# Code documentation

October 7, 2025

The present document provides technical documentation for the Python-based routines to compute the organization metrics $L_{\text{org}}$ and $I_{\text{org}}$ freely available on github at `https://github.com/giobiagioli/organization_indices` and used in Biagioli and Tompkins (2023).

Given the distribution of a set of objects over a subregion of the plane, the routines compute all-pair separation distances and estimate the corresponding nearest-neighbor cumulative distribution function (NNCDF) and Besag's $L$-function. The latter is a measure of the mean number of neighbors of any element of the pattern for a range of neighborhood sizes, rescaled by the spatial density of events (Ripley, 1976; Besag, 1977; Ripley, 1979, 1981) The routines also compute the NNCDF and $L$-function that would be expected theoretically if the *same* number of objects were *randomly* distributed over the same region. The comparison of the theoretical and estimated functions allows to compute indices that assess the level of organization of convective cloud scenes, respectively $I_{\text{org}}/RI_{\text{org}}$ (Tompkins and Semie, 2017; Seifert and Heus, 2013; Biagioli and Tompkins, 2023) and $L_{\text{org}}/dL_{\text{org}}$ (Biagioli and Tompkins, 2023).

The present routines are designed to operate under a broad range of configurations, both in terms of domain geometry and details of the calculation algorithms. They therefore include a set of flags that specify under which conditions the organization degree of a scene is assessed (square/rectangular domains with open/periodic boundaries, use of four-connectivity clustering algorithms, Poisson/binomial models as reference for spatial randomness).

## 1 Input parameters

The basic syntax of the main routine `_compute_organization_indices` is

```
_compute_organization_indices(params)
```

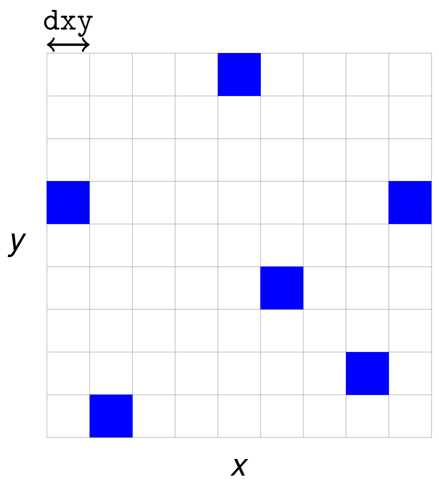
where `params` is a dictionary whose keys are the following variables:

`dxy, cnv_idx, rmax, bins, periodic_BCs, periodic_zonal, clustering_algo, binomial_continuous,binomial_discrete, edge_mode.`

The input parameters dxy, cnv_idx, rmax, bins describe the specifics of the gridded dataset and the details of the binning procedure to compute the pair separation distances, and in particular:

- dxy is the horizontal grid spacing, assumed to be the same in both directions. The user can provide it in units of length (e.g., if the resolution of the dataset is $\Delta x = 2$ km, then dxy should be set to 2000) or in grid units ($\texttt{dxy} = 1$). Please note that, in this latter case, the cloud-to-cloud distances as measured by the routine are in grid units as well.

- cnv_idx is the 2D binary field of convective ($\texttt{cnv\_idx} = 1$) and non-convective ($\texttt{cnv\_idx} = 0$) grid cells, obtained by applying some specific thresholding approach to relevant variables or resulting from synthetic generation of scenes. The Python object class must be numpy.ndarray. In the example below, the convective grid boxes are represented as blue squares and the corresponding binary matrix cnv_idx is shown.



$$\Rightarrow \texttt{cnv\_idx} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- rmax is the maximum search radius ($r_{\max}$ in the manuscript) or box size ($\ell_{\max}$) for evaluation of neighbor counts. Typical values for practical cases are provided in section 2. Units must be consistent with those specified in dxy.

- bins is the array (Python object type numpy.ndarray) specifying the rightmost edge of distance/box size bands in which to evaluate the event counts. Typically, bin edges range from 0 to rmax with spacing given by a multiple (or a submultiple) of the grid resolution:

```
bins = np.arange(0, rmax+bin_w*dxy, bin_w*dxy)
```

  where bin_w is the parameter that sets the bin width. The user can alternatively use the command np.linspace.
  For instance, assume the discrete organization index $dL_{\text{org}}$ is to be computed for

a dataset with spatial resolution dxy $= 1$ km over a doubly-periodic square observation window of size 500 km. The user might want to investigate the spatial organization trends of a cloud pattern over the whole domain, in which case rmax $=$ 500e3 (units of length). A reasonable choice for the width of box size bins would be bin_w*dxy $= 2$ km (i.e., bin_w $= 2$). For any point of the pattern, the number of neighboring events are counted which are less (or equal) than 2 km, 4 km, 6 km, $\ldots$, 500 km from the base point.

The parameters periodic_BCs and periodic_zonal are boolean and relate to details of the domain geometry. In particular,

- periodic_BCs is to specify whether the domain is periodic in both $x$ and $y$ directions, in which case periodic_BCs = True.

- periodic_zonal is to specify whether the domain is periodic in the zonal direction and open in the meridional direction (periodic_zonal = True).

The flags periodic_BCs and periodic_zonal cannot be simultaneously True, in which case the code returns an error. If the user wants to prescribe open boundary conditions, then both periodic_BCs and periodic_zonal are to be set False.

For open gridded domains, the parameter edge_mode accounts for edge effect correction strategies in the computation of the estimated $L$-functions. Possible choices are:

- edge_mode = 'none', in which case no correction strategies are employed.

- edge_mode = 'besag', in which case the area-based correction method for edge effects (Besag, 1977) is adopted (cf. Fig. 3c in Biagioli and Tompkins (2023)).

Note that, in case of open domains, the routines are not designed to operate if continuous reference models for randomness (either Poisson or binomial) are assumed, namely, if one wants to consider the case of open boundaries, the flag binomial_discrete must be set True (see below). The user can anyway refer to the python class RipleysKEstimator in the package astropy, that contains built-in functions available for edge correction methods in case of random Poisson processes (see https://docs.astropy.org/en/stable/stats/ripley.html).

The parameters clustering_algo, binomial_continuous, binomial_discrete are boolean and are related to the specifics of the cluster identification algorithm and the choice of the theoretical model for random distributions of points (Poisson vs binomial). In detail,

- clustering_algo specifies if a four-connectivity clustering algorithm is to be used for identifying adjacent convective cells as a single cluster (clustering_algo = True). If False, contiguous convective objects are treated as separate entities.

- binomial_continuous uses the continuous binomial model as a standard of randomness if True. Note that binomial_continuous = True only works if doubly-periodic boundaries are prescribed. The analytical formulas for the reference random distributions in this case are provided in Biagioli and Tompkins (2023), equations (18) for square domains and (22) for non-square domains.

- binomial_discrete uses the discrete binomial model (cf. Fig. 3d in Biagioli and Tompkins, 2023) as a reference for randomness, if True.

Note that the two options binomial_continuous and binomial_discrete cannot be set simultaneously True, with the code returning an error message in this case. If the user wants to consider the Poisson model eqn. (10) for complete spatial randomness, the flags binomial_continuous and binomial_discrete have to be set False.

## 2 Main example cases

In this section, we first show a possible way of generating random synthetic cloud field scenes for square and rectangular domains and then, assuming both periodic and open boundaries and Poisson and binomial reference random models, we show which input parameters are needed to compute the indices $L_{\text{org}}/dL_{\text{org}}$. The user can create or analyze clustered, regular or composite scenes through input of a proper cnv_idx argument.

In order to create a random cloud field scene *without replacement* (i.e., no two objects can occupy the same location), the simple sequence of Python commands reported below can be used, where nx and ny are the number of grid points in the $x$ and $y$ directions, respectively, and ncnv is the number of convective elements in the domain.

```
cnv_idx = np.zeros((ny, nx), dtype=int)
coords = np.random.choice(ny * nx, ncnv, replace=False)
new_loc = np.unravel_index(coords, (ny, nx))
cnv_idx[new_loc] = 1
```

### 2.1 Square domains

#### 2.1.1 Poisson model for randomness, cyclic boundaries

In this case, the theoretical $L$-function for spatial randomness is given by $\tilde{L}(r) = r_{\text{max}}^{-1} r$ and the estimated one is eqn. (11) in Biagioli and Tompkins (2023), rescaled by $r_{\text{max}}$ and with $w_i = 1$. The user can select the desired maximum search radius $r_{\text{max}}$. Here, and in all the examples shown in Biagioli and Tompkins (2023), $r_{\text{max}}$ is the maximum distance between any two points in a periodic square domain of size $D$, i.e., $r_{\text{max}} = D/\sqrt{2}$. Consequently bins is set as bins = np.arange(0,rmax+dxy,dxy), where dxy is the dataset resolution. The value of clustering_algo is to be specified according to the description above. The other input parameters must be periodic_BCs = True, periodic_zonal = False, binomial_continuous = False, binomial_discrete = False, edge_mode = 'none'. The result is similar to Fig. 4a in Biagioli and Tompkins (2023).

#### 2.1.2 Continuous binomial model for randomness, cyclic boundaries

In this case, the theoretical and estimated $L$-functions are given by eqns. (18) and (11) in Biagioli and Tompkins (2023), both normalized by $r_{\text{max}}$ and with $w_i = 1$. The maximum search radius $r_{\max}$ is again $r_{\max} = D/\sqrt{2}$. The other arguments must be `periodic_BCs = True`, `periodic_zonal = False`, `binomial_continuous = True`, `binomial_discrete = False`, `edge_mode = 'none'`. The result is similar to Fig. 4d in Biagioli and Tompkins (2023).

#### 2.1.3 Discrete binomial model for randomness, cyclic boundaries

In this case, the theoretical and estimated $L$-functions are given by eqns. (19) and (20) in Biagioli and Tompkins (2023), both normalized by $\ell_{\max}$ and with $w_i = 1$. In a square periodic domain of size $D$, $\ell_{\max}$ (`rmax` in the routine) is set to `rmax` $= D$, which is the maximum size of a search box centered on any given object of the pattern. In view of this, a possible choice for `bins` is `bins = np.arange(0,rmax+2*dxy,2*dxy)`, where the factor of 2 is due to the fact that now boxes (and not distances) are considered (cf. Figs. 3a and 3d in Biagioli and Tompkins, 2023). The other arguments must be `periodic_BCs = True`, `periodic_zonal = False`, `binomial_continuous = False`, `binomial_discrete = True`, `edge_mode = 'none'`. The result is similar to Fig. 4g in Biagioli and Tompkins (2023).

#### 2.1.4 Discrete binomial model for randomness, open boundaries

The case of open boundaries is typical of the analysis of observational datasets. The theoretical and estimated $L$-functions are again given by eqns. (19) and (20) in Biagioli and Tompkins (2023), both normalized by $\ell_{\max}$. The weighting factors $w_i$ in the estimated $L$-function are now imposed as specified in the text (Section 4d). In a square open domain of size $D$, the maximum search box size is given by $\ell_{\max} = 2D$, hence `rmax` $= 2D$, but, as usual, the user can restrict the analysis to any subregion of the observation window and select `rmax` accordingly. The other arguments must be `periodic_BCs = False`, `periodic_zonal = False`, `binomial_continuous = False`, `binomial_discrete = True`, `edge_mode = 'besag'`. With `edge_mode = 'none'`, the correcting factors $w_i = 1$ and the estimated $L$-function will exhibit a regularity bias due to event undercounting in the long range.

#### 2.1.5 Discrete binomial model for randomness, zonally cyclic boundaries

In this case, the estimated $L$-function is given by eqn. (20) in Biagioli and Tompkins (2023), with weights $w_i$ obeying the area-based local correction strategy described in the manuscript. The theoretical $L$-function is instead

$$L(\ell_n) = \begin{cases} \sqrt{\frac{N_c - 1}{N_c}\left(n^2 - 1\right)\Delta x^2} \sim n\Delta x \equiv \ell_n & \text{for } 0 \leq \ell_n \leq D, \\ \sqrt{\frac{N_c - 1}{N_c}\left(\left(n^2 - 1\right)\Delta x^2 - n\Delta x(n\Delta x - D)\right)} \sim \sqrt{\ell_n D} & \text{for } \ell_n > D, \end{cases} \tag{1}$$

where $D$ is the size of the square periodic domain and the approximation holds for reasonable sample sizes. Both the theoretical and estimated functions are to be rescaled by

$\ell_{\max}$ (`rmax` in the routine), which, when considering the spatial tendencies of a pattern over the entire domain, must be `rmax` $= 2D$. The other arguments are `periodic_BCs = False`, `periodic_zonal = True`, `binomial_continuous = False`, `binomial_discrete = True`, `edge_mode = 'besag'` (recall that the domain is open in the meridional direction).

## 2.2 Non-square domains

Given a rectangular region of width $D_x$ and height $D_y$, in the sequel without loss of generality we will assume $D_x > D_y$, but the routines are designed to operate also in the other case.

### 2.2.1 Poisson model for randomness, cyclic boundaries

The theoretical and estimated $L$-functions are eqns. (10) and (11) in Biagioli and Tompkins (2023), both normalized by $r_{\max}$ and with $w_i = 1$. $r_{\max}$ is the maximum distance between any two points in a cyclic domain of width $D_x$ and height $D_y$ i.e., $r_{\max} = \sqrt{D_x^2 + D_y^2}/2$. The other input parameters are as in section 2.1.1.

### 2.2.2 Continuous binomial model for randomness, cyclic boundaries

In this case, the theoretical and estimated $L$-functions are given by eqns. (22) and (11) in the manuscript, normalized by $r_{\max}$ and with $w_i = 1$. $r_{\max}$ is again set to $r_{\max} = \sqrt{D_x^2 + D_y^2}/2$. The other arguments are as in section 2.1.2.

### 2.2.3 Discrete binomial model for randomness, cyclic boundaries

In this case, the theoretical and estimated $L$-functions are given by eqns. (23) and (20) in Biagioli and Tompkins (2023), normalized by $\ell_{\max}$ and with $w_i = 1$. In a rectangular periodic domain of width $D_x$ and height $D_y$, the maximum size of a box for neighbor search that is centered on any given object of the pattern is $\ell_{\max} = \max\{D_x, D_y\}$ and `rmax` is chosen accordingly in the specification of the input parameters. The other arguments, including `bins`, are as in section 2.1.3.

### 2.2.4 Discrete binomial model for randomness, open boundaries

The theoretical and estimated $L$-functions are provided by eqns. (19) and (20) in Biagioli and Tompkins (2023), both normalized by $\ell_{\max}$. The weighting factors $w_i$ in the estimated $L$-function are computed according to the area-based edge-correction strategy (cf. Section 4d in the manuscript) if `edge_mode = 'besag'`. In a rectangular open domain of width $D_x$ and height $D_y$, the maximum search box size is $\ell_{\max} = 2\max\{D_x, D_y\}$, and `rmax` in the routine can be selected as such. However, the user can restrict the analysis to any subregion of the observation window and choose the desired `rmax`. The other arguments must be as in section 2.1.4.

### 2.2.5 Discrete binomial model for randomness, zonally cyclic boundaries

The estimated $L$-function is given by eqn. (20) in Biagioli and Tompkins (2023) rescaled by $\ell_{\max}$ and with weighting factors $w_i$ calculated according to the area-based correction technique for edge effects. Regarding the computation of the theoretical $L$-function, distinct cases have to be considered:

- if $D_x \geq 2D_y$, the maximum search box size is $\texttt{rmax} = D_x$ and, since the search boxes are entirely contained within the domain, no corrections for multiple counting due to the periodic continuation of the domain in the zonal direction have to be imposed. The theoretical $L$-function is therefore provided by eqn. (19) in the manuscript, rescaled by $\ell_{\max}$ (`rmax` in the code).

- if $D_x < 2D_y$ (or if $D_y > D_x$), the maximum search box size is $\texttt{rmax} = 2D_y$ and the search boxes can exceed the domain in the zonal direction, therefore a correction must be included to avoid multiple counting. Using simple arguments from the stochastic geometry, analogous to those used in Biagioli and Tompkins (2023), we have

$$L(\ell_n) = \begin{cases} \sqrt{\dfrac{N_c - 1}{N_c}\left(n^2 - 1\right)\Delta x^2} \sim n\Delta x = \ell_n & \text{for } 0 \leq \ell_n \leq D_x, \\ \sqrt{\dfrac{N_c - 1}{N_c}\left(n\Delta x D_x - \Delta x^2\right)} \sim \sqrt{\ell_n D_x} & \text{for } \ell_n > D_x. \end{cases} \tag{2}$$

In any case, the other input parameters must be as in section 2.1.5.

## 3 Output

The main routine `_compute_organization_indices` returns the values of the indices $I_{\text{org}}$, $RI_{\text{org}}$ and $L_{\text{org}}/dL_{\text{org}}$ and also the theoretical and estimated NNCDFs and $L$-functions to allow visual inspection of their profiles. In detail, the output consists of a dictionary containing the following entries:

- `I_org` and `RI_org`, values of the $I_{\text{org}}$ and $RI_{\text{org}}$ indices as defined in Biagioli and Tompkins (2023), eqns. (2) and (3).

- `L_org`, value of the index $L_{\text{org}}$ or $dL_{\text{org}}$ depending on whether a continuous (Poisson/binomial) or discrete (binomial) model is assumed as a standard of randomness.

- `NNCDF_theor` and `NNCDF_obs`, theoretical and estimated nearest-neighbor cumulative distribution functions. The theoretical one is the Weibull cumulative distribution function eqn. (1) in Biagioli and Tompkins (2023), the observed one is derived from the distribution of the objects in the given scene.

- `Besag_theor` and `Besag_obs`, theoretical and estimated Besag's $L$-functions, computed as discussed in section 2 for the different cases.

## References

Besag, J. (1977). Contribution to the Discussion on Dr. Ripley's Paper. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(2):193–195.

Biagioli, G. and Tompkins, A. M. (2023). Measuring Convective Organization. *Journal of the Atmospheric Sciences*, 80(12):2769 – 2789.

Ripley, B. D. (1976). The second-order analysis of stationary point processes. *Journal of Applied Probability*, 13(2):255–266.

Ripley, B. D. (1979). Tests of 'Randomness' for Spatial Point Patterns. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(3):368–374.

Ripley, B. D. (1981). *Spatial Statistics*. Wiley Series in Probability and Statistics. Wiley.

Seifert, A. and Heus, T. (2013). Large-eddy simulation of organized precipitating trade wind cumulus clouds. *Atmospheric Chemistry and Physics*, 13(11):5631–5645.

Tompkins, A. M. and Semie, A. G. (2017). Organization of tropical convection in low vertical wind shears: Role of updraft entrainment. *Journal of Advances in Modeling Earth Systems*, 9(2):1046–1068.