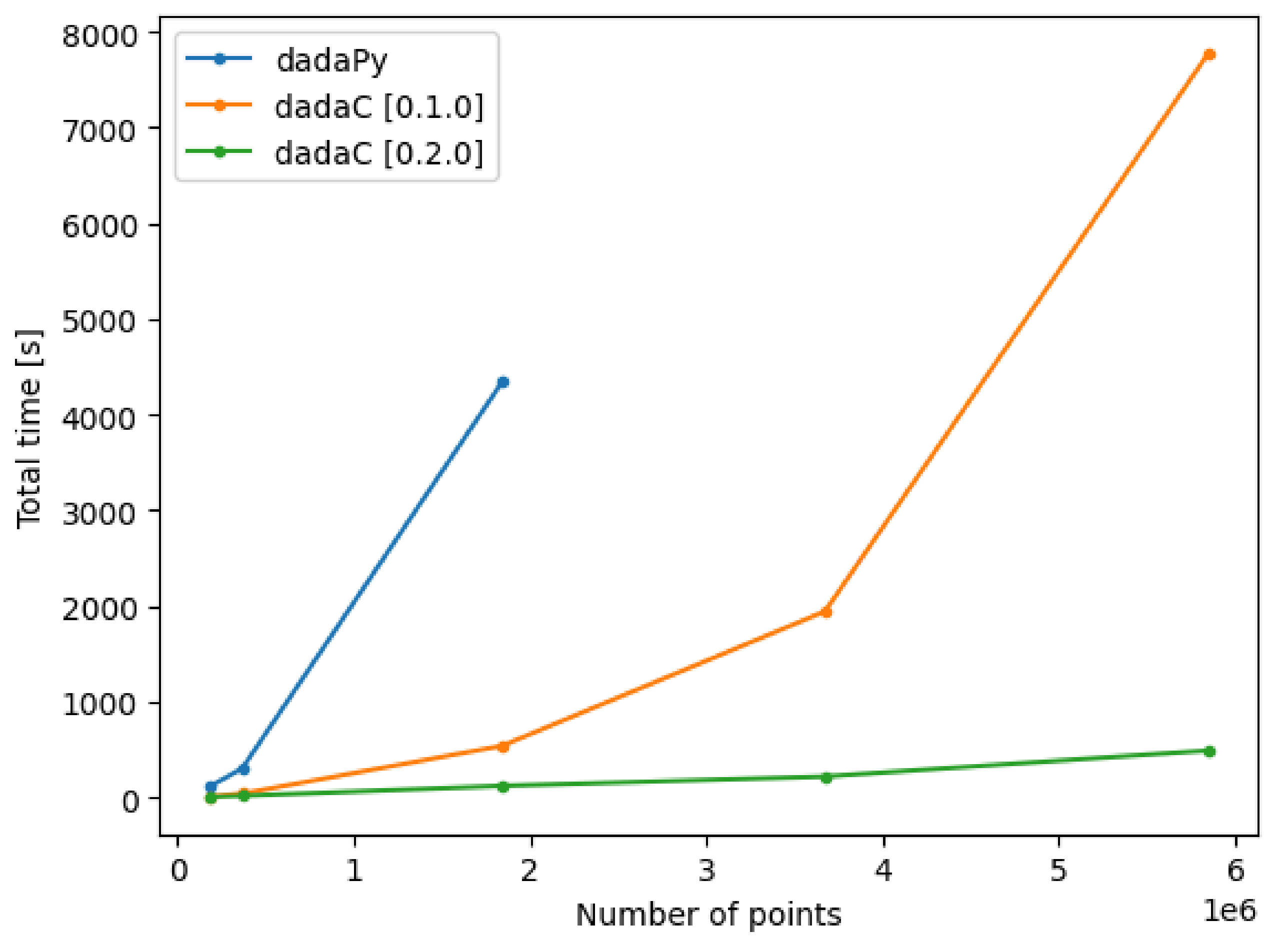
ADP parameters impact

xy projection of the FoF group
`g6802296_091_0001`



from dadaPy to dadaC

C implementation of ADP



LR_091_0001 [1.8M points]			
	dadaPy	**dadaC [0.1.0]**	**dadaC [0.2.0]**
k-NN search	879.7 s	90.4 s	97.8 s
ID	—	3.6 s	6.9 s
Density	68.42 s	2.9 s	3.2 s
H1	2496.3 s	332.9 s	1.7 s
H2	10.56 s	1.3 s	1.1 s
H3	943.8 s	107.3 s	1.2 s
Total	4342.6 s	539.6 s	120.6 s

dadaPy is the original implementation in Cython (from the official package).

dadaC [0.2.0] is the final version with major algorithmic optimizations developed by us. Parallelism leveraged using OpenMP.

dadaC output in all versions was binary equal to the one produced by dadaPy

More than a factor 40 of speed up has been obtained w.r.t the Cython implementation

The main issue is memory imprint, storing knn search results require O(kN) memory

dadaC repository: https://github.com/lykos98/dadaC
dadaPy repository: https://github.com/sissa-data-science/DADApy

dadaC in distributed memory

The quest for nearest neighbors

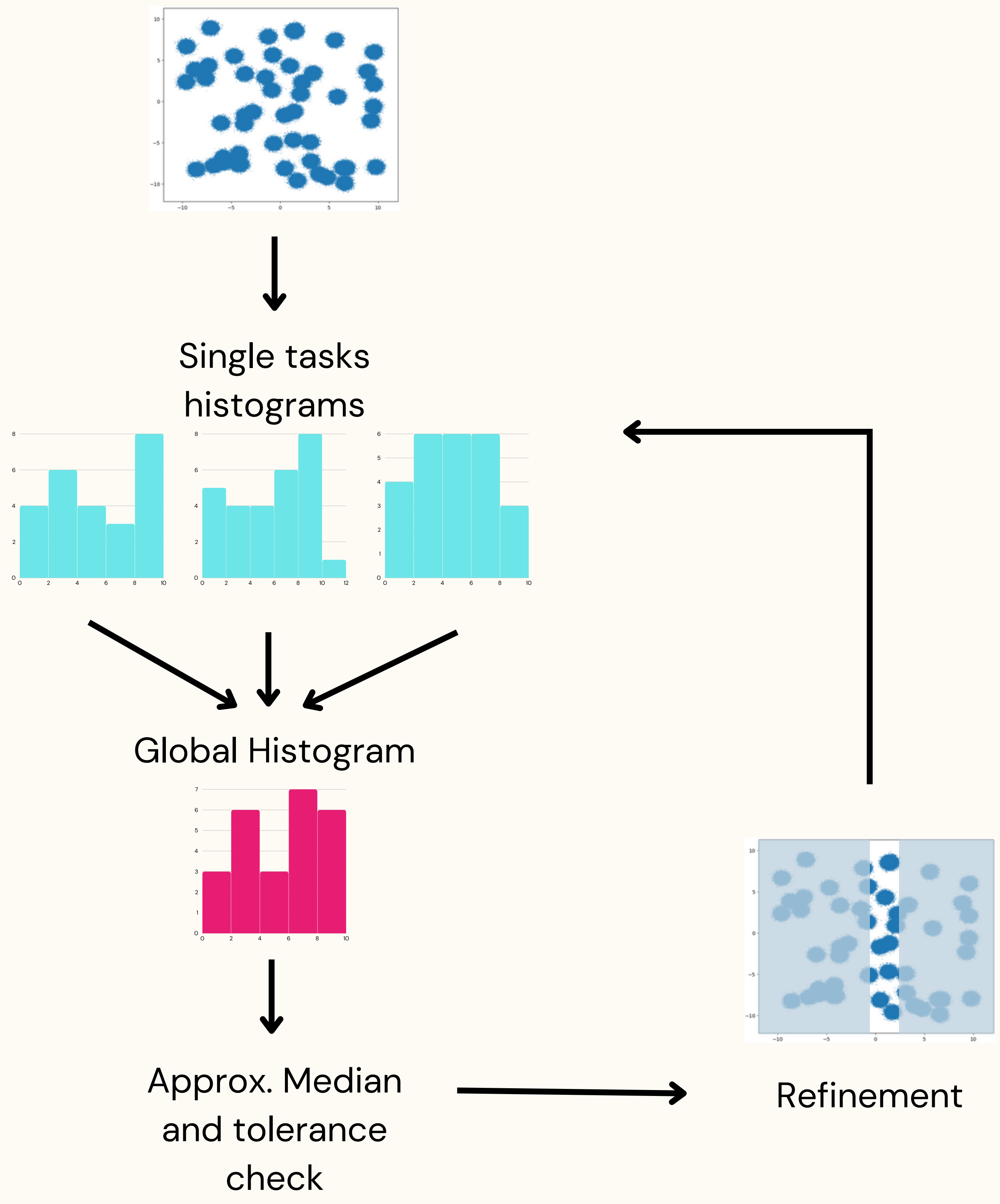
Tree-based methods are very good for low-dimensional data (as in this case).

Such methods already exist for 3d data (eg. oct-tree implemented in the GADGET software).

For d > 3 the best alternative is to build a kd-tree, the main challenge is to compute in parallel the medians required by the algorithm.

The assumption is to have data scattered across MPI task without being decomposed into nonoverlapping domains

Iterative binning approach for approximate medians



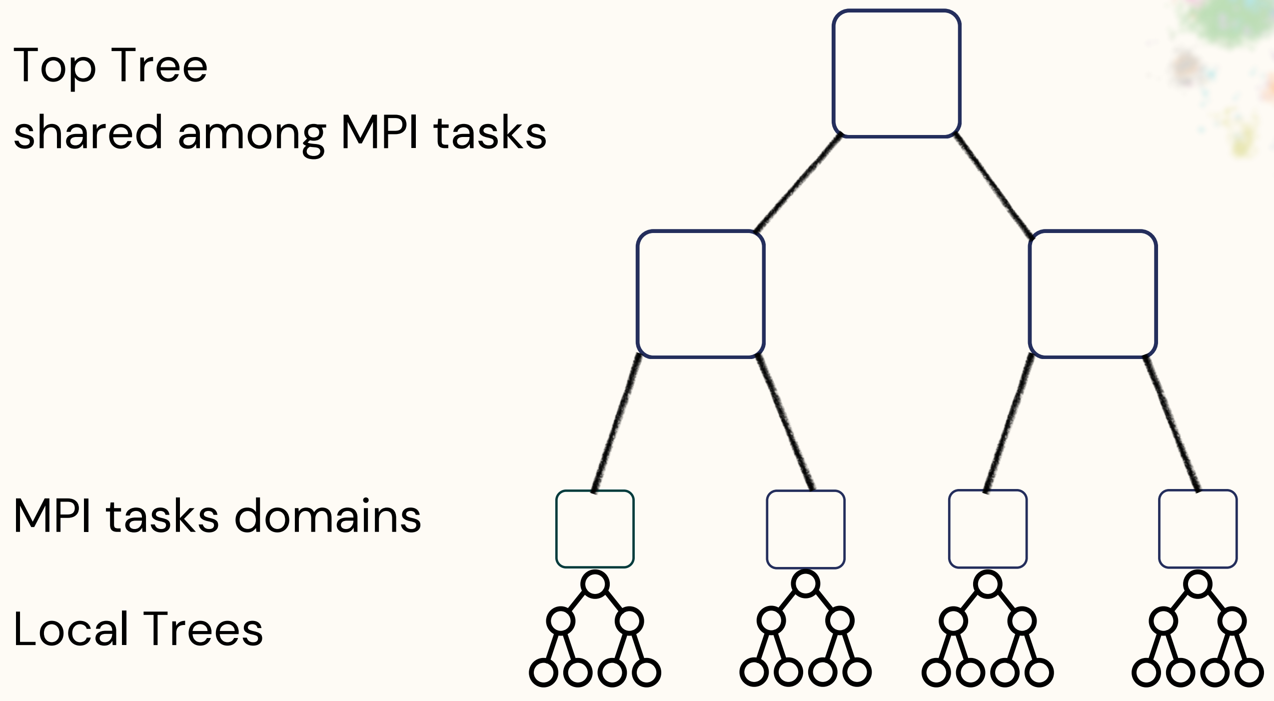
dadaC going parallel
The quest for nearest neighbors

Two level structure:

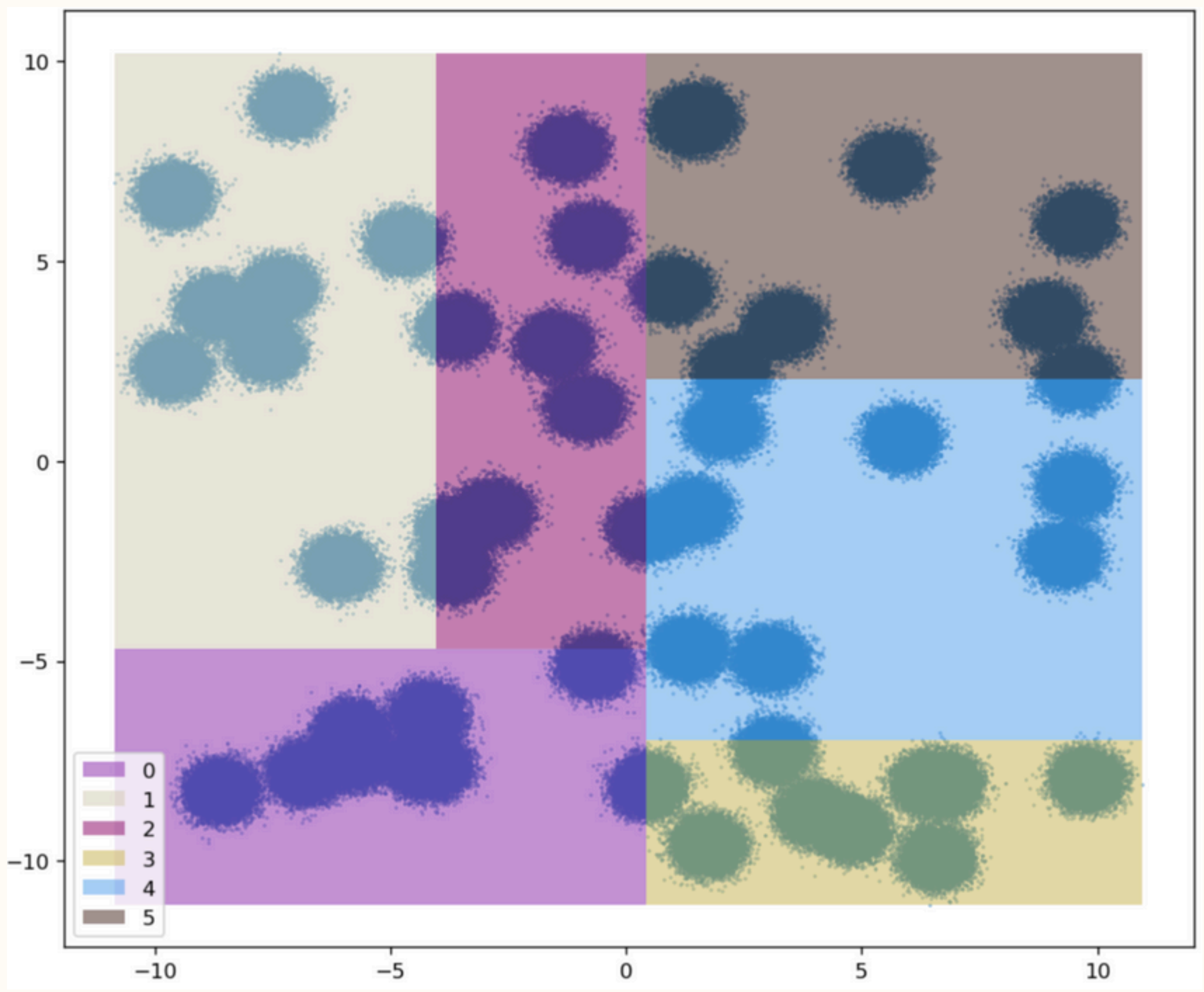
Top tree:
- shared among MPI tasks, each leaf is the domain of a specific task.
- Top nodes do not contain actual points but splitting planes that
- Each leaf contains the same number of points (up to tolerance)

Local trees:
- proper kd-trees built on the domain.
- knn search starts by walking local trees
- top tree is used to compute intersections between the neighborhood of each point and other domains

*The implementation is still in development, in the preliminary test datasets of over 2*10^8 points were handled succesfully*



View of the domain decomposition for 6 tasks for a 2d example

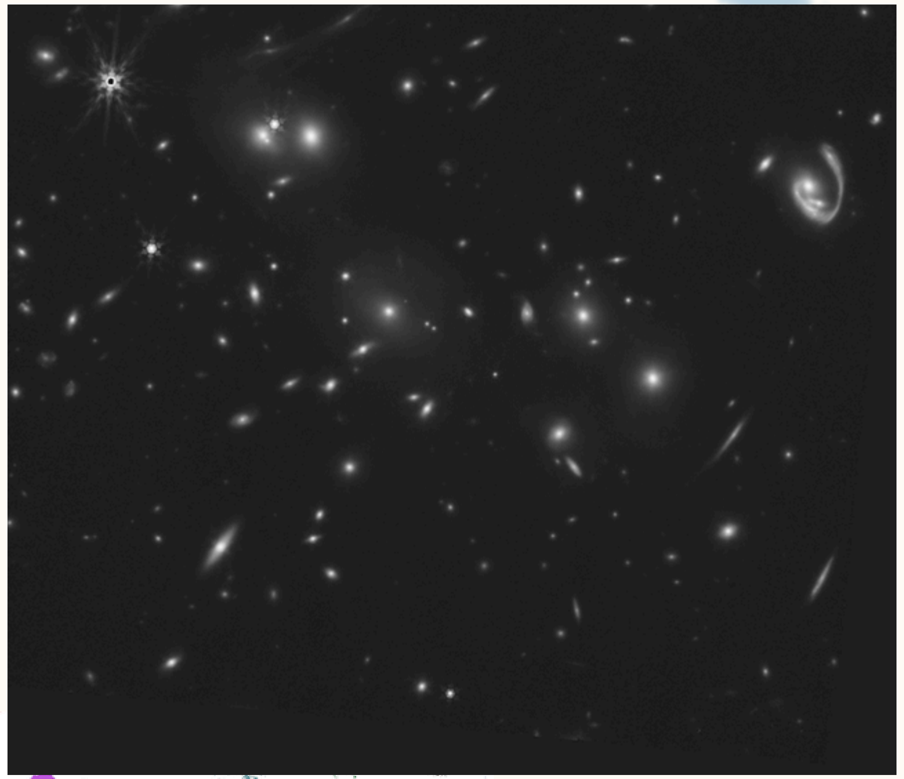


ADP in image processing

Density based clustering & deblending

A source in an astronomical image is a region of pixel with luminosity values higher than the surrounding zone.

The notion of what can be considered a source is so very similar to what a cluster is in a density-based framework



ASTErIsM [2] -> combination of DBSCAN and DENCLUE,
Soucextractor++ [3] -> leverages a hierarchical approach based on successive luminosity thresholds.

[2] Tramacere, A, Paraficz, D, Dubath, P, Kneib, JP, Courbin, F. "ASTErIsM: application of topometric clustering algorithms in automatic galaxy detection and classification". Monthly Notices of the Royal Astronomical Society 2016; 463(3):2939-2957.

[3] https://github.com/astrorama/SourceXtractorPlusPlus

ADP in image processing

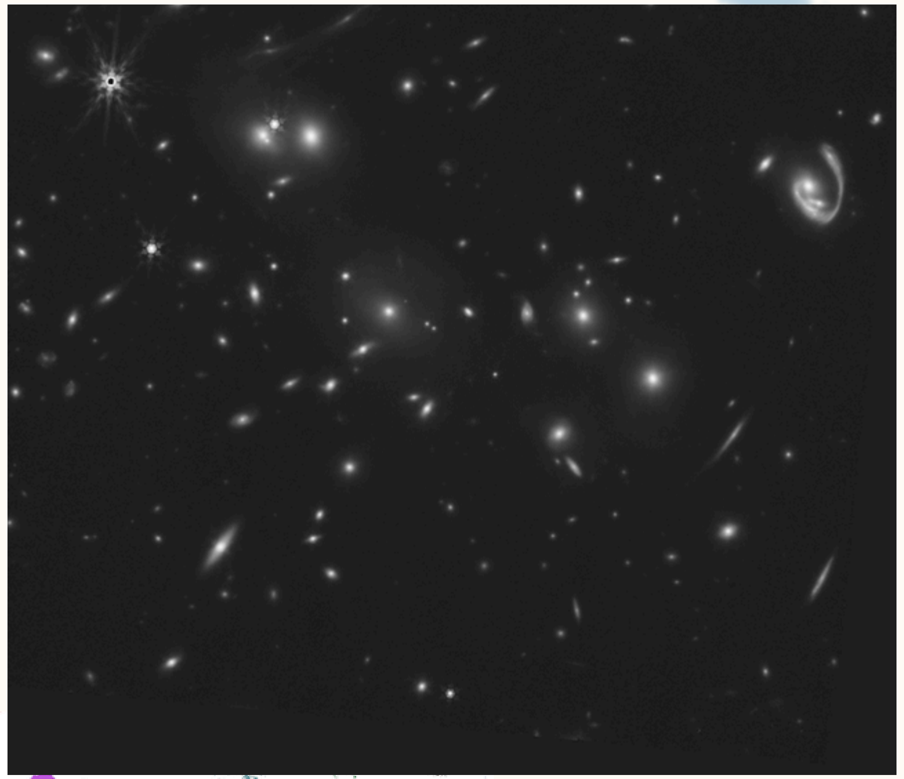
Density based clustering & deblending

Modern telescopes (eg. JWST and Euclid) with higher resolution and sensitivity detect a large number of sources in a small area of the sky.

This increased density leads to more frequent overlaps or blends between objects, making it difficult to distinguish individual sources

Moreover, higher-resolution images unveil sources with more complex morphologies which are more challenging to separate in individual components

The idea is to use ADP as a second step after a detection phase performed with SourceExtractor++



ADP in image processing

Density based clustering & deblending

The detection phase returns a mask to filter out background pixels

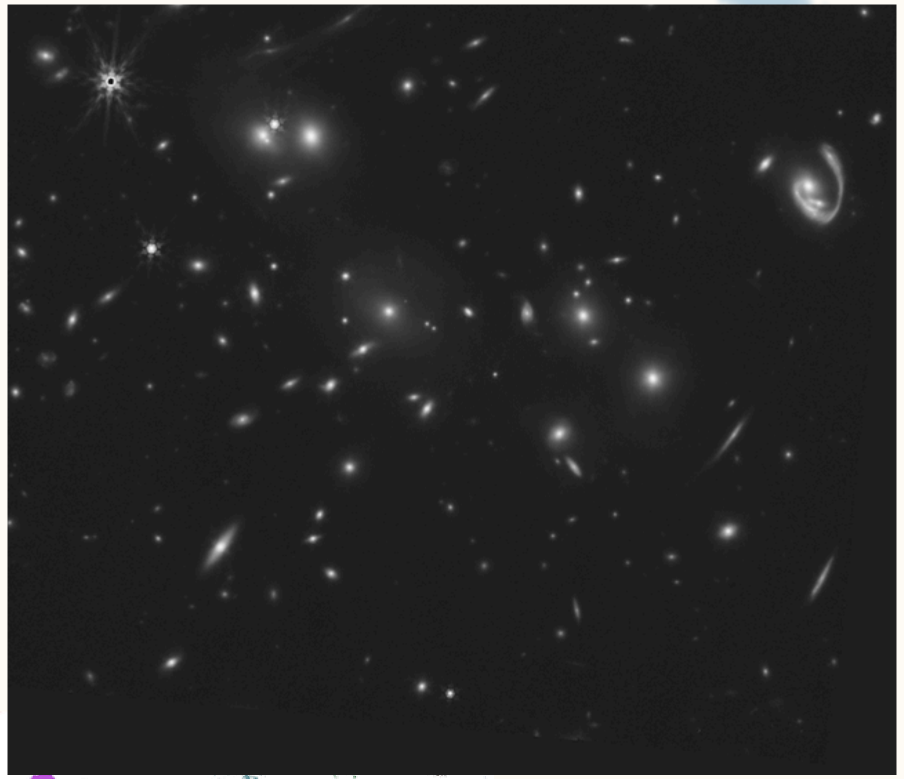
ADP2D

ADP was adapted to tackle image segmentation 2 ways

- Nearest neighbors of a point became neighboring pixels, with k being the radius of the neighborhood
- Density values are obtained as the average over neighboring pixels, error has been taken as the standard deviation

This study is in collaboration with Marius Lepinzan, PhD student at UniTS working at INAF-OATS

Note: ADP can also be applied on its own by employing the background rejection strategy described in ref [2]



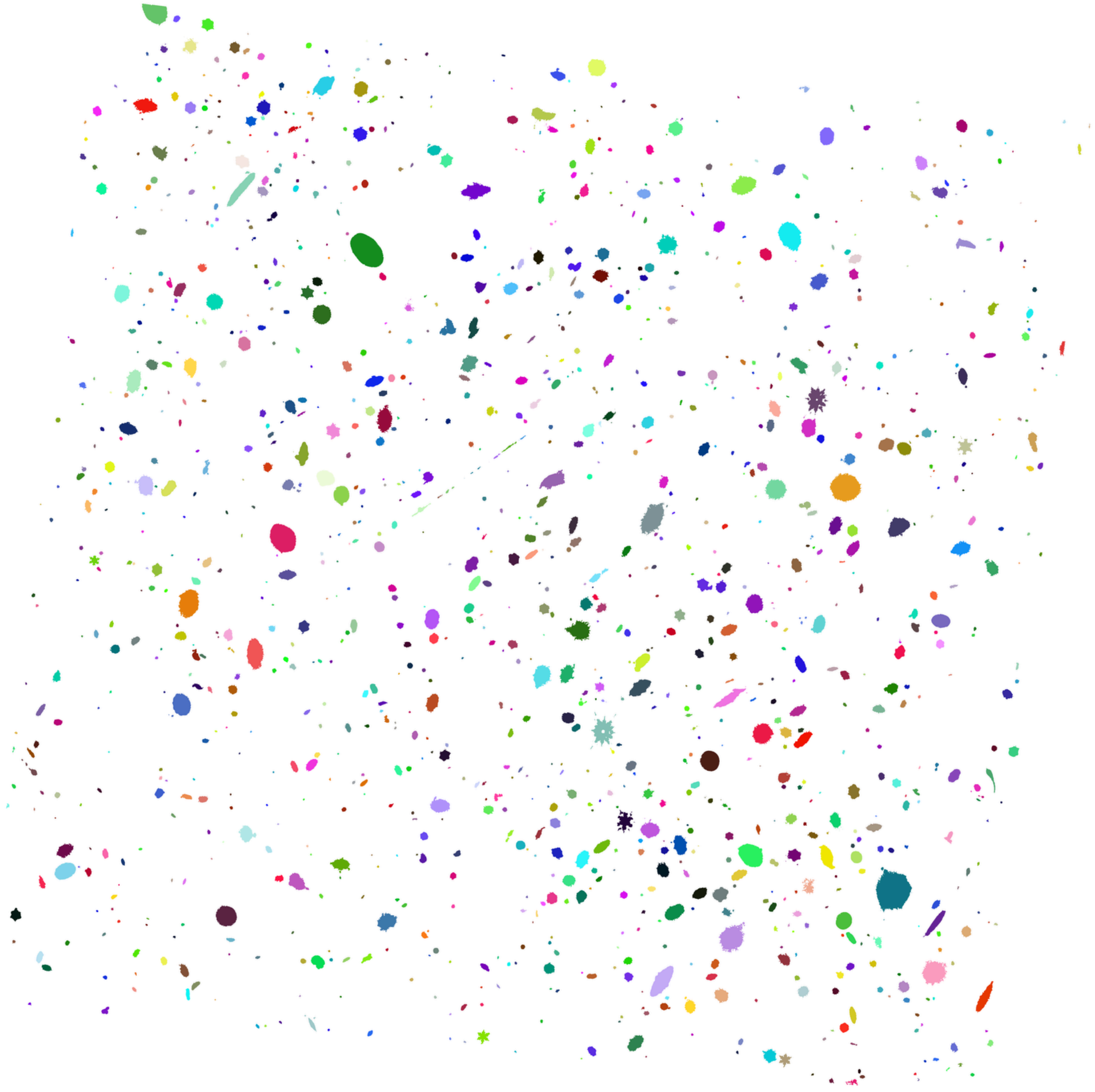
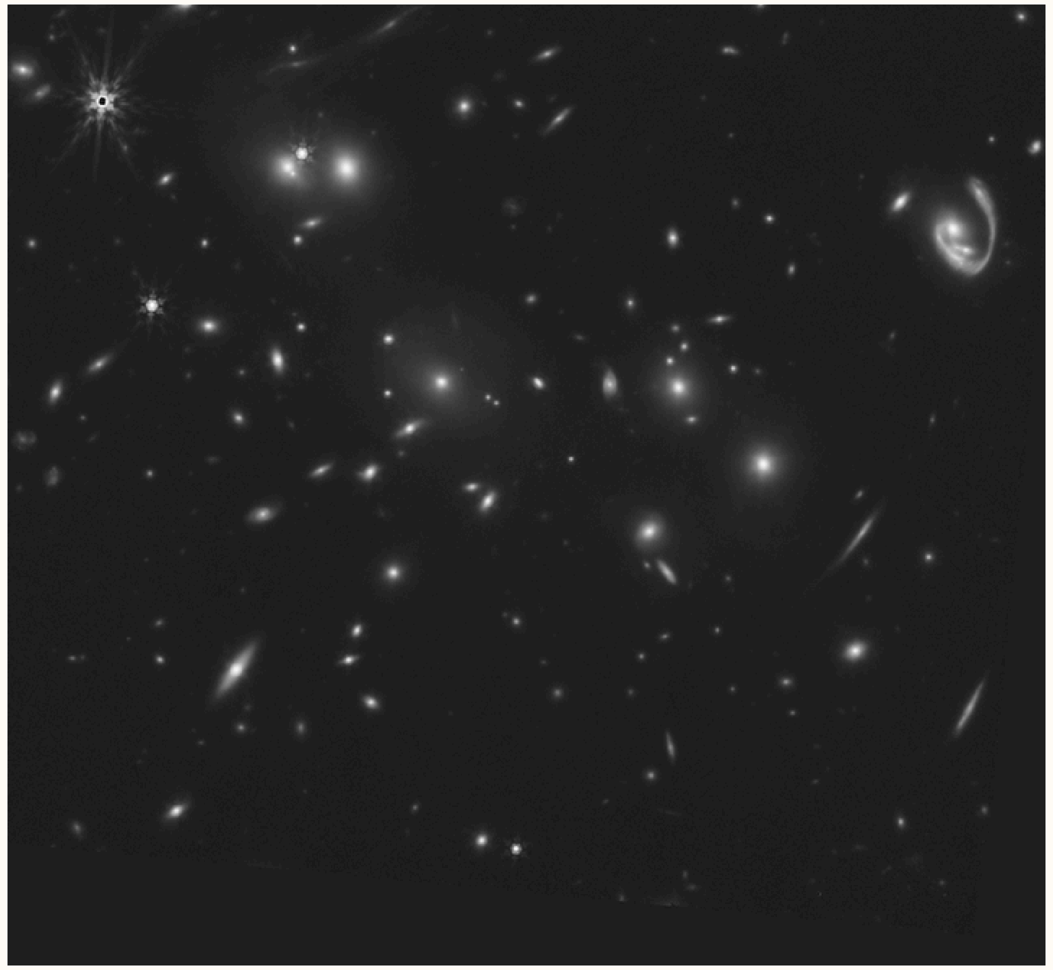
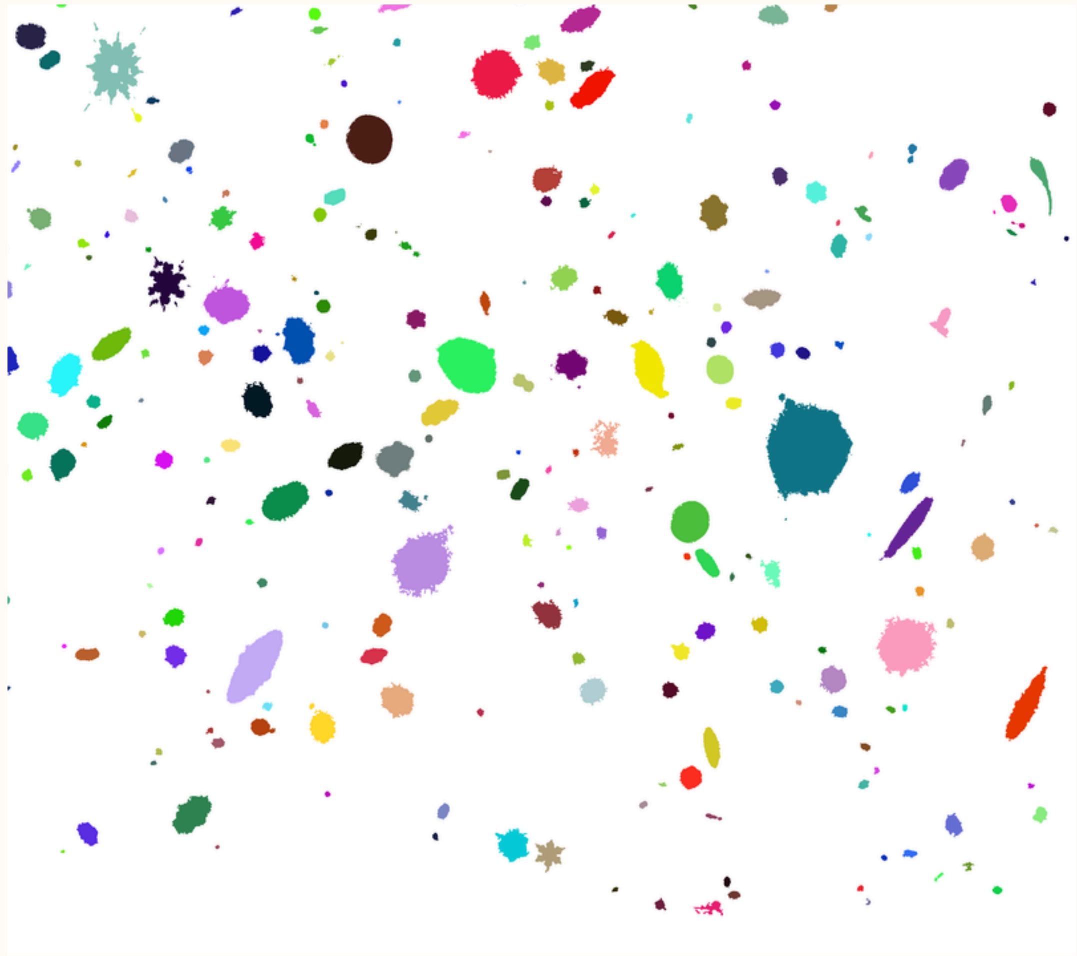



Image segmentation using ADP of “El gordo cluster”,
5600 x 6400 pixels, computational time ~1 min, ~1300 sources found



Original image

Detection phase



Deblending phase

Comparison of the results of detection phase (obtained with SourceExtractor) and the output of ADP on the resulting mask

Image credits

Images from

- “JWST’s PEARLS: Prime Extragalactic Areas for Reionization and Lensing Science: Project Overview and First Results”, Rogier A. Windhorst et al. 2023 AJ 165 13. DOI 10.3847/1538-3881/aca163
- "JWST’s PEARLS: A new lens model for ACT-CL J0102-4915, “El Gordo,” and the first red supergiant star at cosmological distances discovered by JWST", J. M. Diego et al. 2023 A&A 672, A3. DOI 10.1051/0004-6361/202245238
- "The JWST PEARLS View of the El Gordo Galaxy Cluster and of the Structure It Magnifies" Brenda L. Frye et al. 2023 *ApJ* 952 81. DOI 10.3847/1538-4357/acd929
- "PEARLS: Low Stellar Density Galaxies in the El Gordo Cluster Observed with JWST" Timothy Carleton et al. 2023 Accepted for publication in ApJ. DOI 10.48550/arXiv.2303.04726
- "Are JWST/NIRCam color gradients in the lensed z=2.3 dusty star-forming galaxy El Anzuelo due to central dust attenuation or inside-out galaxy growth?" Patrick S. Kamieneski et al. 2023 Accepted for publication in ApJ. DOI 10.48550/arXiv.2303.05054

Thank you for your attention!

Backup

Experimental Results
Assessment strategy

The output of ADP was compared to the one of SUBFIND and the following metric were computed.

Normalized Mutual Information
X, and Y are the cluster assignment of ADP and SUBFIND. It measures the amount of information shared between two random variables. The value ranges from 0 (no similarity) to 1.

$$\mathrm{NMI}(X,Y) = \frac{2 \cdot I(X,Y)}{H(X)+H(Y)}$$

Precision and Recall
Computed by comparing the assignment of couples of particles

$$P = \frac{TP}{TP+FP} \quad R = \frac{TP}{TP+FN}$$

Name	Fraction of points belonging to clusters
LR_091_0001	15 %
LR_091_0000	23 %
g5503149_091_0001	33 %
g6802296_091_0001	18 %
g1574117_Me14_091_0001	23 %

Table of the fraction of particles assigned to a cluster according to SUBFIND. The algorithm also detects particles that are bounded to the FoF and not to any of the substructures or either unbounded from the FoF itself.

- **TP**: a couple generates a true positive signal when the two points have the same cluster label in both methods. (E.g. points i and j are both in cluster c for SUBFIND and both in cluster c' for ADP)
- **FP**: for the first method the points belong to different clusters and for the second one they belong to the same group
- **FN**: for the first method the points belong to the same cluster and for the second one they belong to different ones.
- **TN**: both methods agree on the fact that the points belong to different clusters.

Experimental Results

ADP parameters impact

Histograms of cluster population (number of particles per cluster)