

<sup>1</sup> DRAFT VERSION MARCH 15, 2019

Corresponding author: Felipe L. Gómez-Cortés  
[fl.gomez10@uniandes.edu.co](mailto:fl.gomez10@uniandes.edu.co)

Typeset using L<sup>A</sup>T<sub>E</sub>X **preprint** style in AASTeX62

## A Large Scale Structure Void Identifier for Galaxy Surveys based on the $\beta$ -Skeleton Graph

FELIPE LEONARDO GÓMEZ-CORTÉS,<sup>1</sup> JAIME E. FORERO-ROMERO,<sup>1</sup> AND XIAO-DONG LI<sup>2</sup>

<sup>1</sup>*Physics Department, Universidad de Los Andes, Bogotá, 111711, Colombia*

<sup>2</sup>*School of Physics and Astronomy, Sun Yat-Sen University, Guangzhou 510297, P. R. China*

### 1. THE ALGORITHM

Our approach solving the problem of finding voids in the cosmic web is from a geometrical point of view. We use the  $\beta$ -Skeleton graph to generate the connections between a big set of points belonging to  $\mathbb{R}^3$  in comoving coordinates. This collection consists of two subsets: LSS points and Random Points.

We call the subset of Large Scale Structure tracers as The Observed Catalog (**OC**). They may come from observed galaxies (*e.g.* SDSS) or dark matter halo catalogs from N-body simulations (*e.g.* Abacus Cosmos). The OC shows the well-known filamentary structure that can be found easily by the  $\beta$ -Skeleton libraries. The number of OC points is named  $N_{OC}$ .

The counterpart is the random set of points. They uniformly fill the space, its  $\beta$ -skeleton shows that the mean number of connections different from the filamentary structure mean connections. This Random Catalog (**RC**) is generated using an uniform PDF. The number of RC points is named  $N_{RC}$ .

The Full Catalog (**FC**) of points is created merging the observed catalog and the random catalog. The random points fill the cosmic voids. Then we calculate the FC's  $\beta$ -skeleton. We find random points connected to the LSS tracers near and into the filaments. But we find random points connected only to random points exclusively inside voids, they can be named "pure void points".

The algorithm uses the FC's  $\beta$ -Skeleton to discard random points connected to LSS tracers. This first filtered  $\beta$ -Skeleton graph is used by the algorithm to identify the first connections of pure void points and stores the list of results in a first particle void list. The algorithm checks element by element if pure voids are connected or share connections, if the condition is true, then the two sets are merged. At the ending we have a set of voids. Each void is a list of points connected between them. Two voids never are connected.

#### 1.1. *Calibrating our free parameters*

How to select  $\beta$  and the ratio  $N_{rand}/N_{obs}$

#### 1.2. *Measuring Void Size*

A first approach is to assign a volume proportional to the number of particles in the void times the density (total number of random particles over total catalog volume).

A second method assumes the voids as ellipsoid objects. We use the standard equation for the surface of an ellipsoid:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \quad (1)$$

so,  $a$ ,  $b$  and  $c$  are the semi-axes of the ellipsoid. The volume enclosed is:  $V = \frac{4\pi}{3}abc$ . Semi-axes are found using the inertia tensor of the point distribution.

$$I = \frac{m}{5} \begin{pmatrix} b^2 + c^2 & 0 & 0 \\ 0 & a^2 + c^2 & 0 \\ 0 & 0 & a^2 + b^2 \end{pmatrix} \quad (2)$$

Numpy's *linalg* module calculates the eigenvalues. It returns

$$I = \begin{pmatrix} I_a & 0 & 0 \\ 0 & I_b & 0 \\ 0 & 0 & I_c \end{pmatrix} \quad (3)$$

and the rotation matrix (eigenvectors).

Assuming  $m = 1$ , or giving a mass  $m_i = 1/N_{void}$  for each one of the  $N_{void}$  particles of the void, we can calculate the void semi-axes independently from the number of particles in the void.

$$\begin{aligned} a &= \sqrt{\frac{5}{2N} (-I_a + I_b + I_c)} \\ b &= \sqrt{\frac{5}{2N} (I_a - I_b + I_c)} \\ c &= \sqrt{\frac{5}{2N} (I_a + I_b - I_c)} \end{aligned} \quad (4)$$

They may not be sort as  $a > b > c$ , but this is solved in a few code lines.

### 1.3. Void Abundance