Spatial statistics: Object-based colocalization

The data

IAC and BOB are proteins in the eastern spruce budworm (*Choristoneura fumiferana*) epidermis

You hypothesize that the spatial interaction between IAC and BOB changes with temperature and season.

Extract coordinates

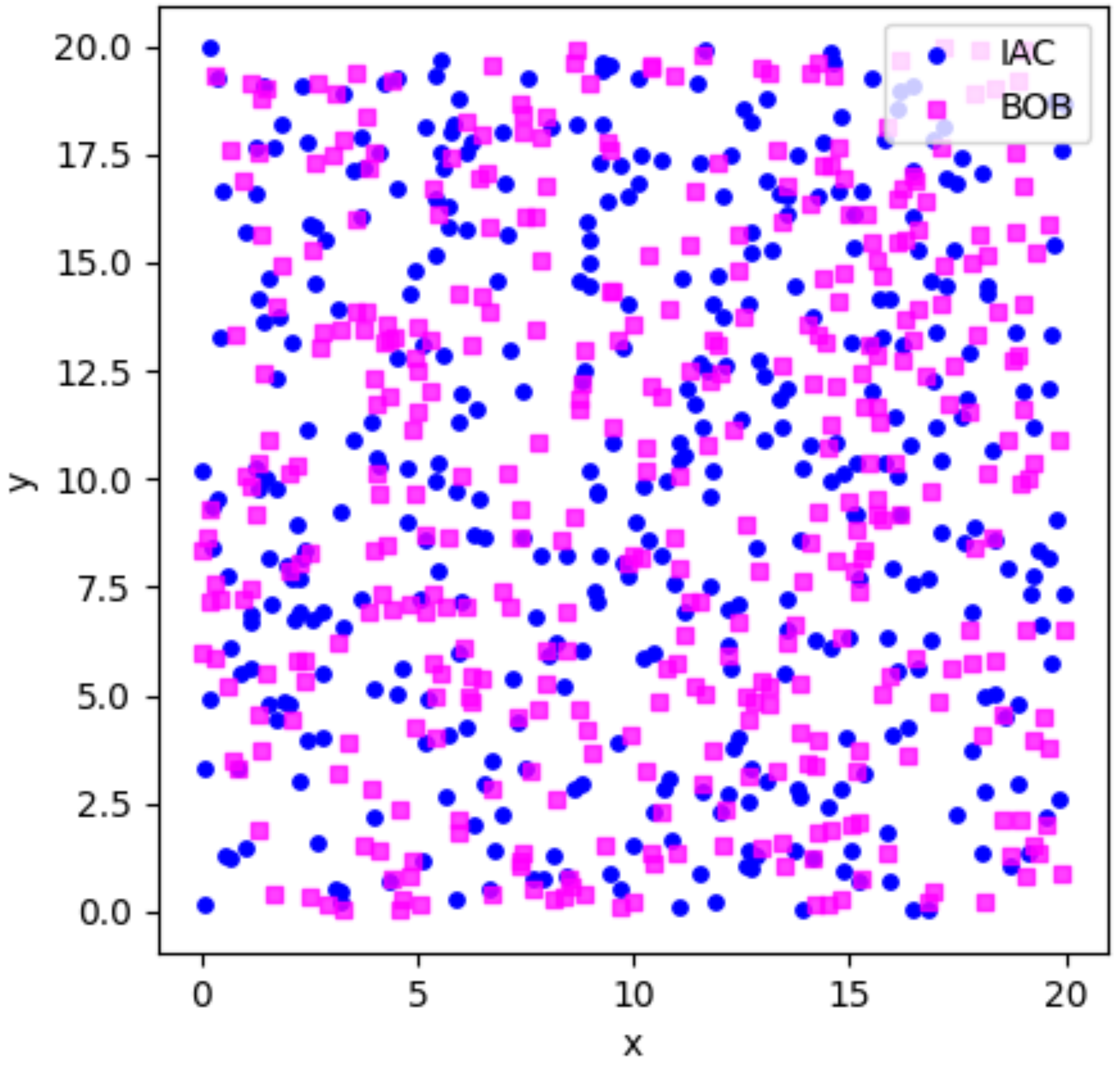
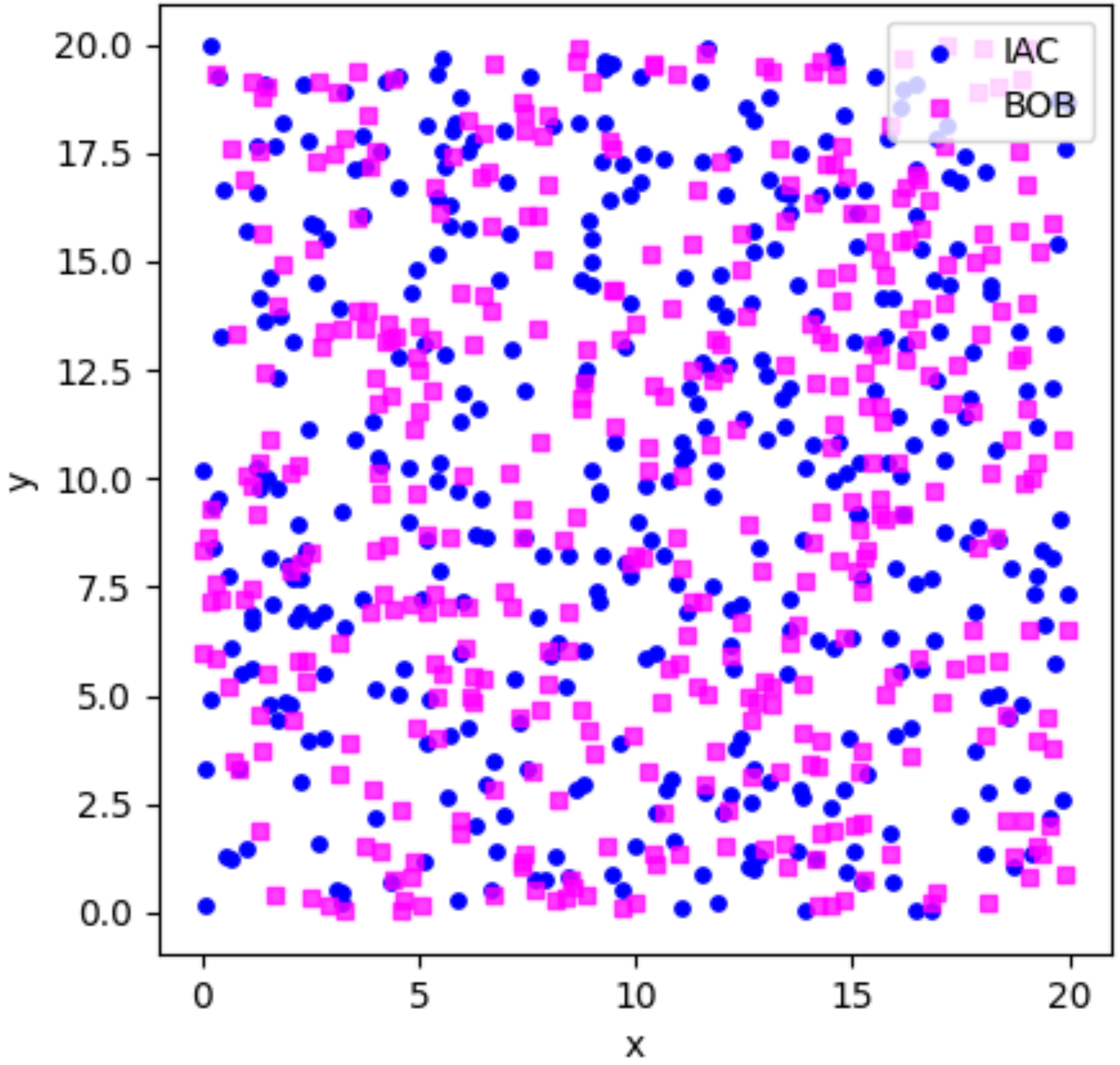


Image: https://commons.wikimedia.org/wiki/File:Choristoneura_fumiferana_larva.jpg; public domain

The data



IAC

x y

```
1.6542,4.98028
2.64547,4.95783
6.90183,17.5005
6.98563,17.8339
7.24287,17.1596
6.57113,17.3483
6.41189,16.8281
6.62601,17.4194
6.21376,17.4324
6.74328,16.7232
7.0172,17.5627
6.52185,16.5556
5.93571,17.4
6.28006,17.0457
...  ...
```

BOB

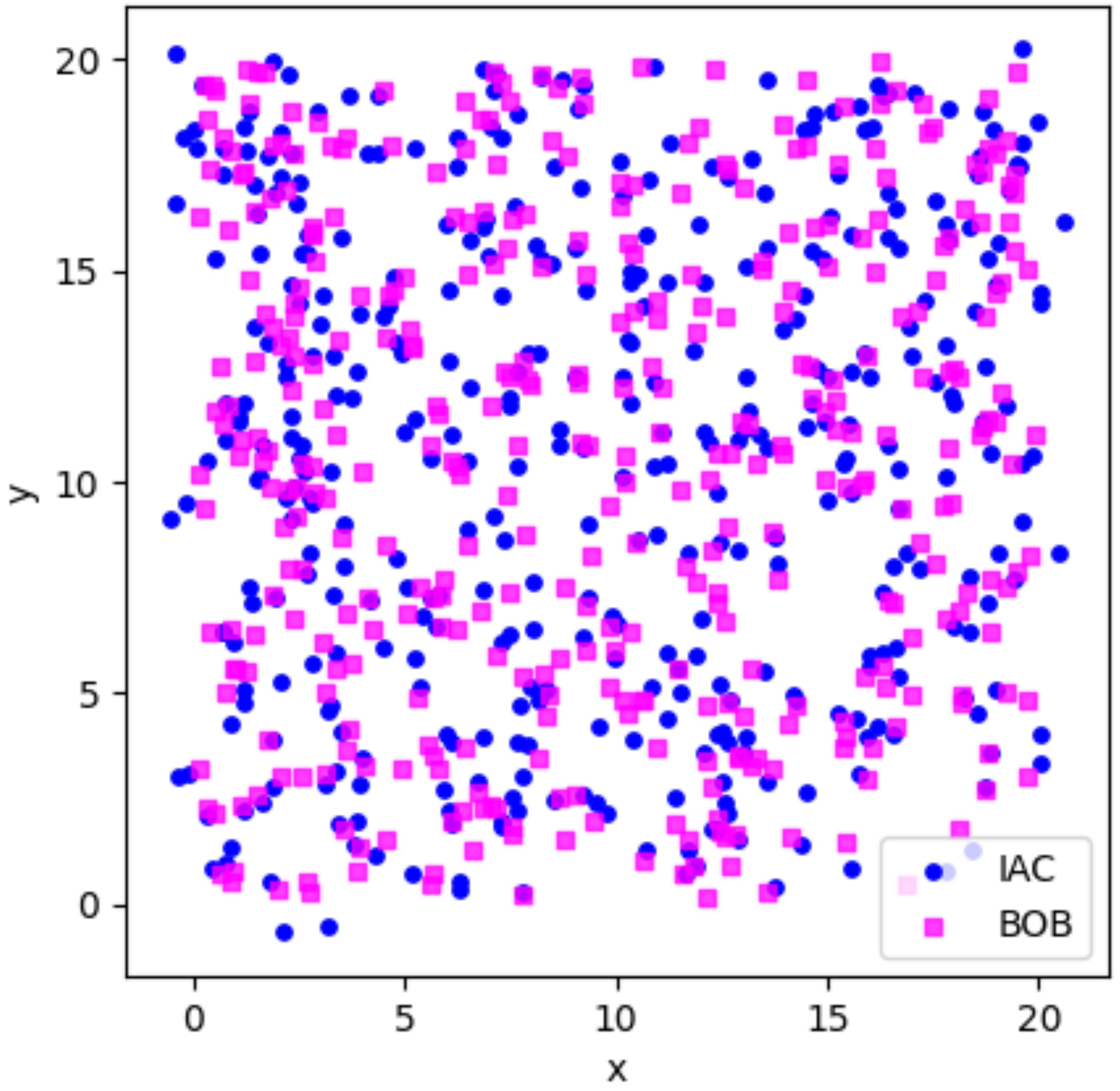
x y

```
2.59176,11.6148
2.35522,12.7033
3.60981,12.9357
2.91734,12.0081
2.46703,12.7667
2.60448,11.849
2.36841,12.6463
1.24649,11.4218
3.67557,4.29607
2.63406,4.7991
2.77047,4.19997
2.90153,4.83014
2.45598,4.98462
4.02456,4.89246
...  ...
```

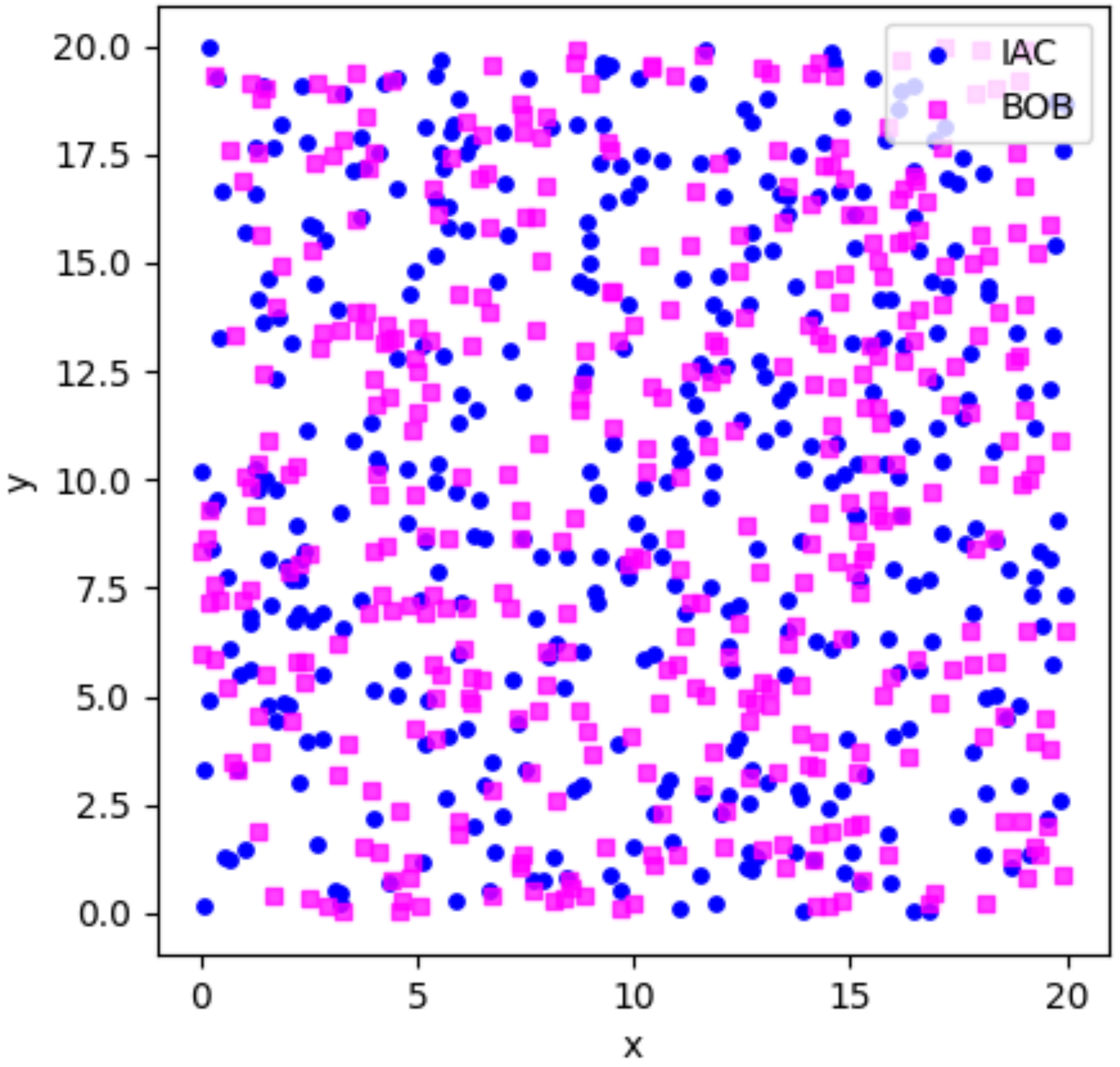
The data — Fall epidermis samples

The data — Fall epidermis samples

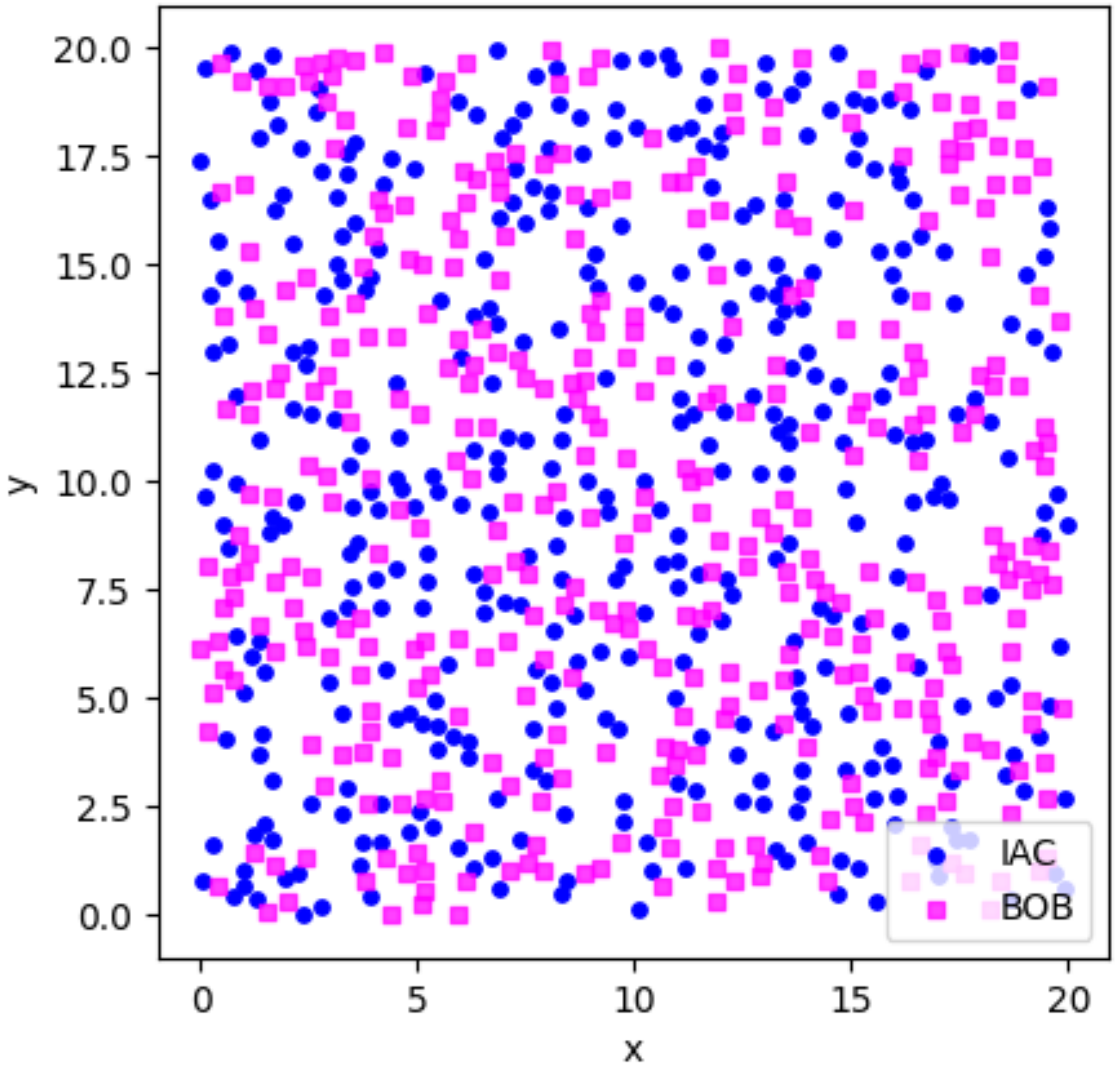
Cold



Medium

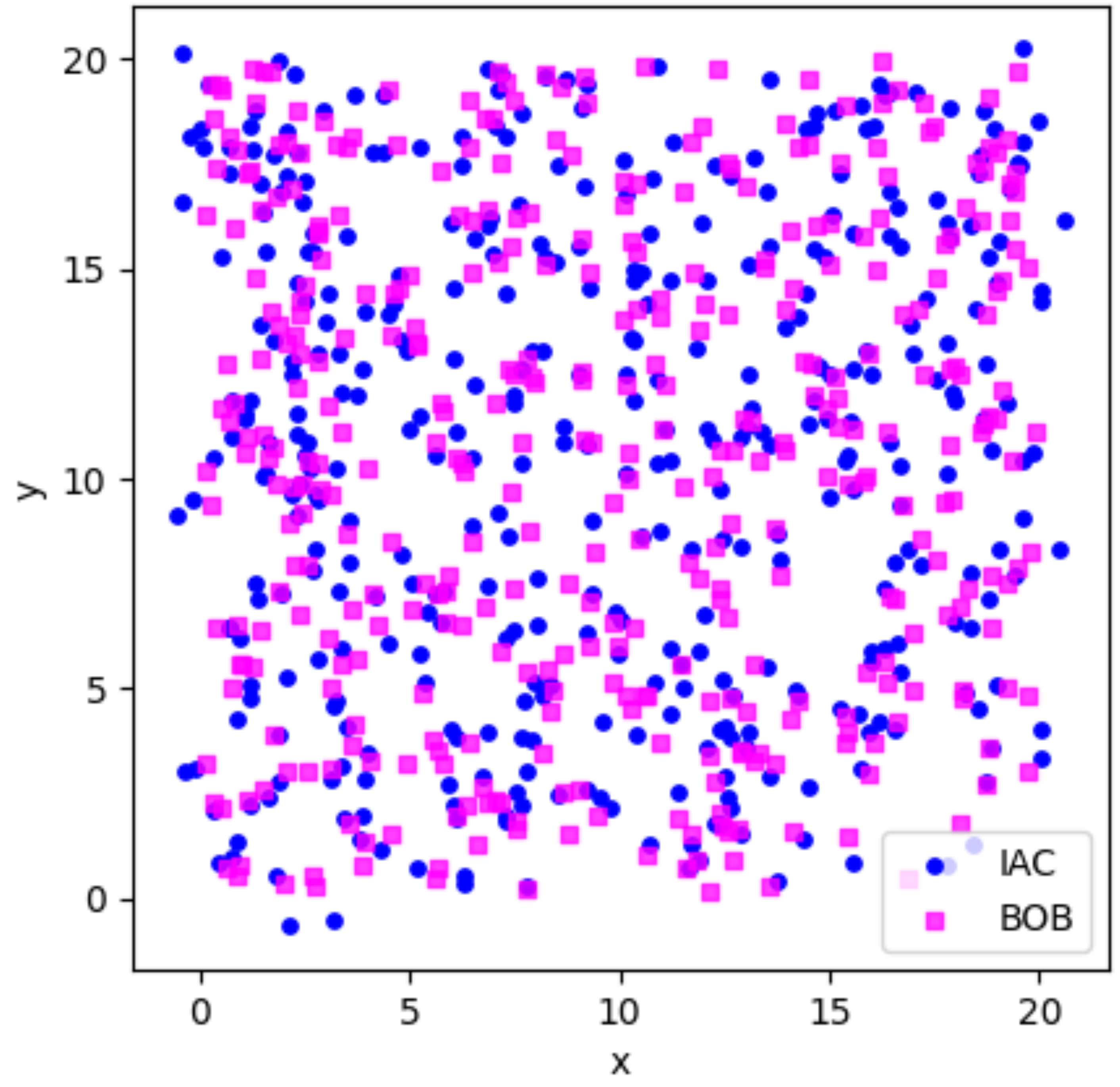


Warm

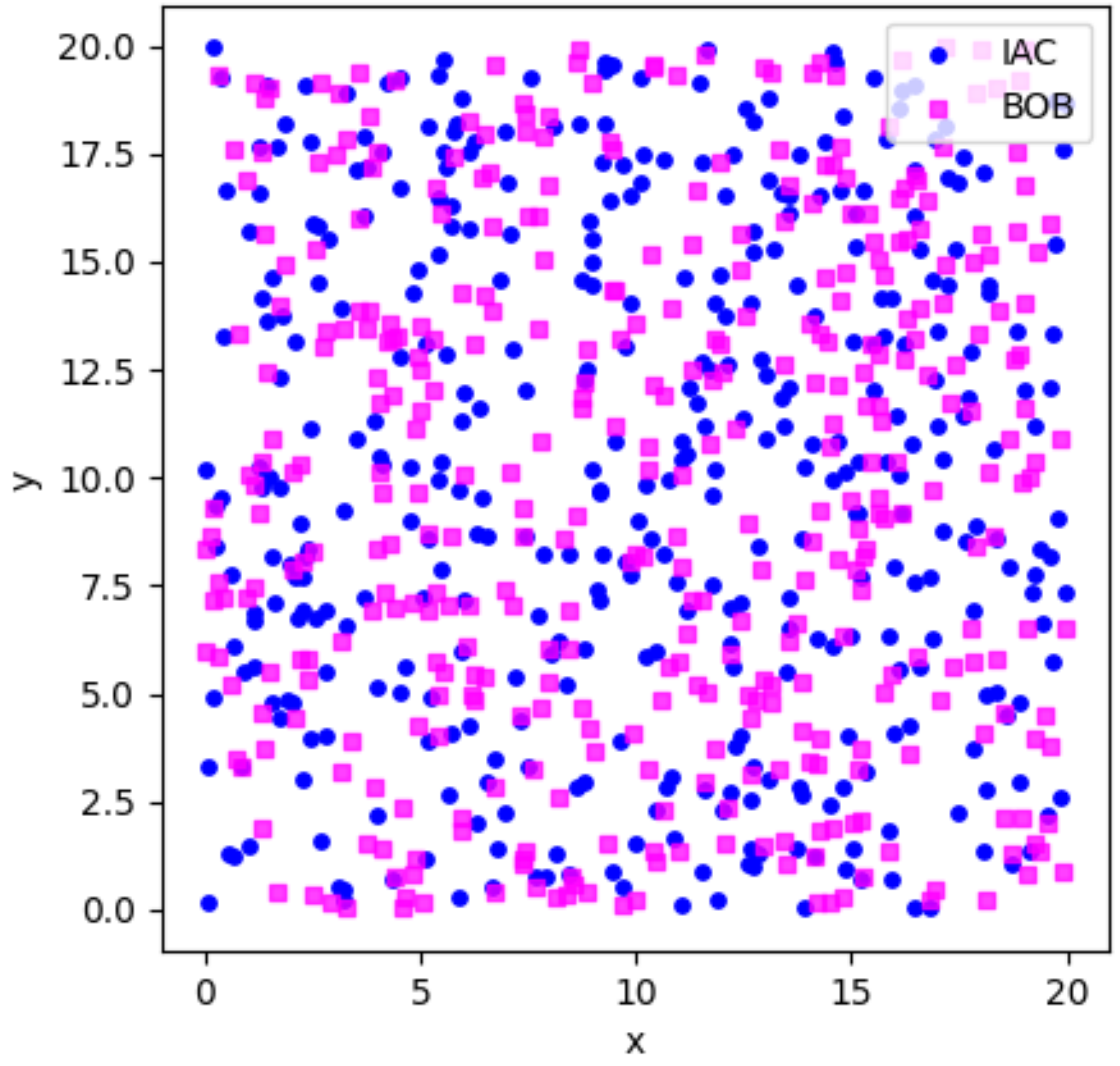


The data — Fall epidermis samples

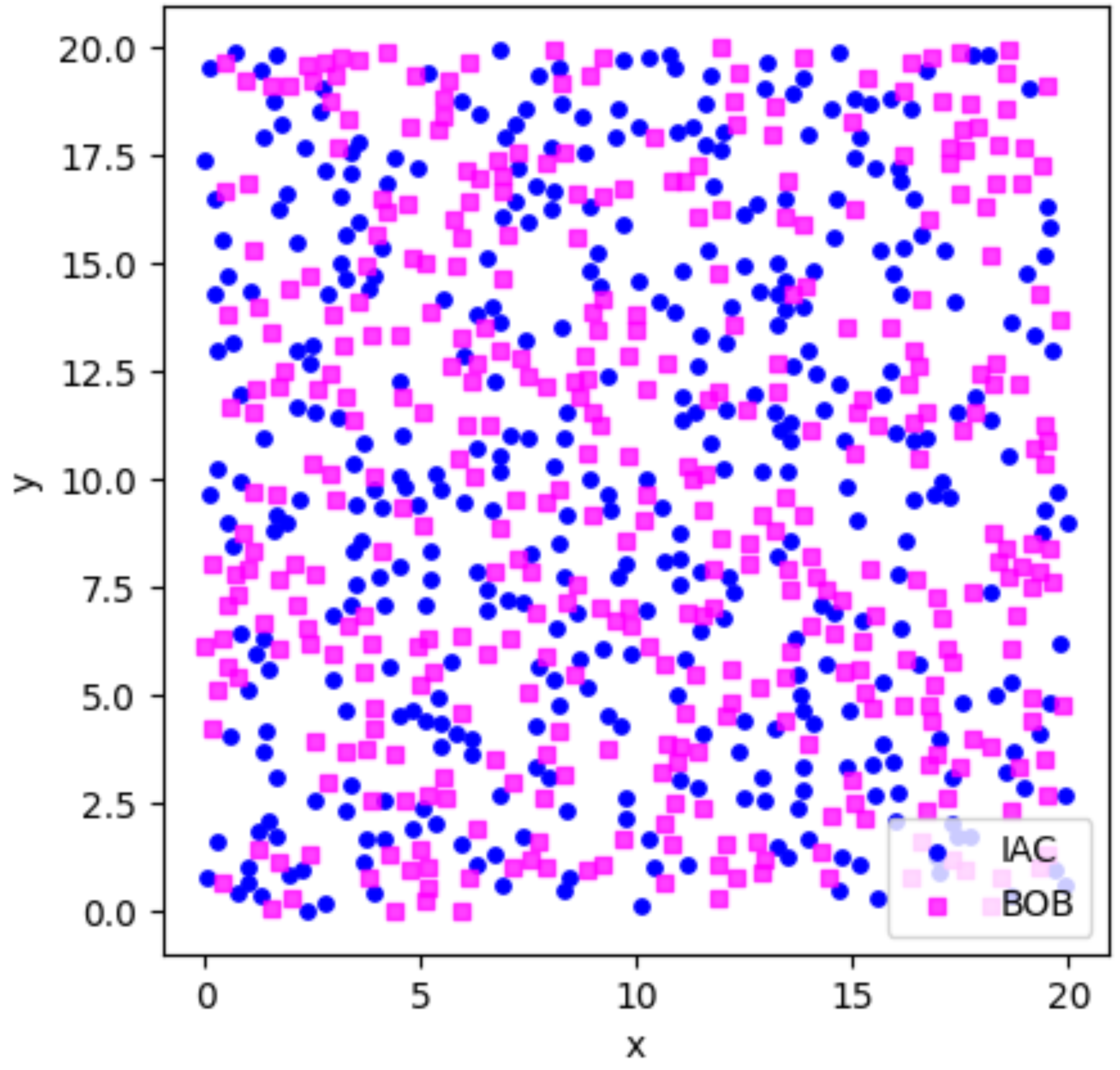
Cold



Medium



Warm



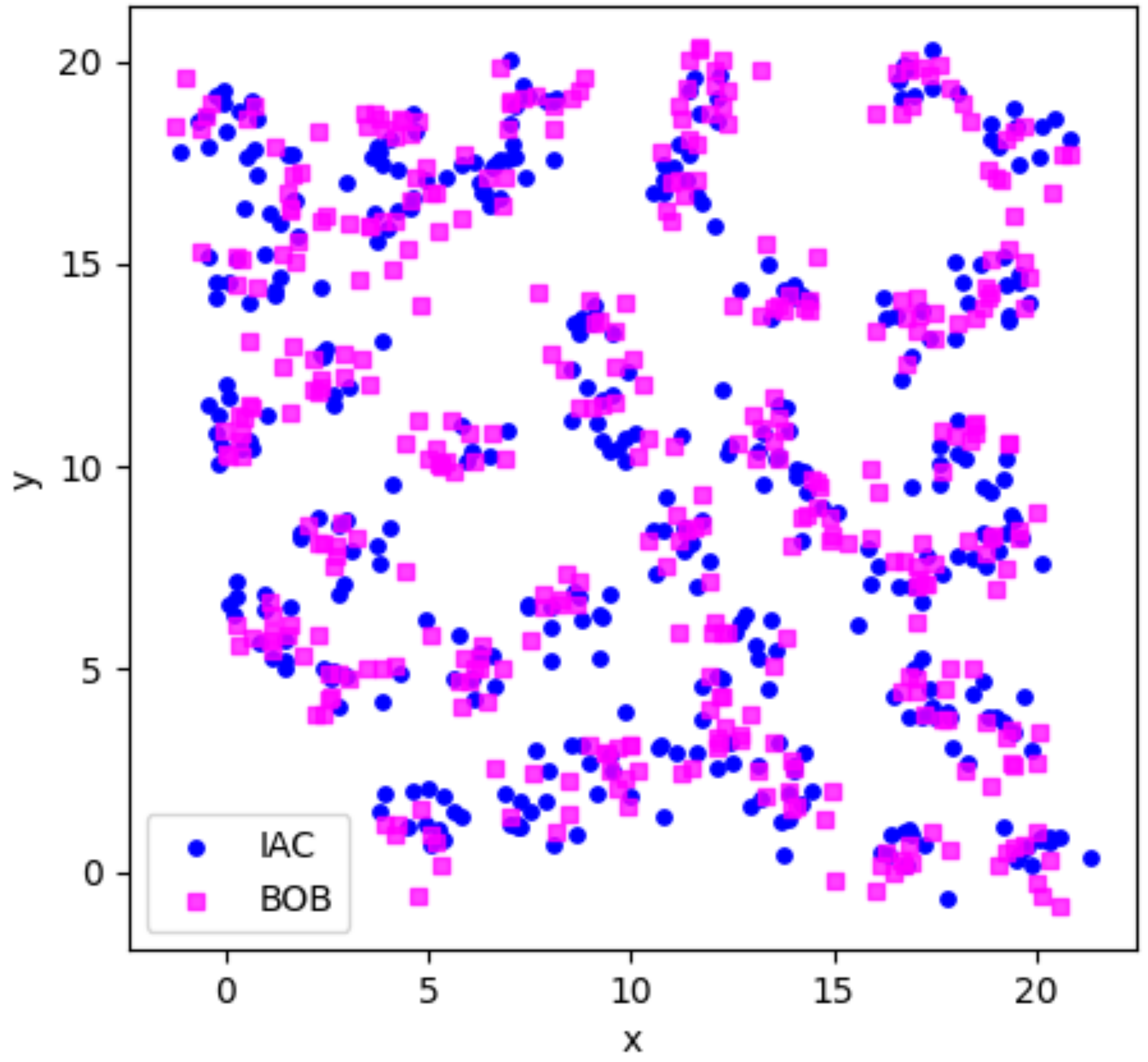
~~Do IAC and BOB attract or repulse each other depending on temperature?~~

Is there an association between attraction and repulsion and temperature?

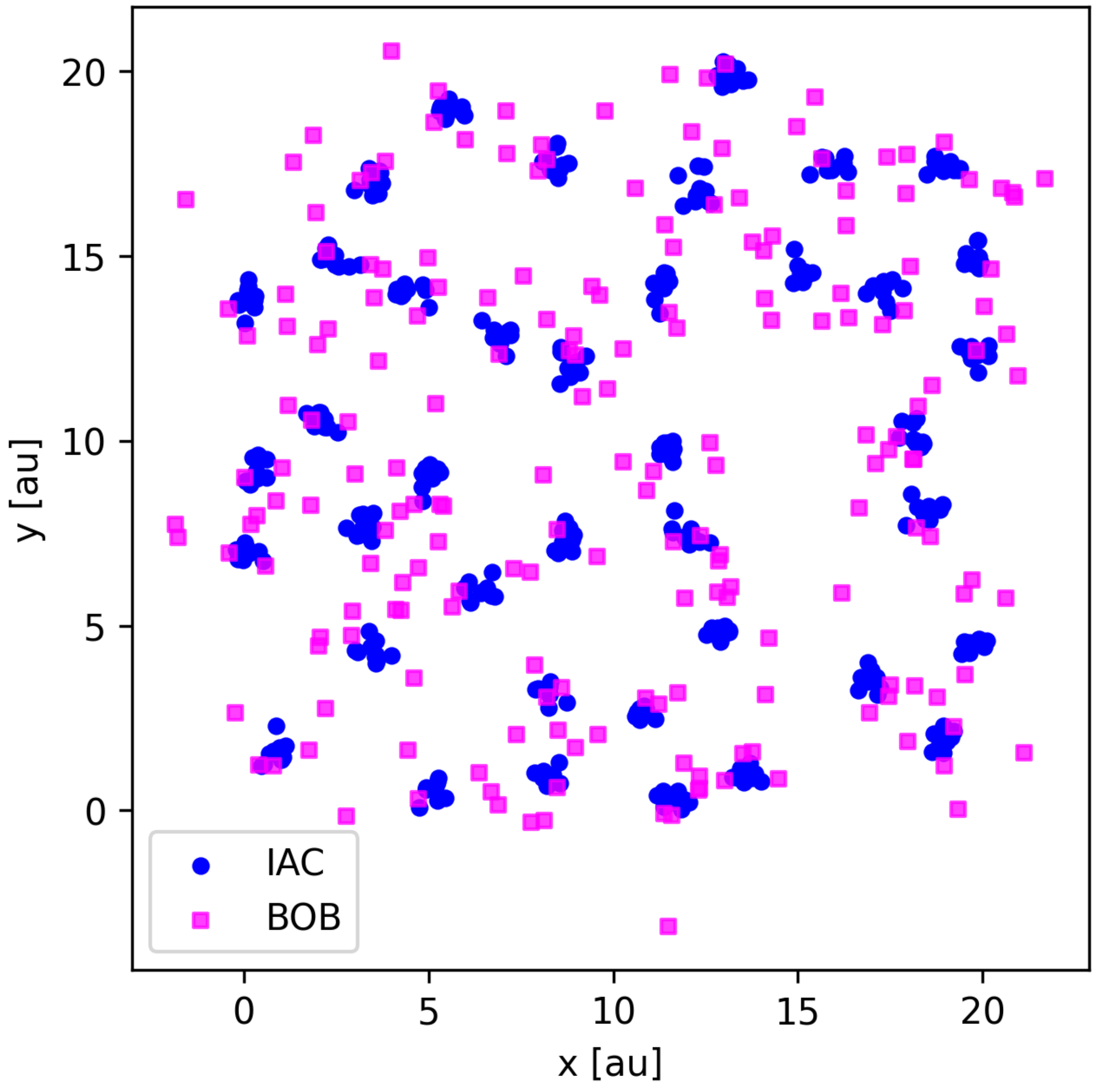
The data — Winter epidermis samples

The data — Winter epidermis samples

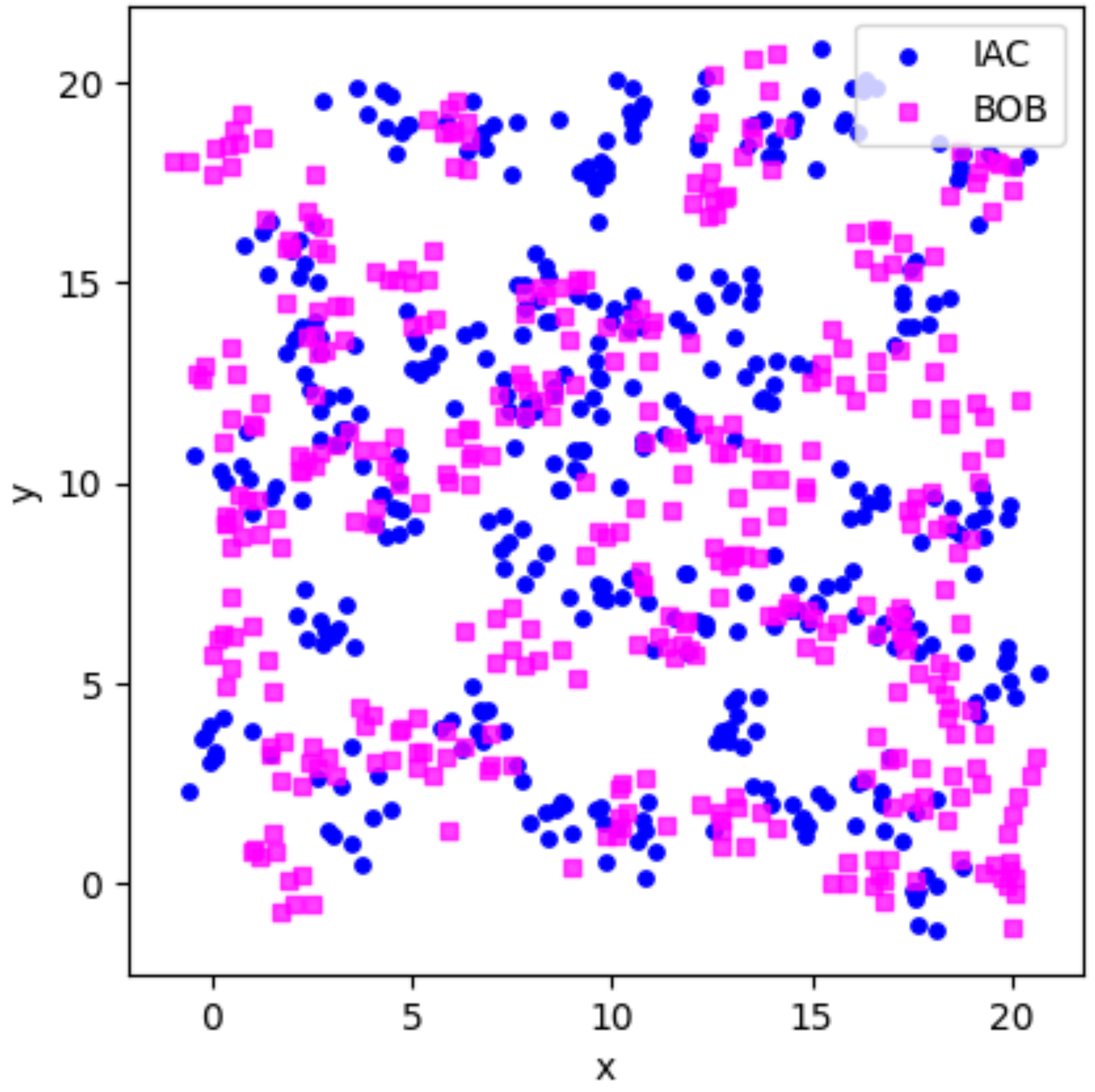
Cold



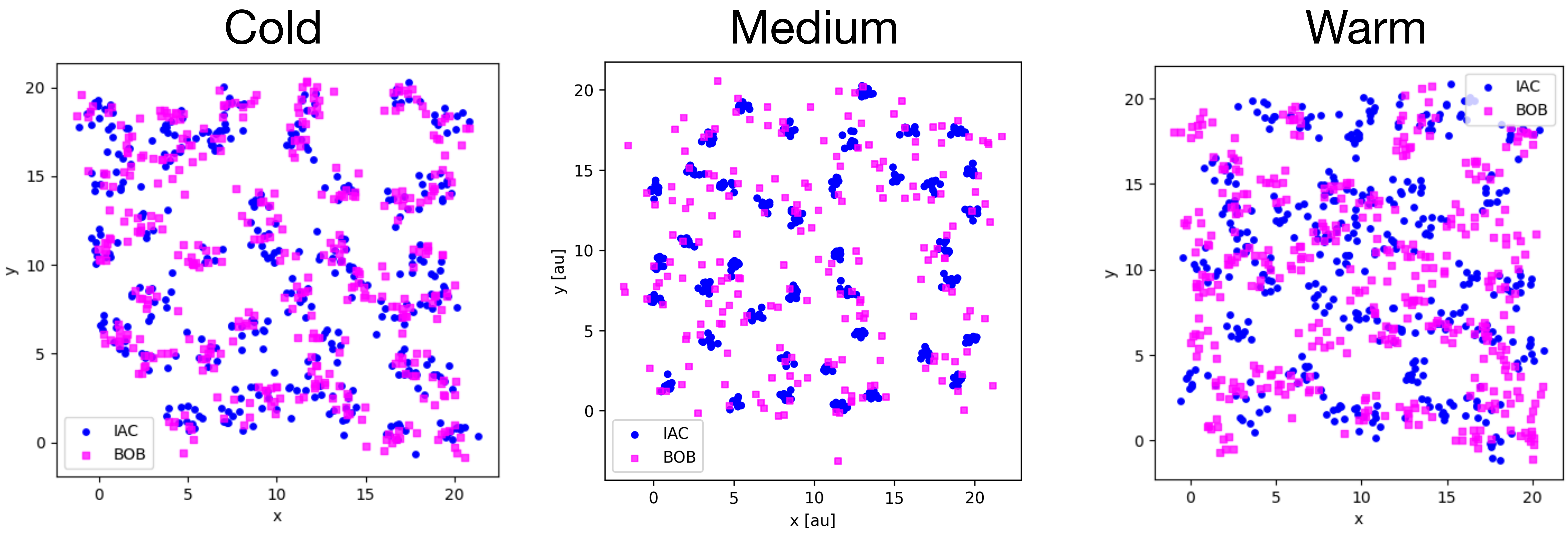
Medium



Warm



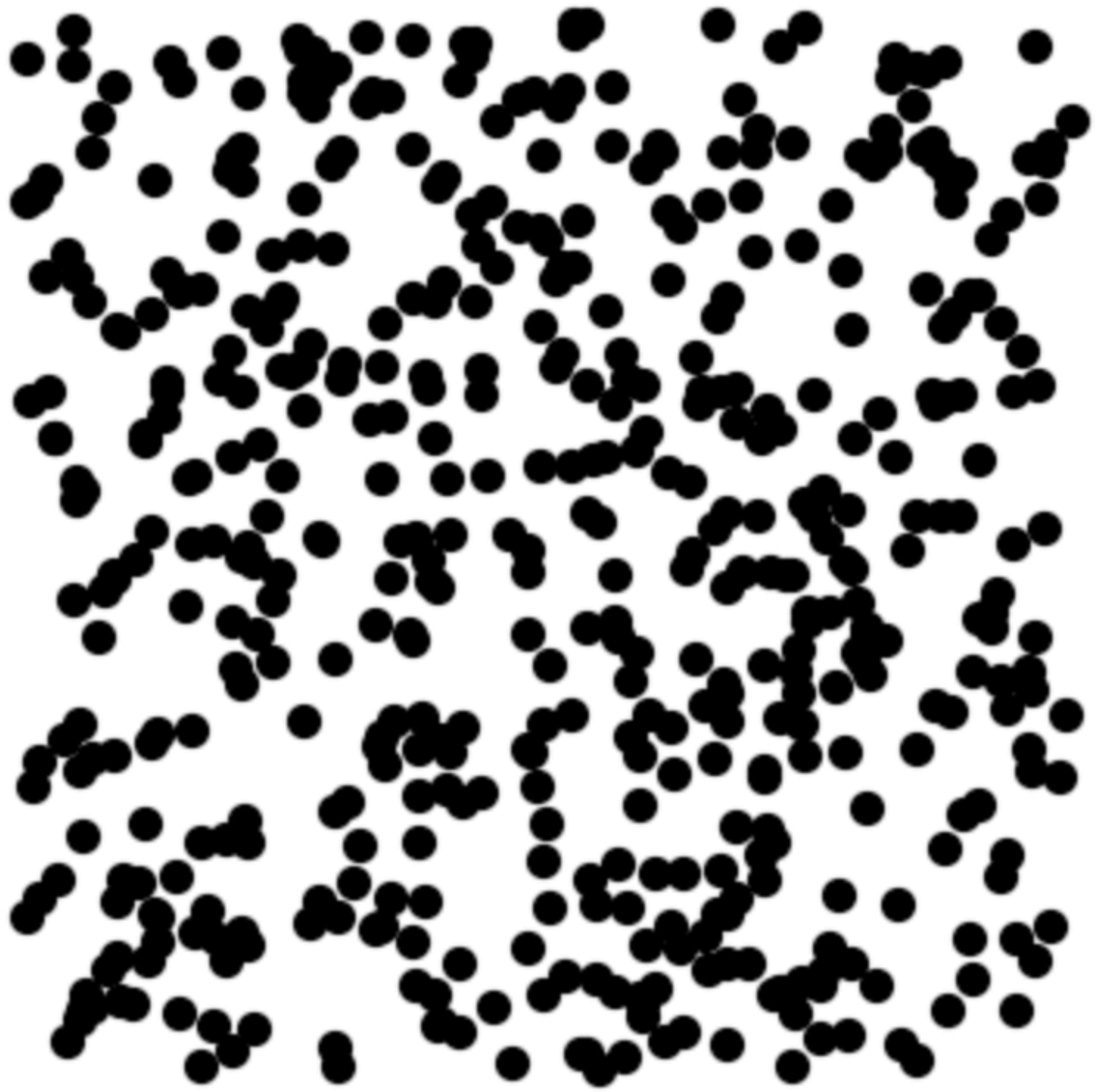
The data — Winter epidermis samples



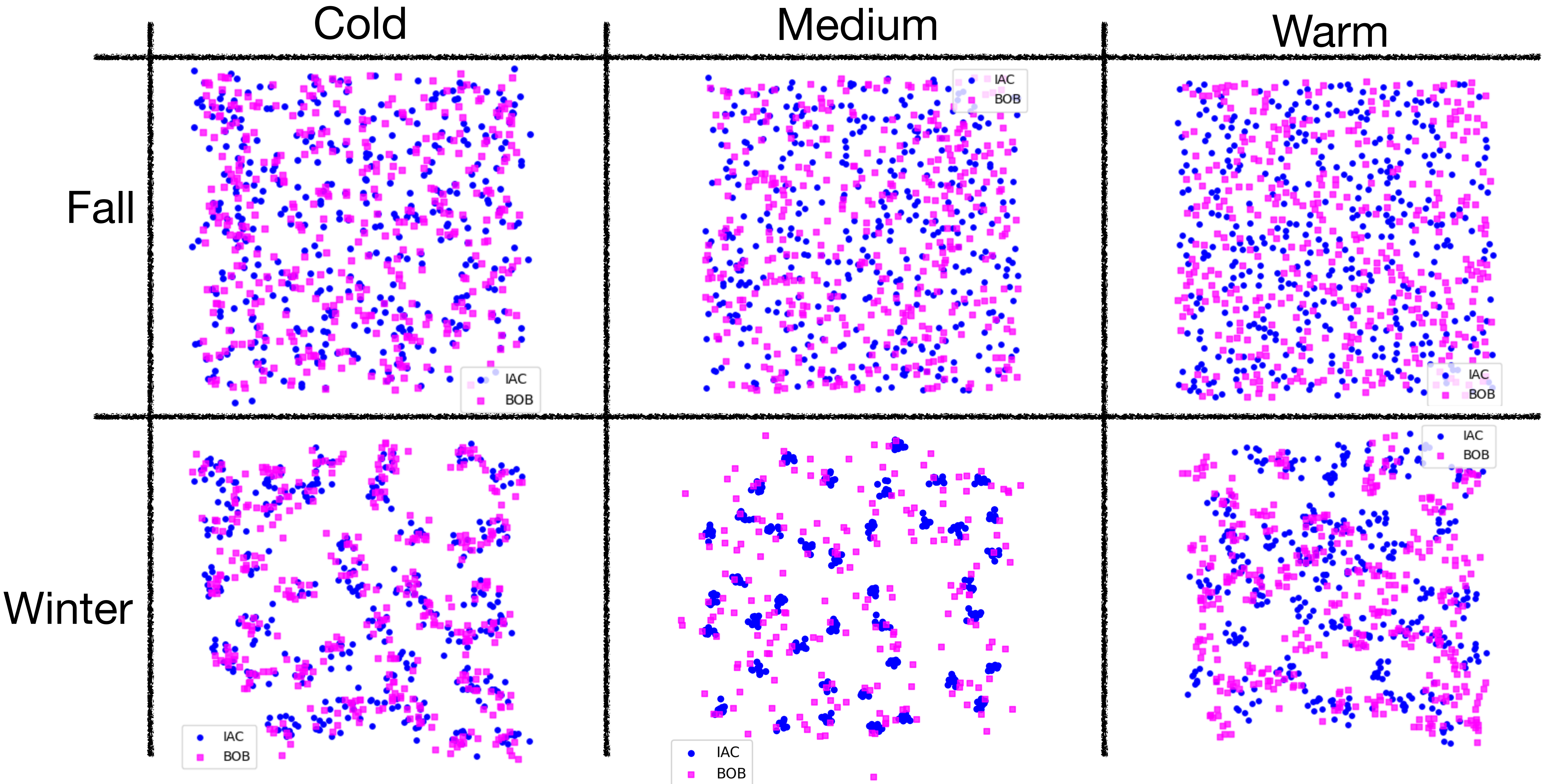
~~Do IAC and BOB attract or repulse each other depending on temperature?~~

Is there an association between attraction and repulsion and temperature?

Q: Do you see patterns?

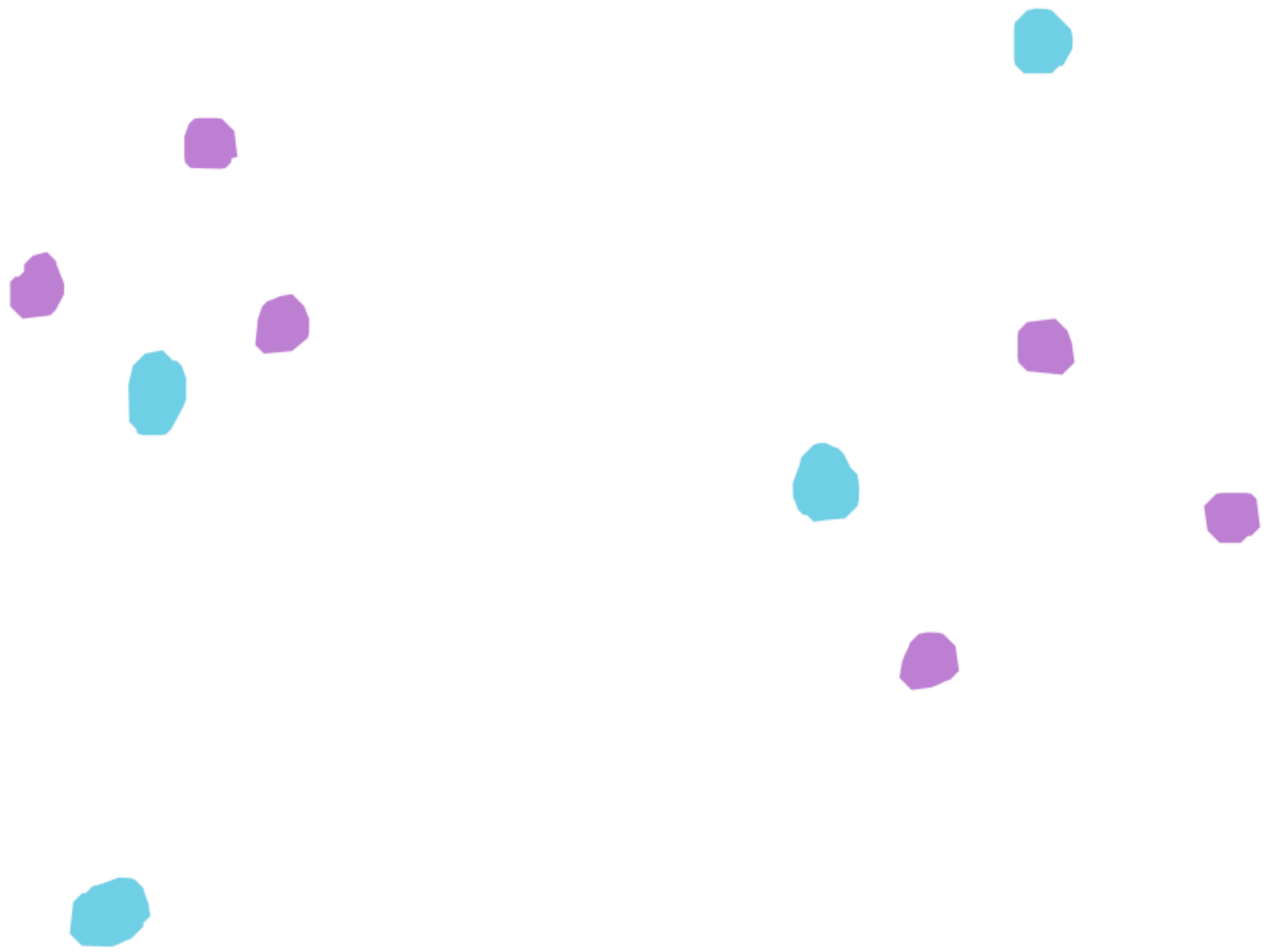


How would you analyze the data?



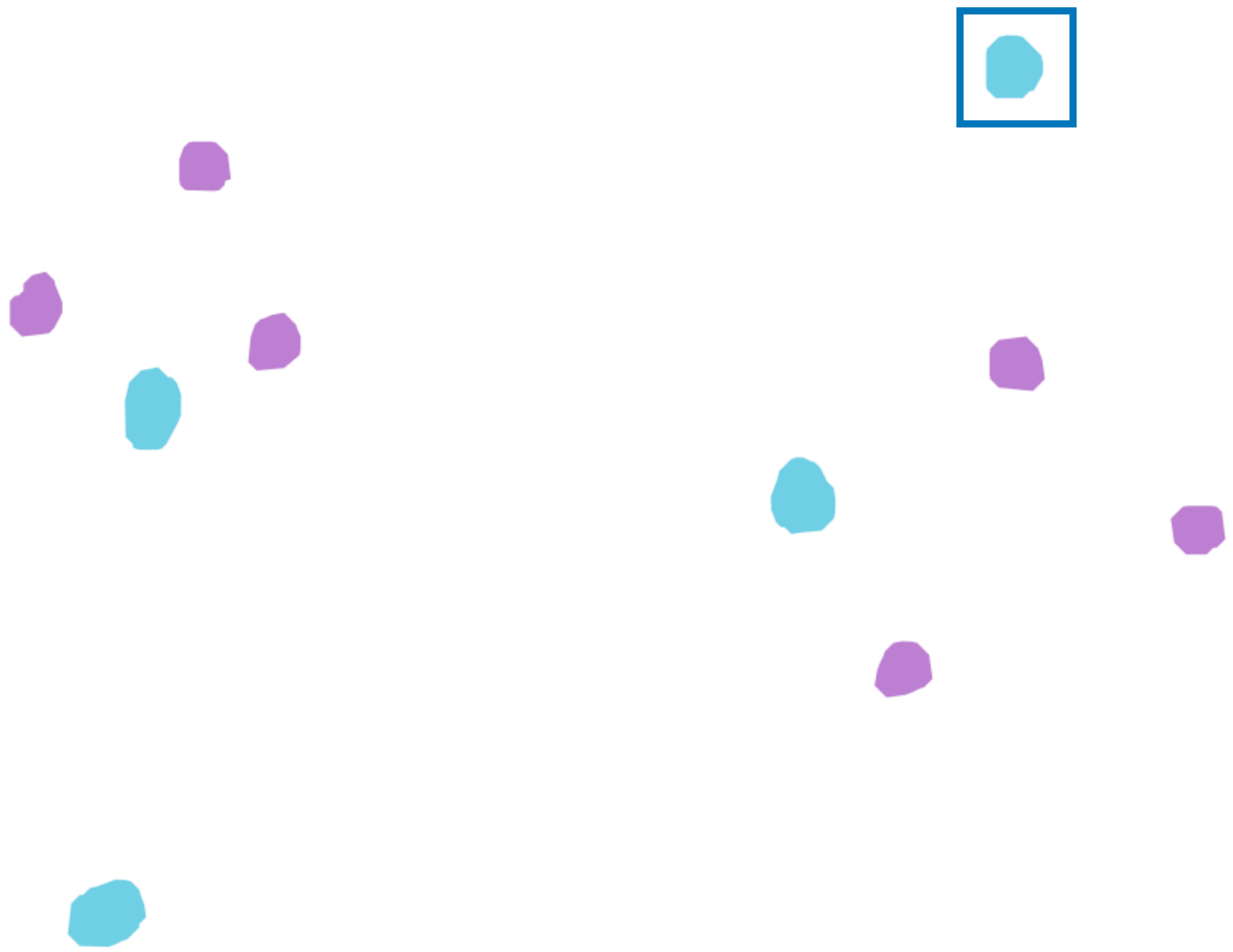
Mean distance to nearest neighbor

IAC → BOB

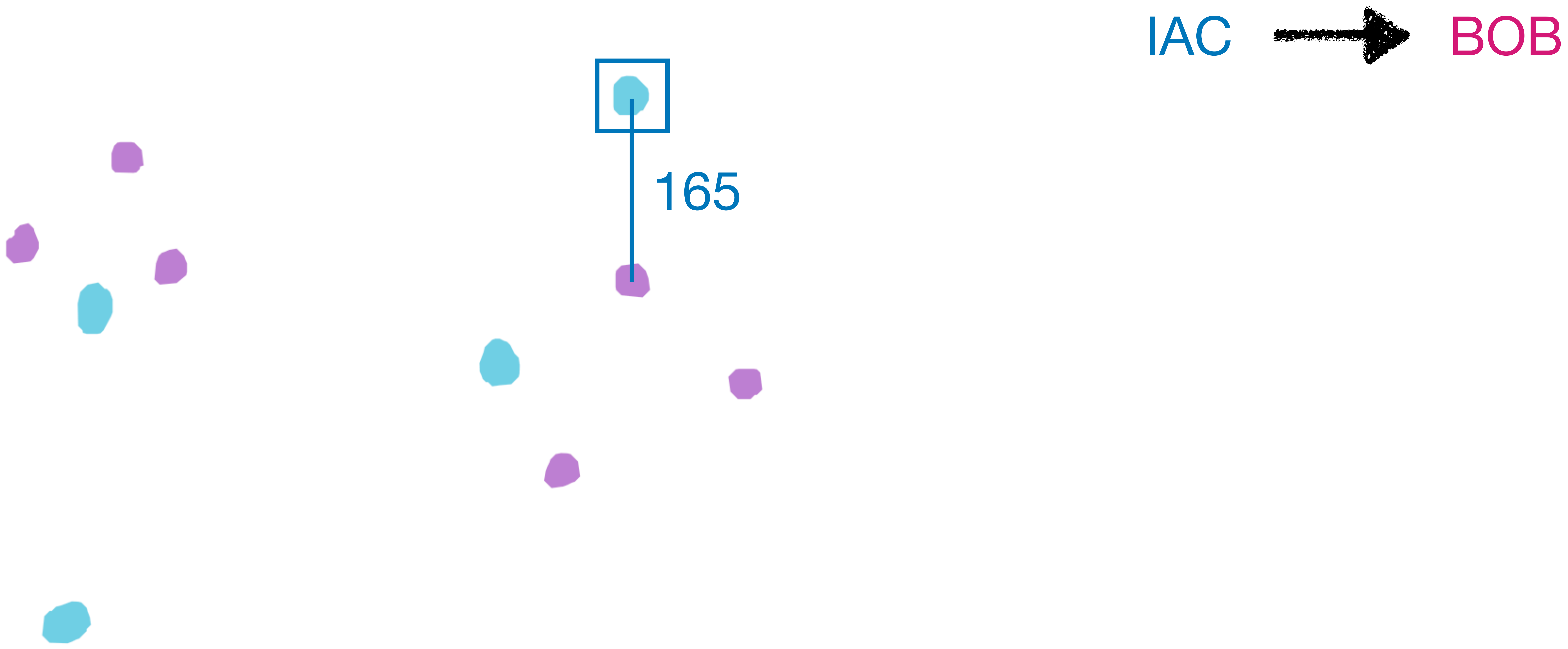


Mean distance to nearest neighbor

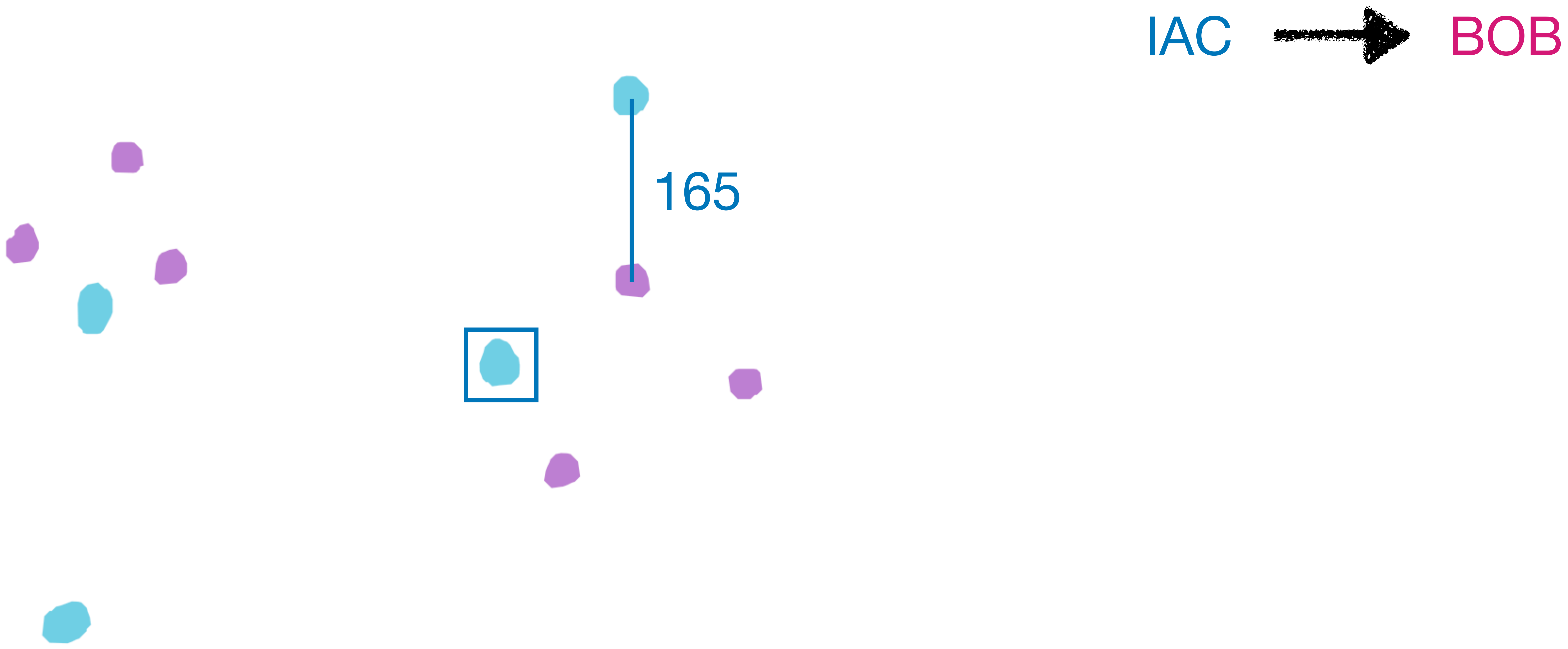
IAC → BOB



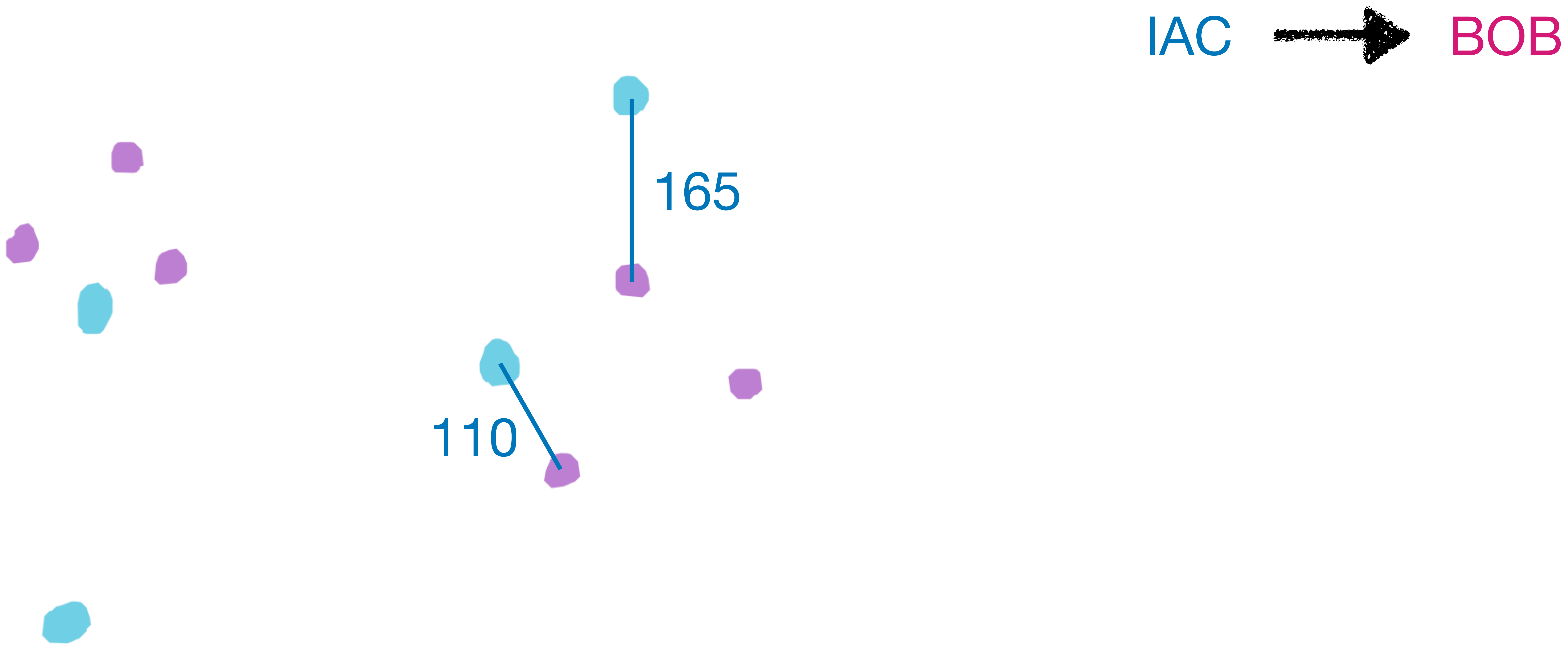
Mean distance to nearest neighbor



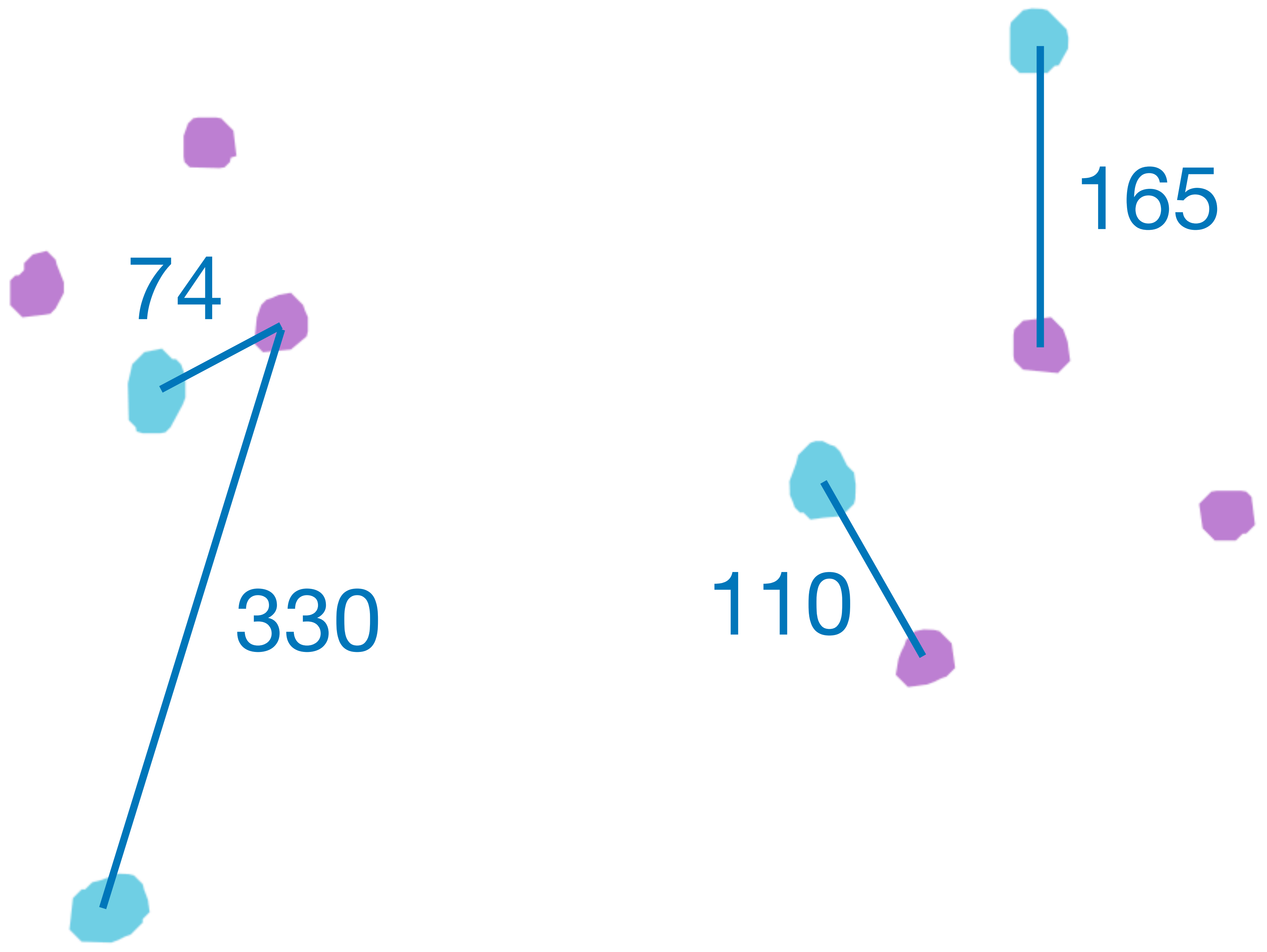
Mean distance to nearest neighbor



Mean distance to nearest neighbor



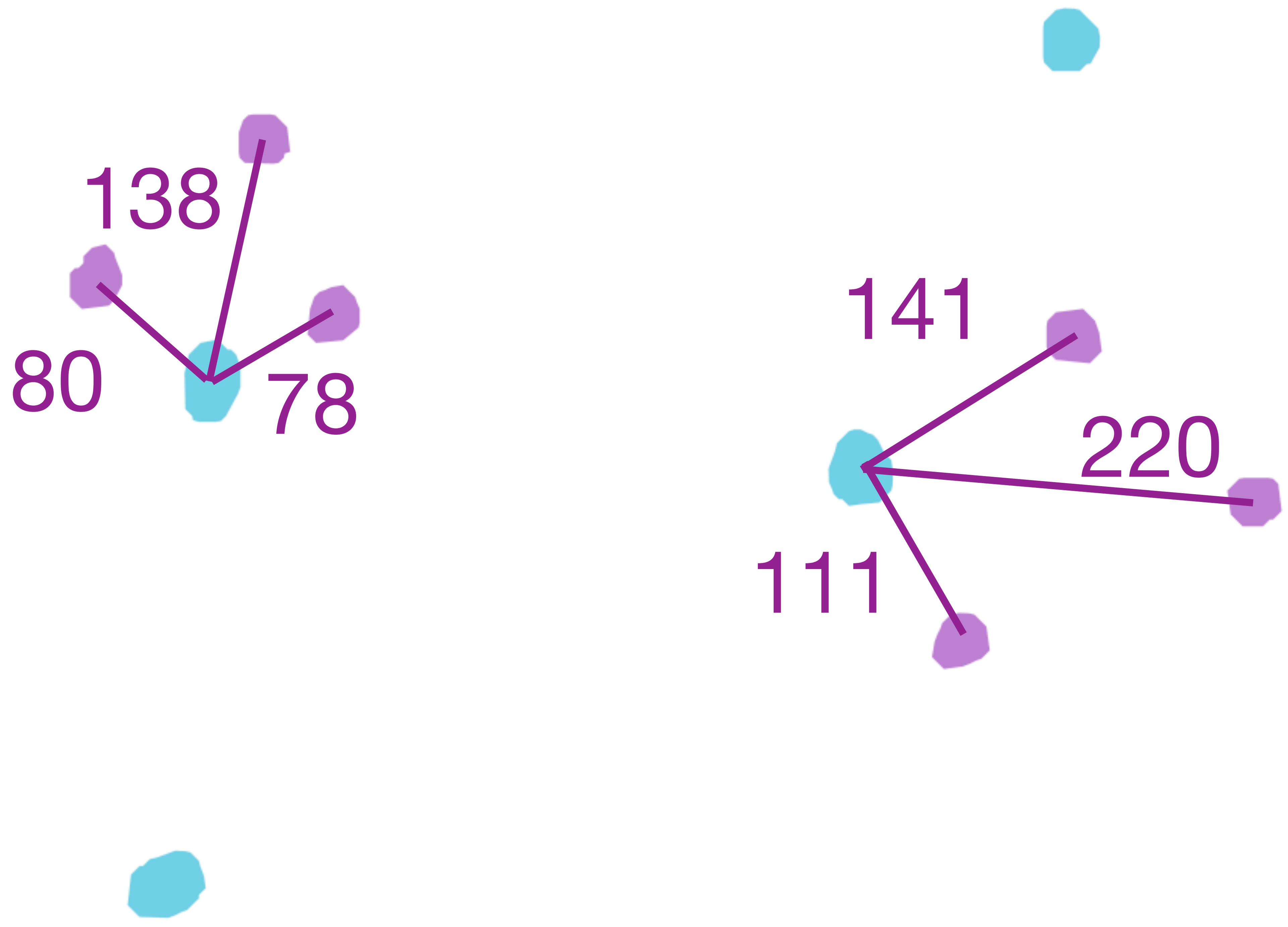
Mean distance to nearest neighbor



IAC → BOB

$$\frac{74 + 330 + 110 + 165}{4} = 169.75$$

Mean distance to nearest neighbor



IAC → BOB

$$\frac{74 + 330 + 110 + 165}{4} = 169.75$$

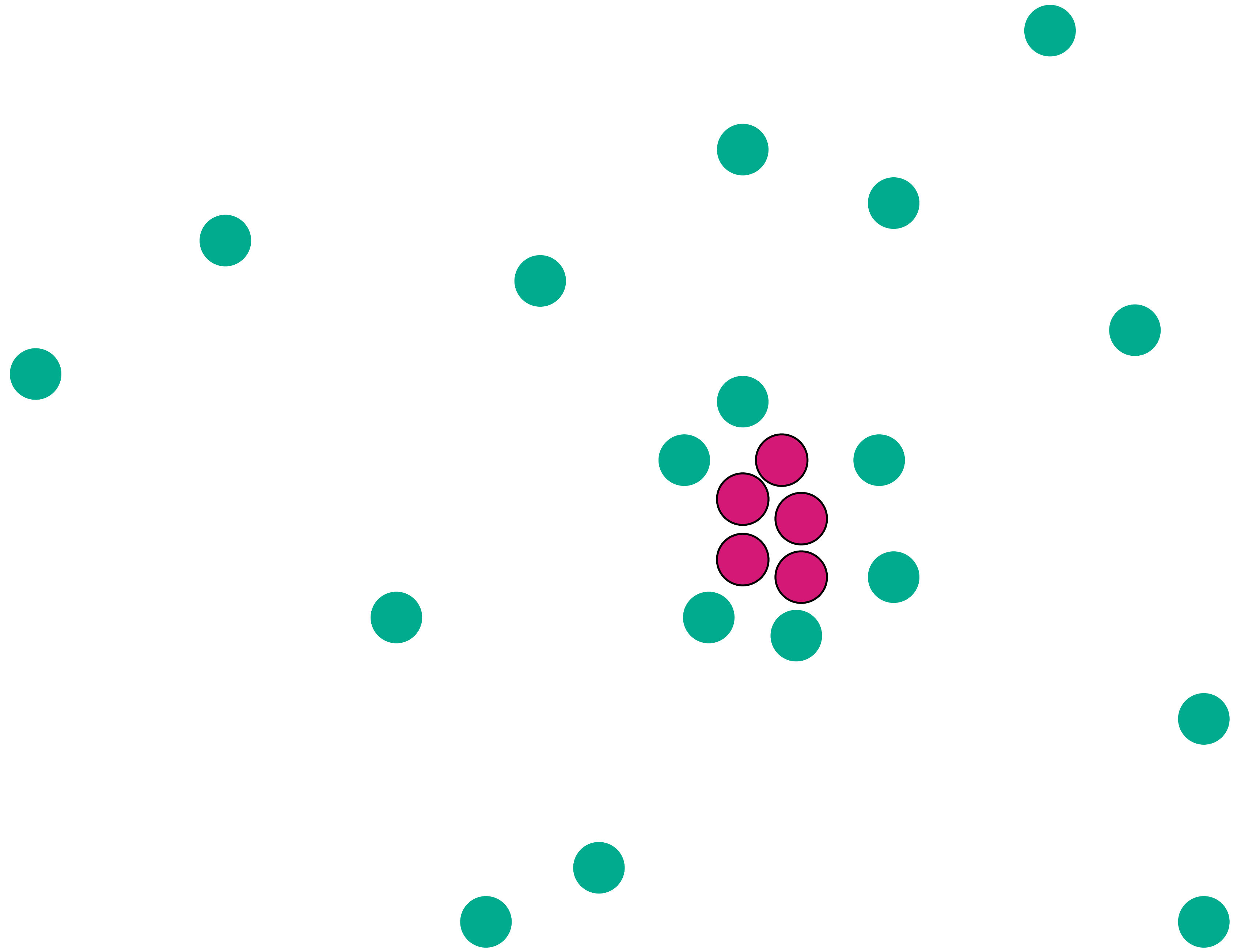
BOB → IAC

$$\boxed{\frac{80+138+78+111+141+220}{6} = 121}$$

Mean distance to nearest neighbor

= Small mean distance

= Large mean distance



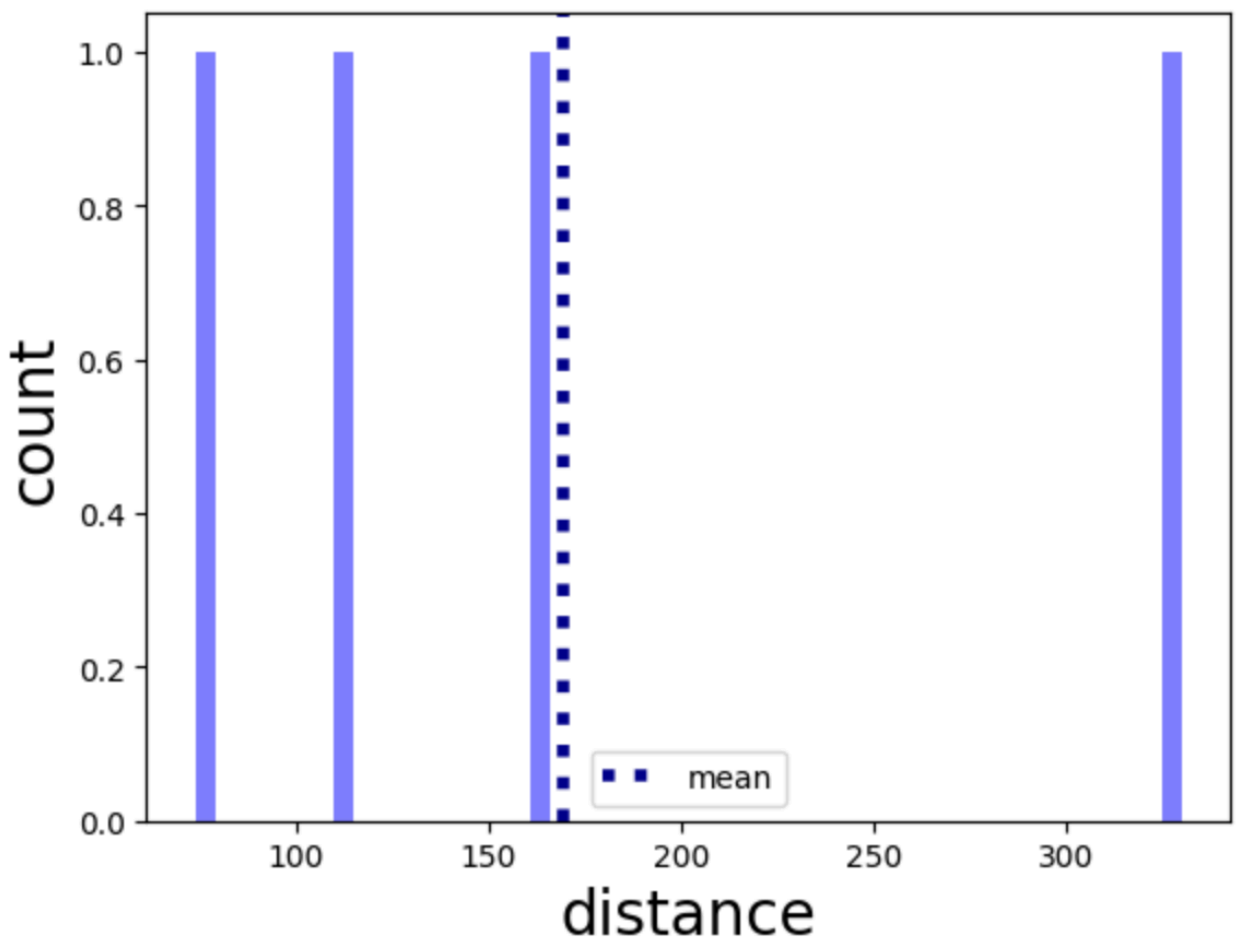
Mean distance to nearest neighbor

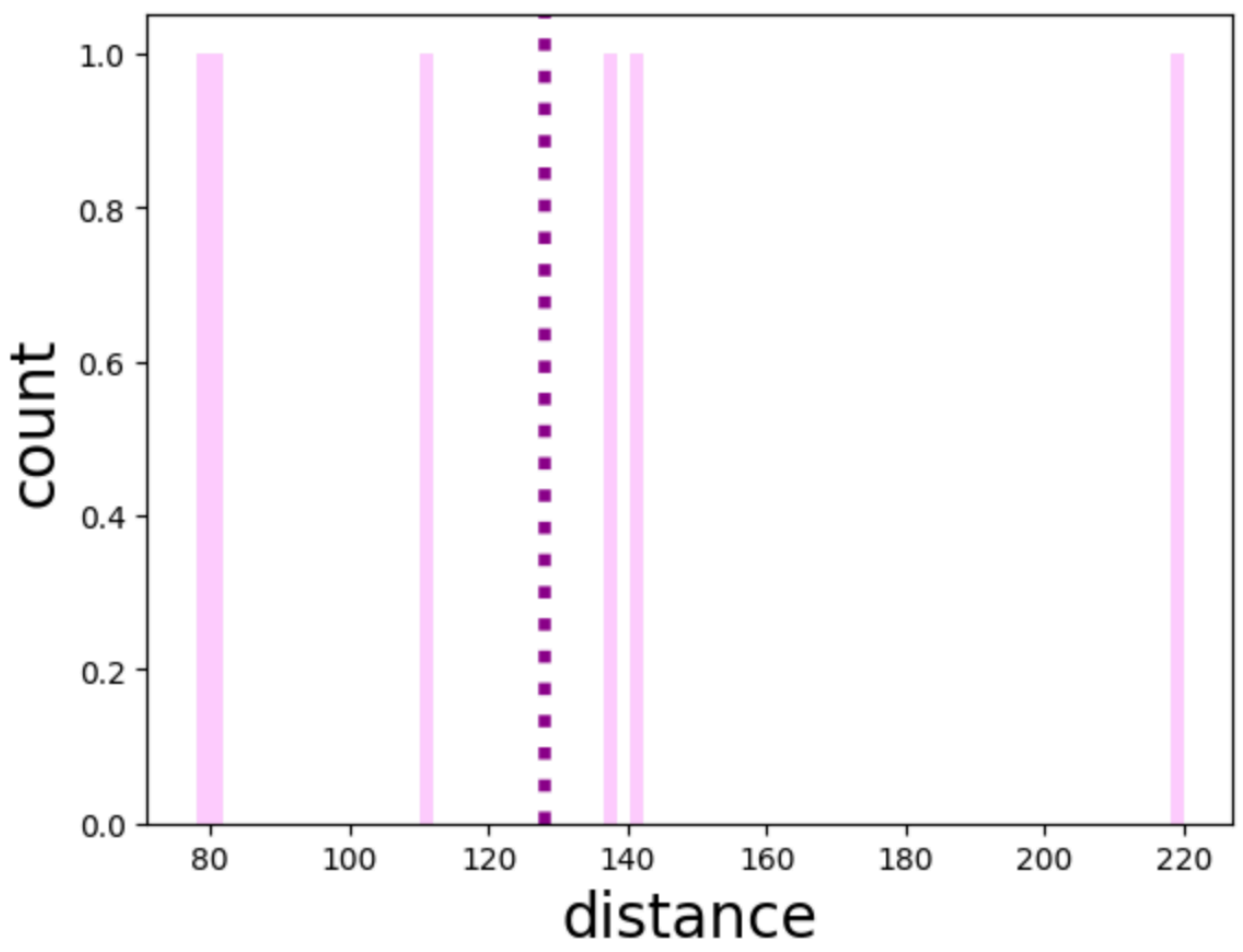
IAC → BOB

Distances	Mean
74, 330, 110, 165	169.75

BOB → IAC

Distances	Mean
80, 138, 78, 111, 141, 220	121

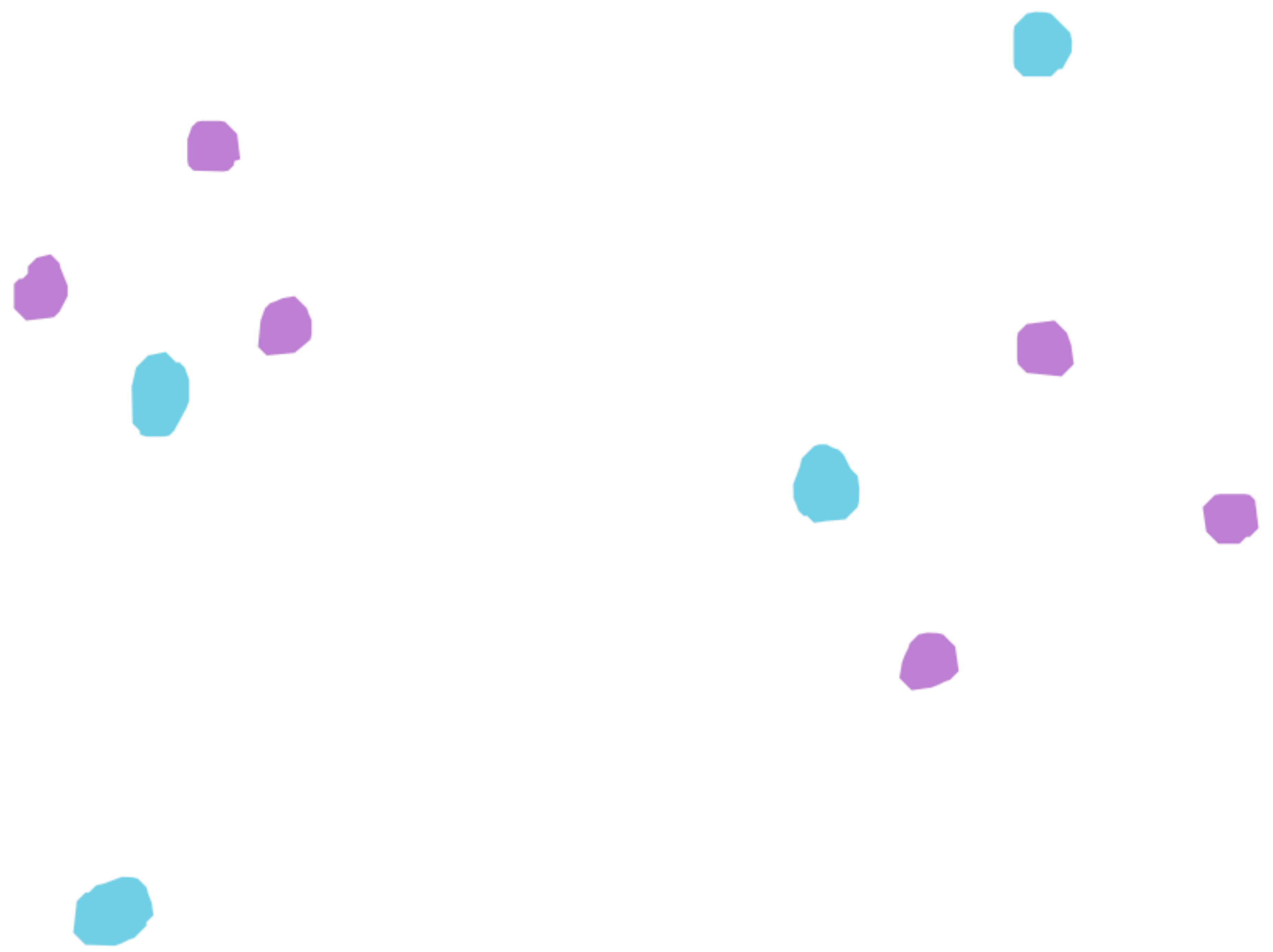




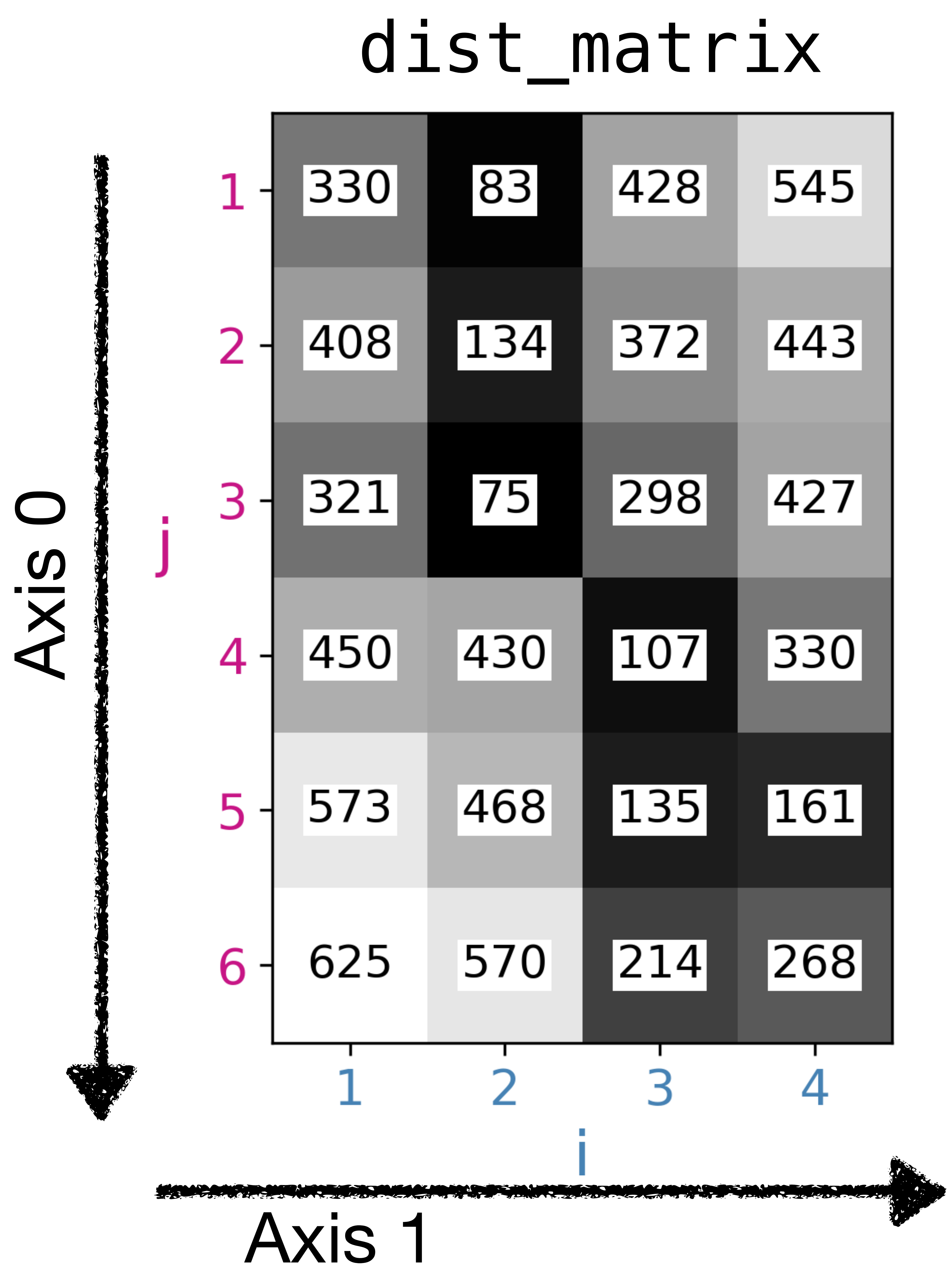
Exercise: Code along: Mean nearest neighbor distance

`dist_matrix`

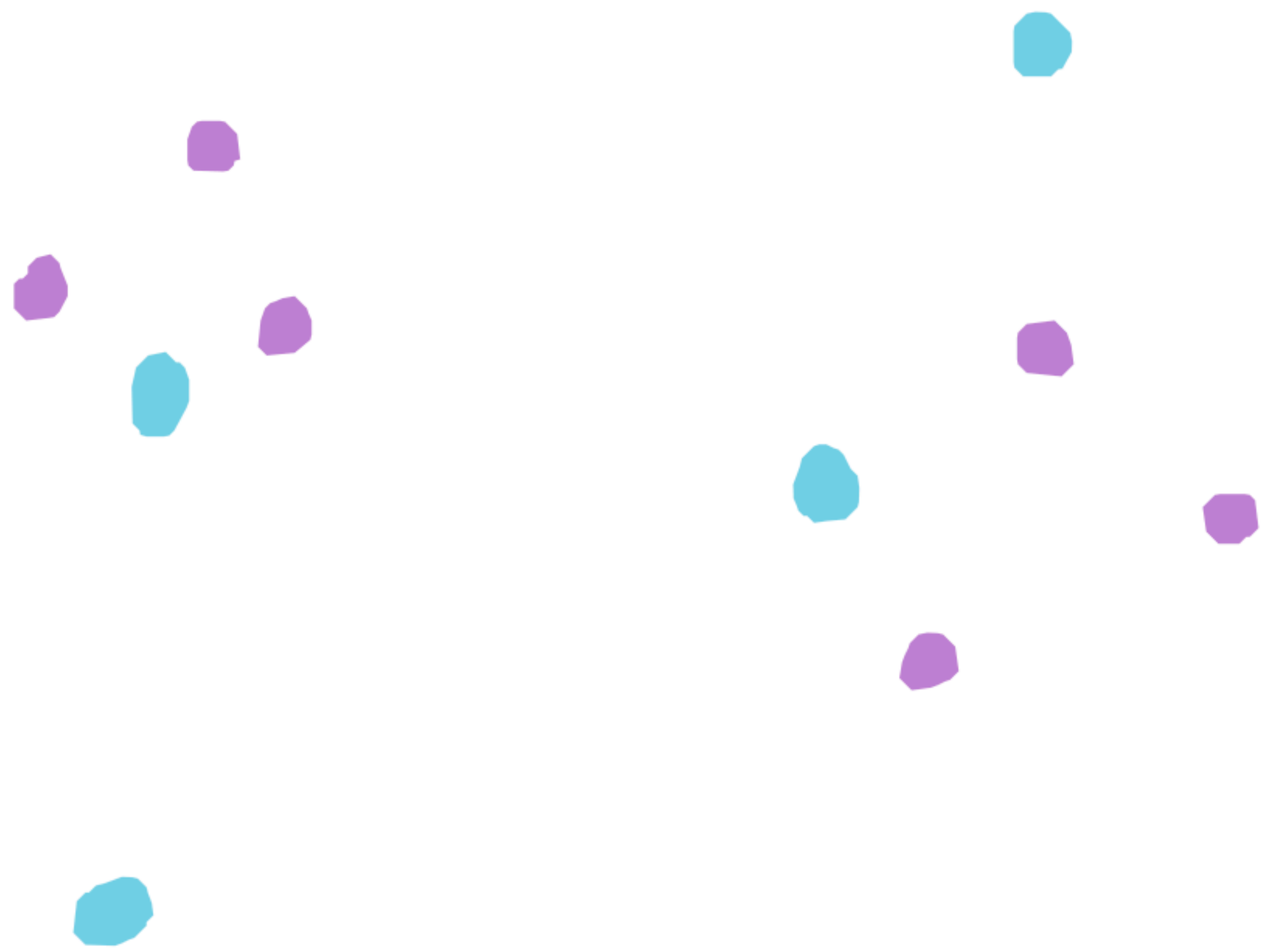
j \ i	1	2	3	4
1	330	83	428	545
2	408	134	372	443
3	321	75	298	427
4	450	430	107	330
5	573	468	135	161
6	625	570	214	268



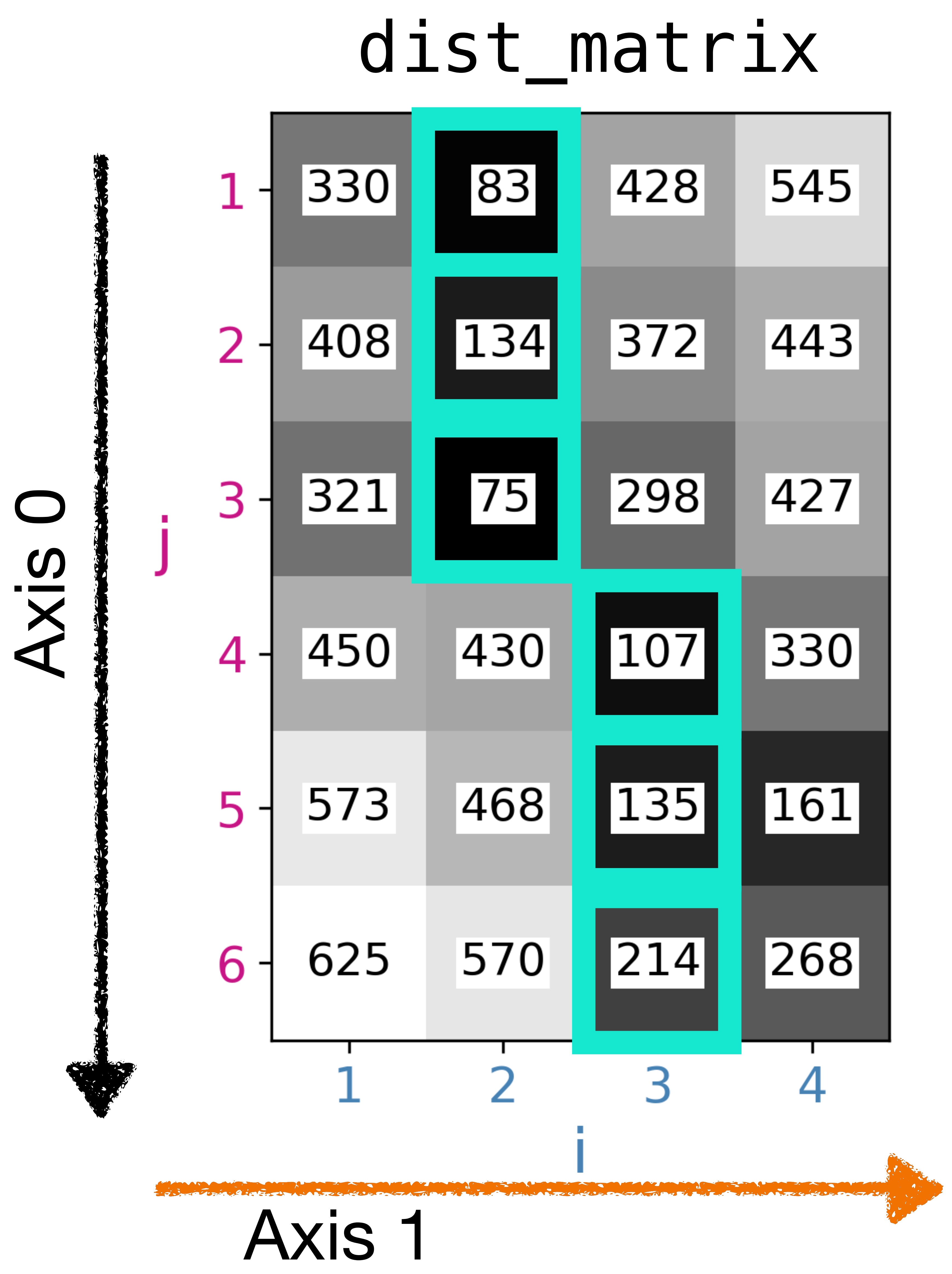
Exercise: Code along: Mean nearest neighbor distance



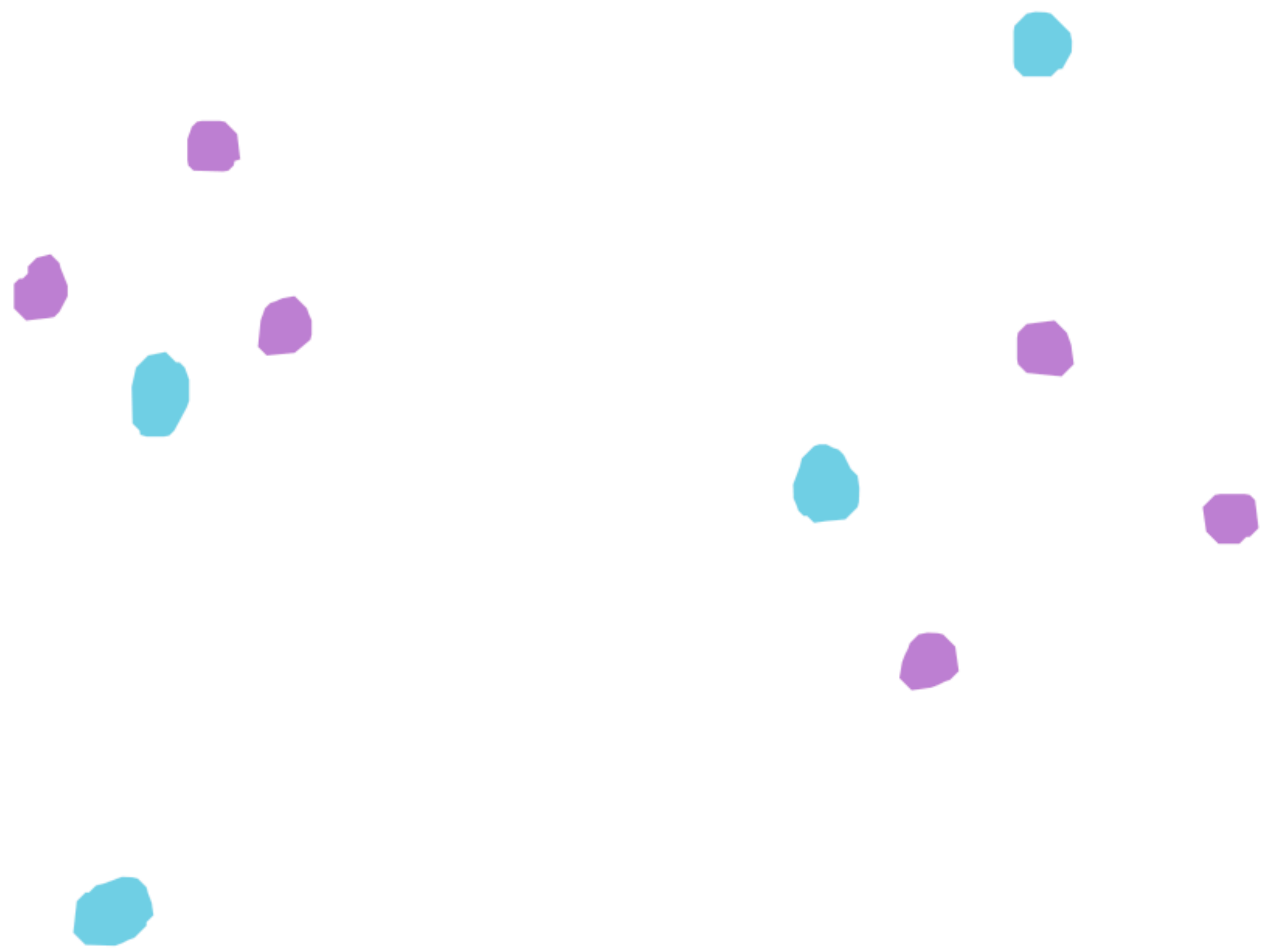

```
np.min(dist_matrix, axis = 1)
```



Exercise: Code along: Mean nearest neighbor distance




```
np.min(dist_matrix, axis = 1)
```



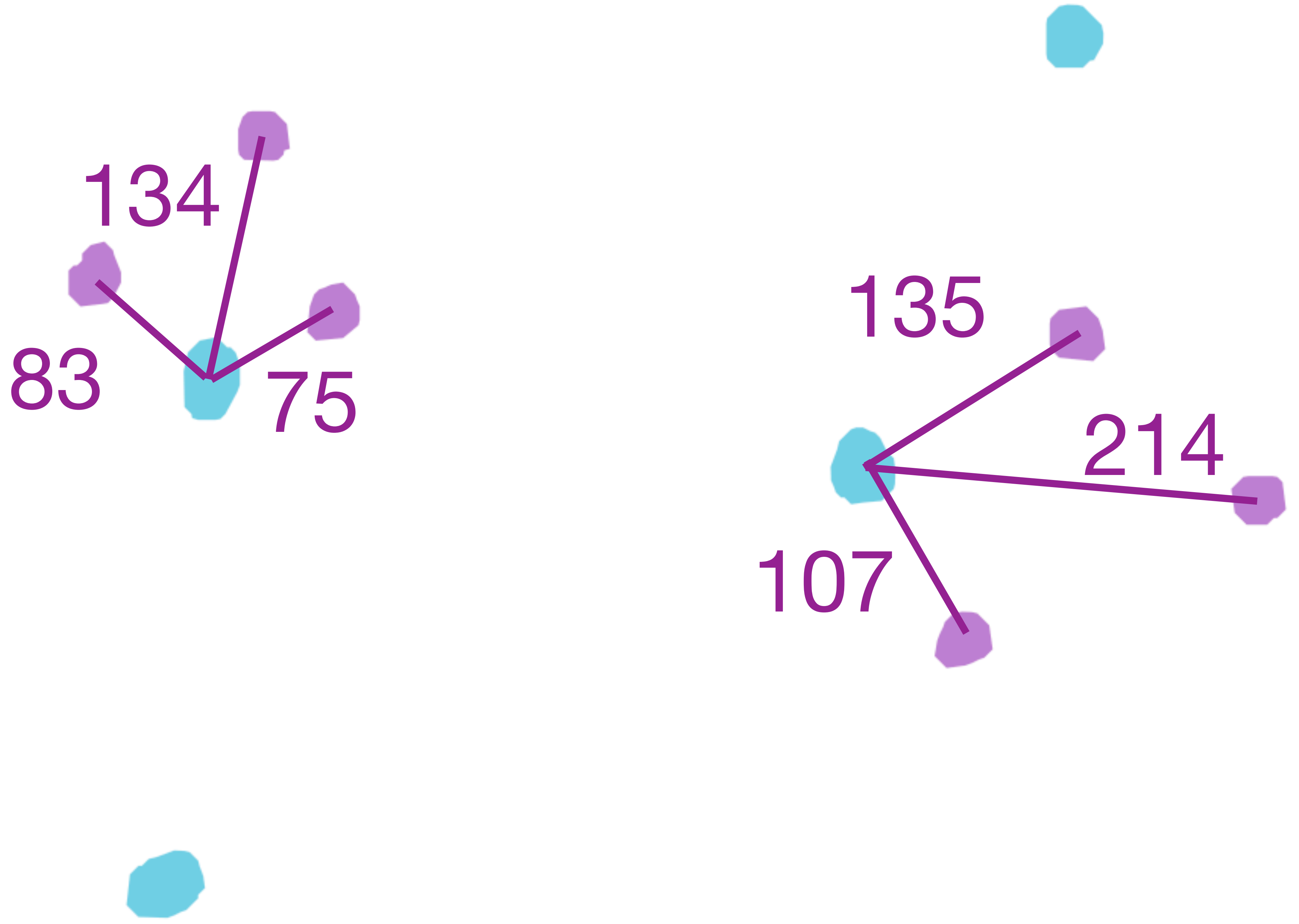
Exercise: Code along: Mean nearest neighbor distance

`dist_matrix`

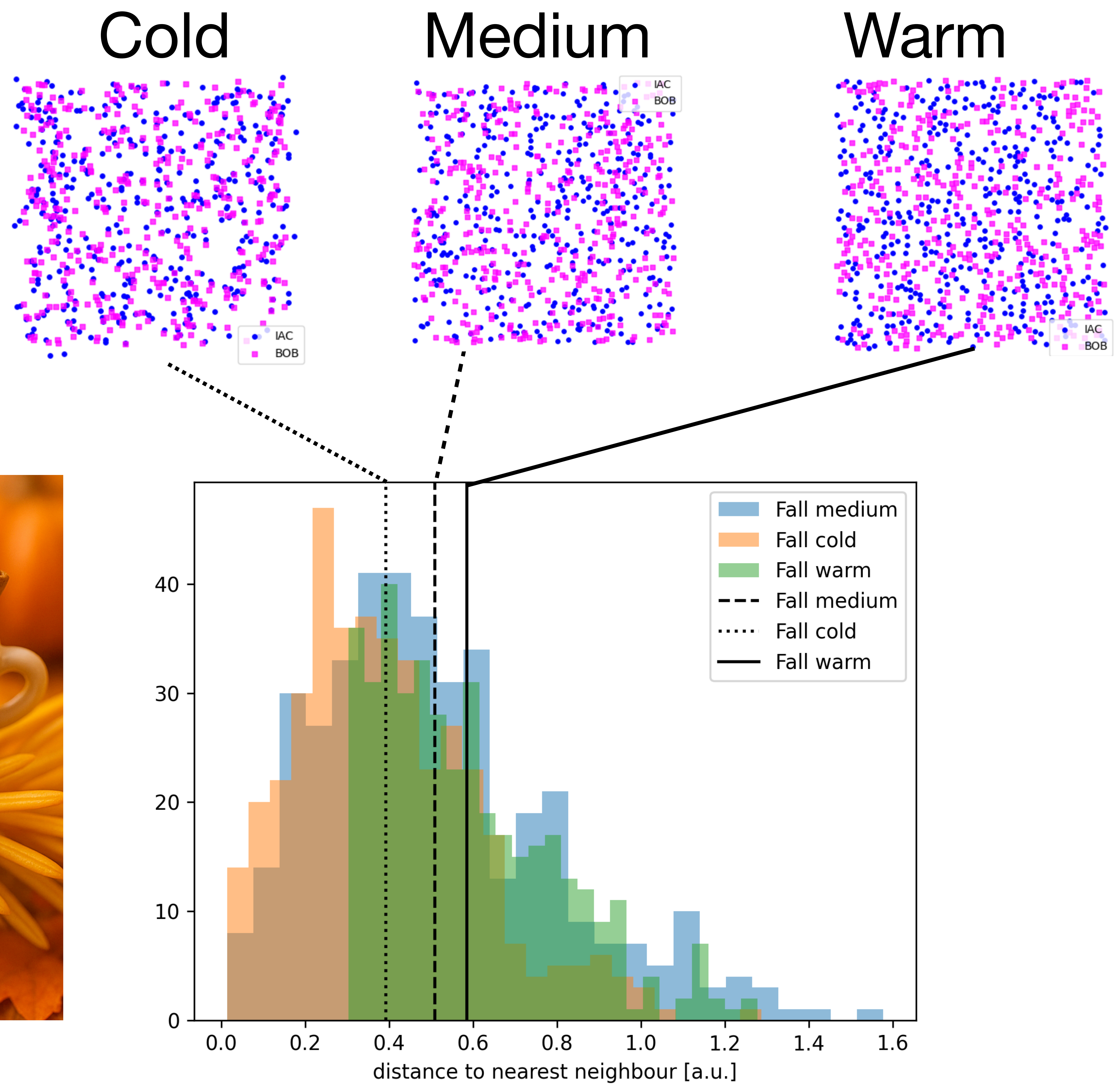
Axis 0 (j) ↓, Axis 1 (i) →

j \ i	1	2	3	4
1	330	83	428	545
2	408	134	372	443
3	321	75	298	427
4	450	430	107	330
5	573	468	135	161
6	625	570	214	268

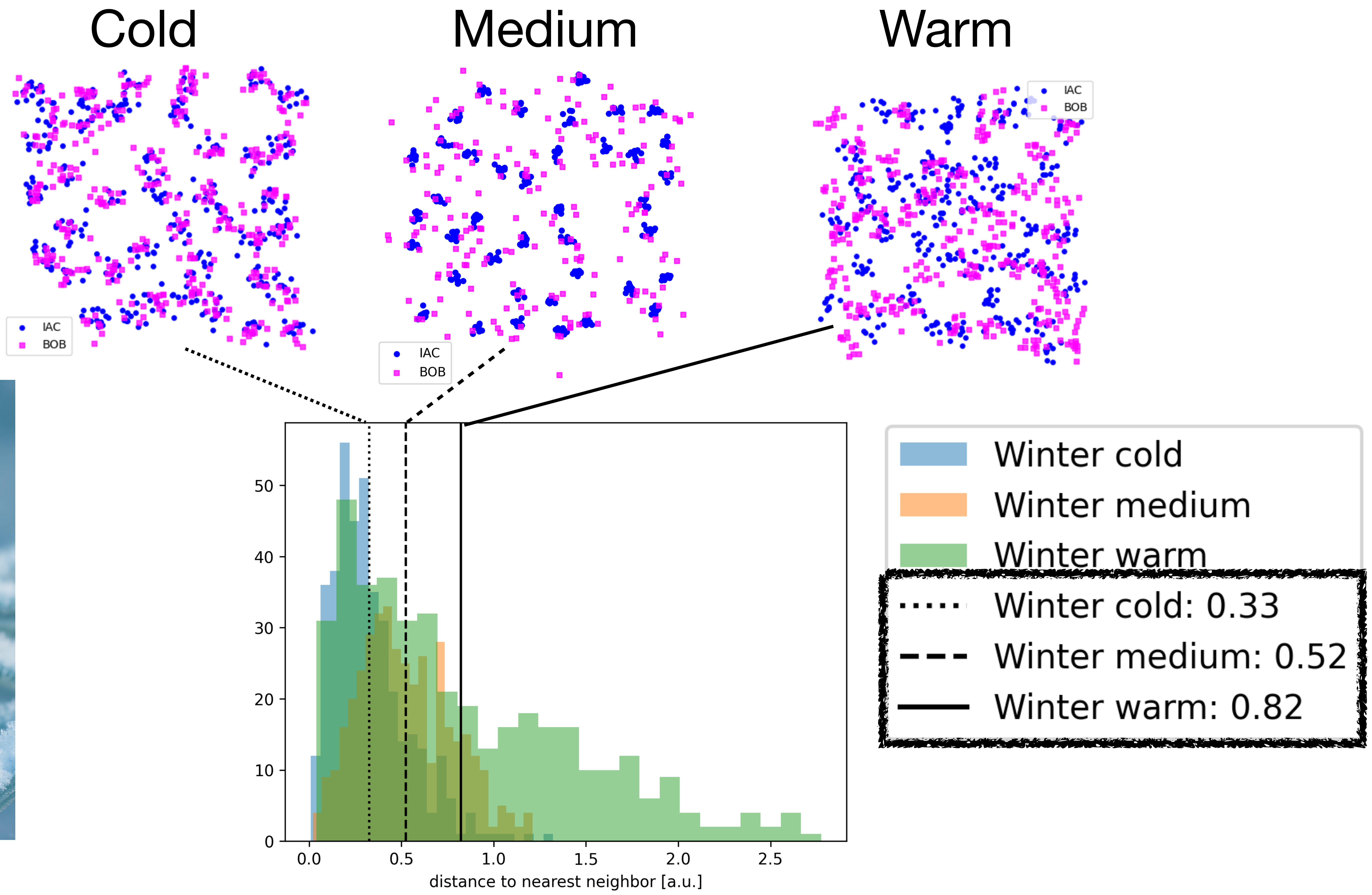
```
np.min(dist_matrix, axis = 1)
```



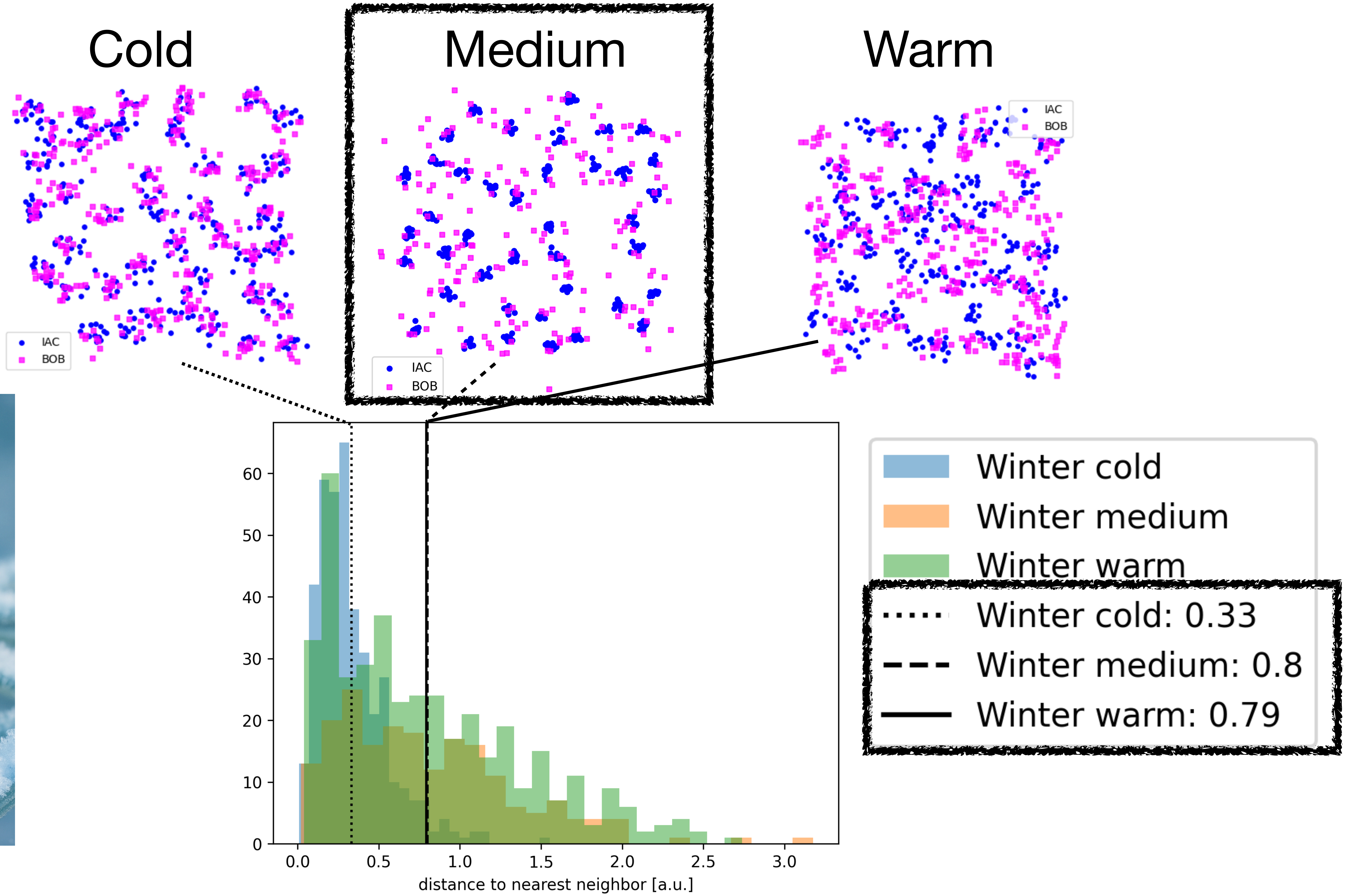
Results: Mean distance IAC -> BOB



Results: Mean distance IAC -> BOB



Results: Mean distance BOB -> IAC

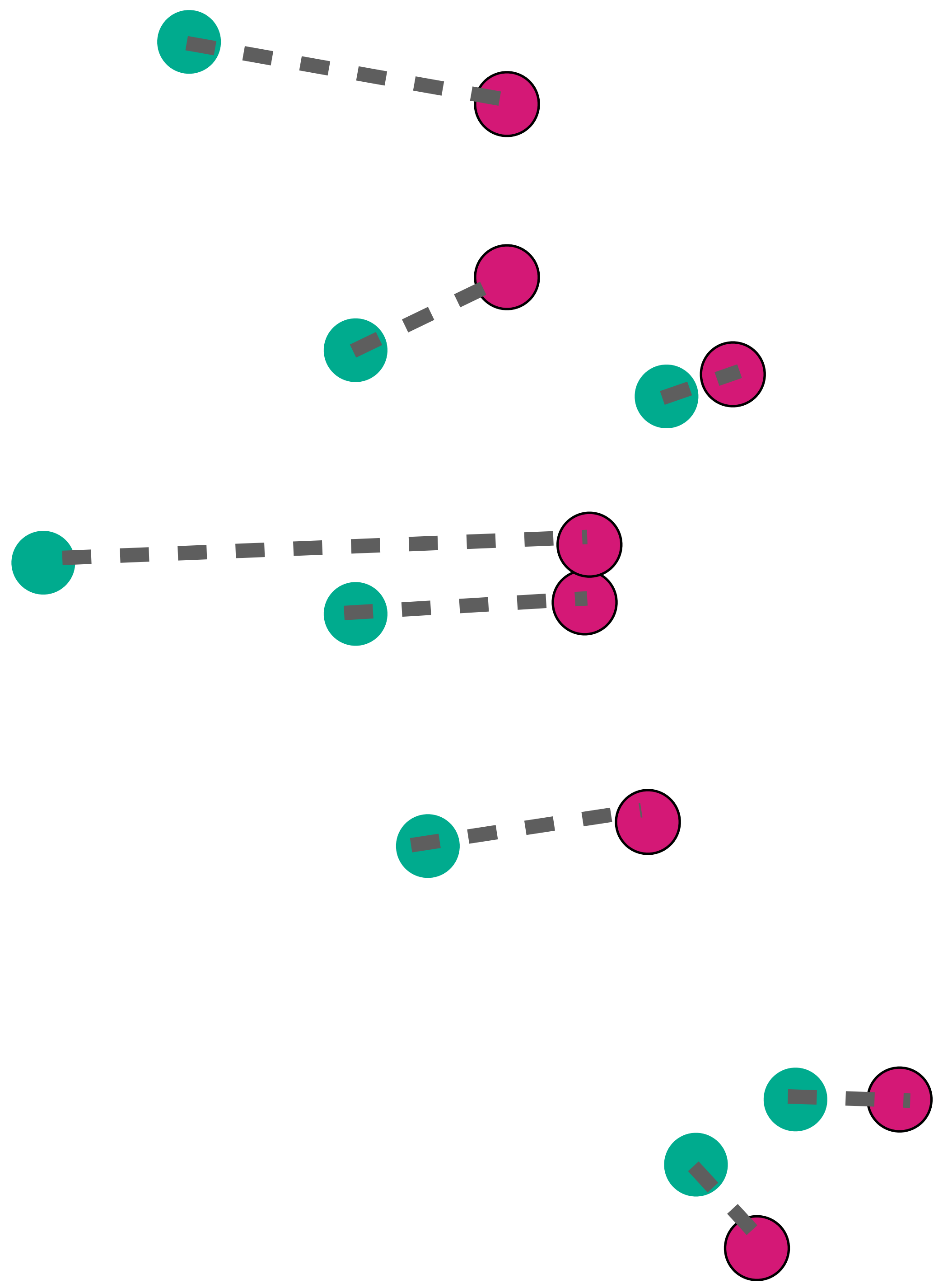


Mean distance to nearest neighbor

- Asymmetric: BOB → IAC ≠ IAC → BOB
- Returns: One number
- Range: Short

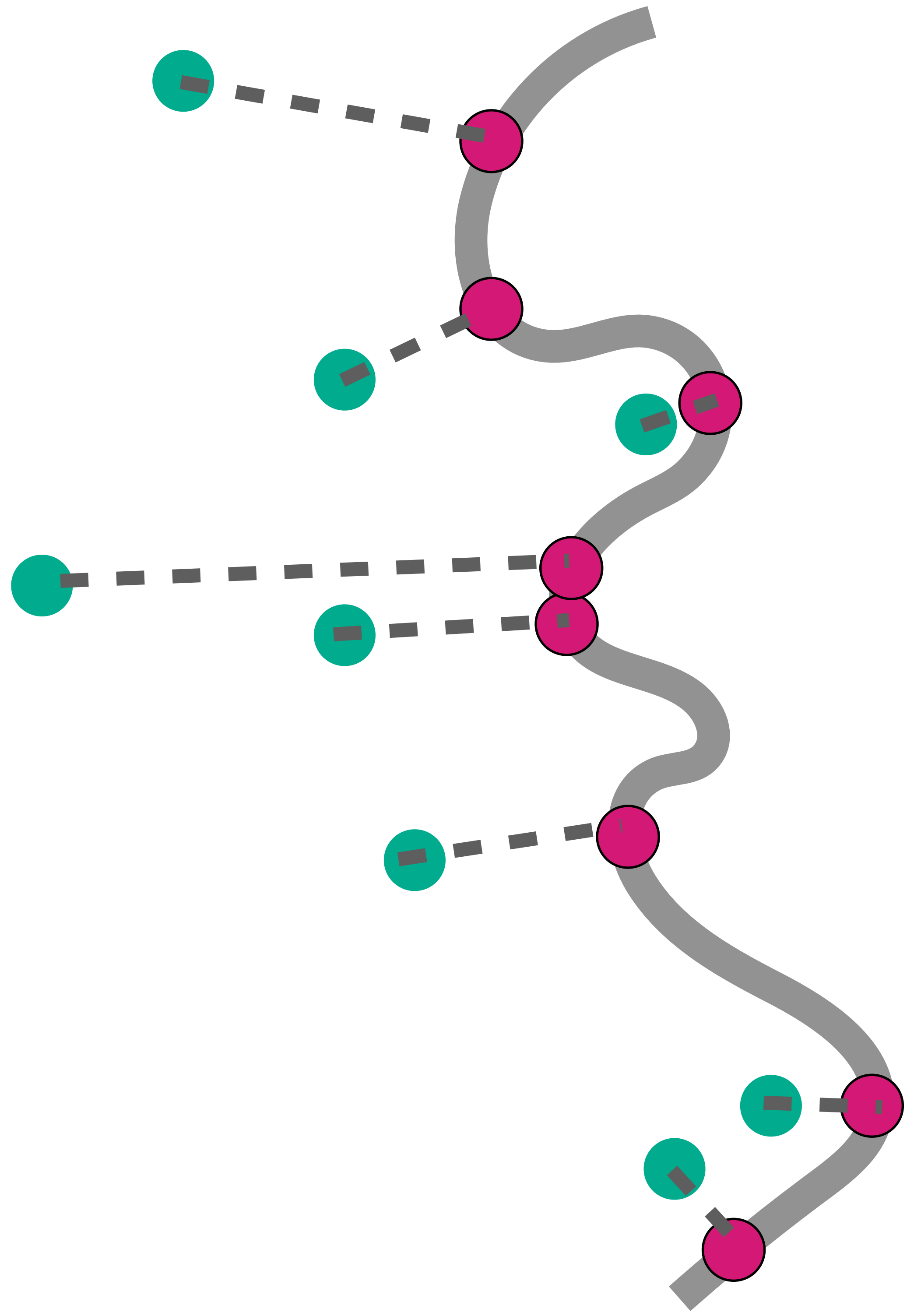
Beyond the mean distance to nearest neighbor

- Similar concepts hold true beyond the realm of just points



Beyond the mean distance to nearest neighbor

- Similar concepts hold true beyond the realm of just points



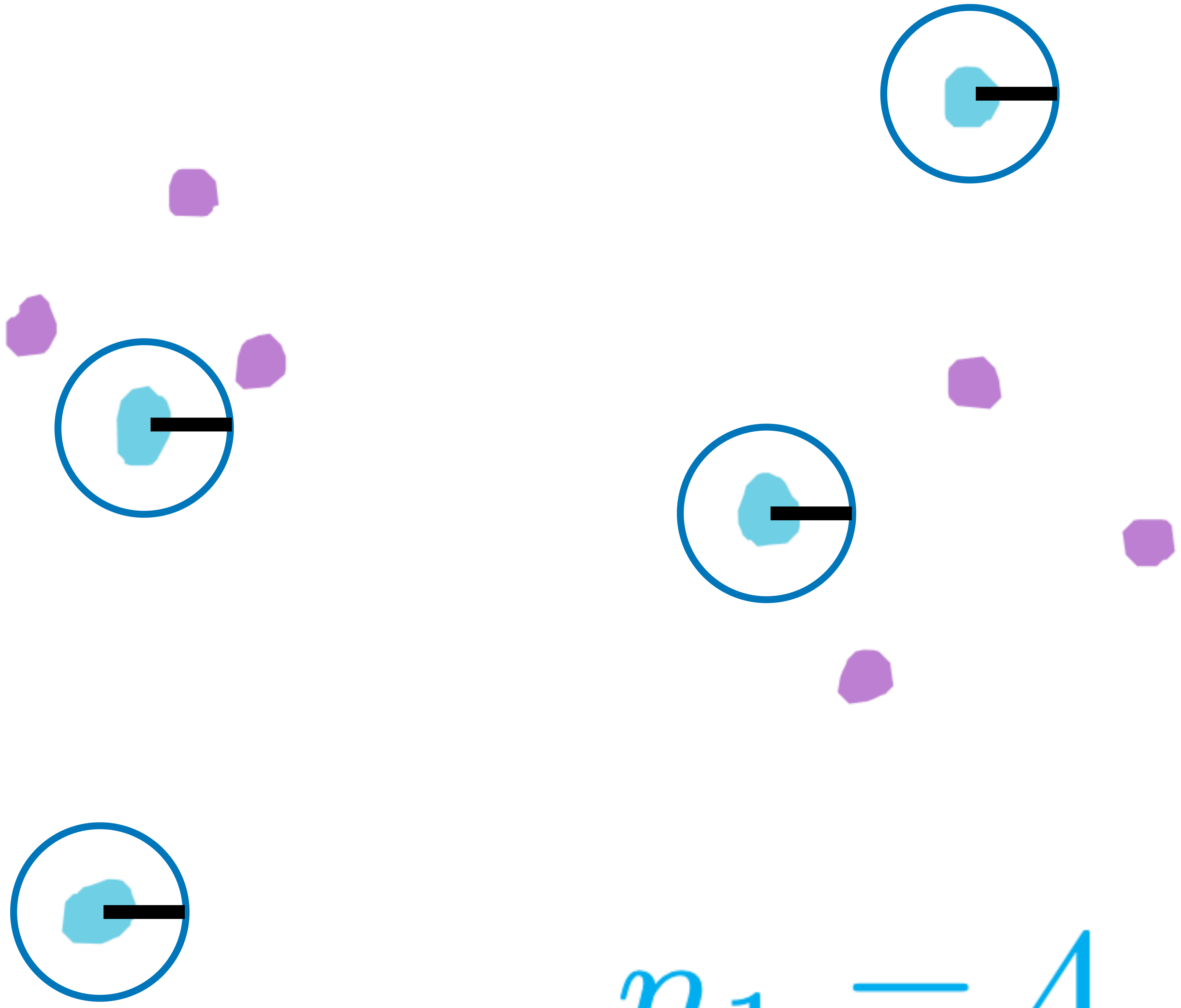
Nearest neighbor function

Mean dist: 1

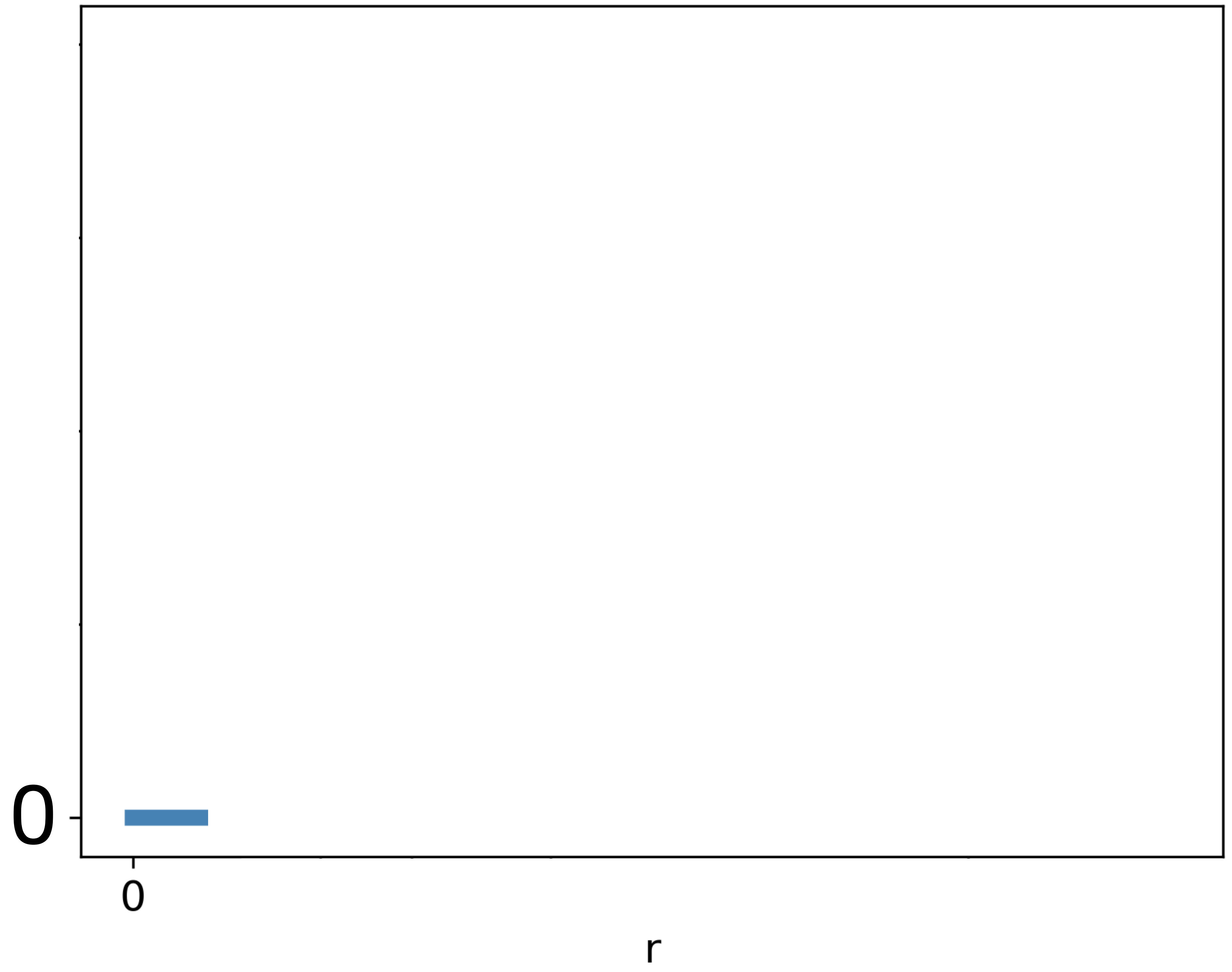
Mean dist: 1

Nearest neighbor function

$$S(r) = \frac{1}{n_1}\sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$

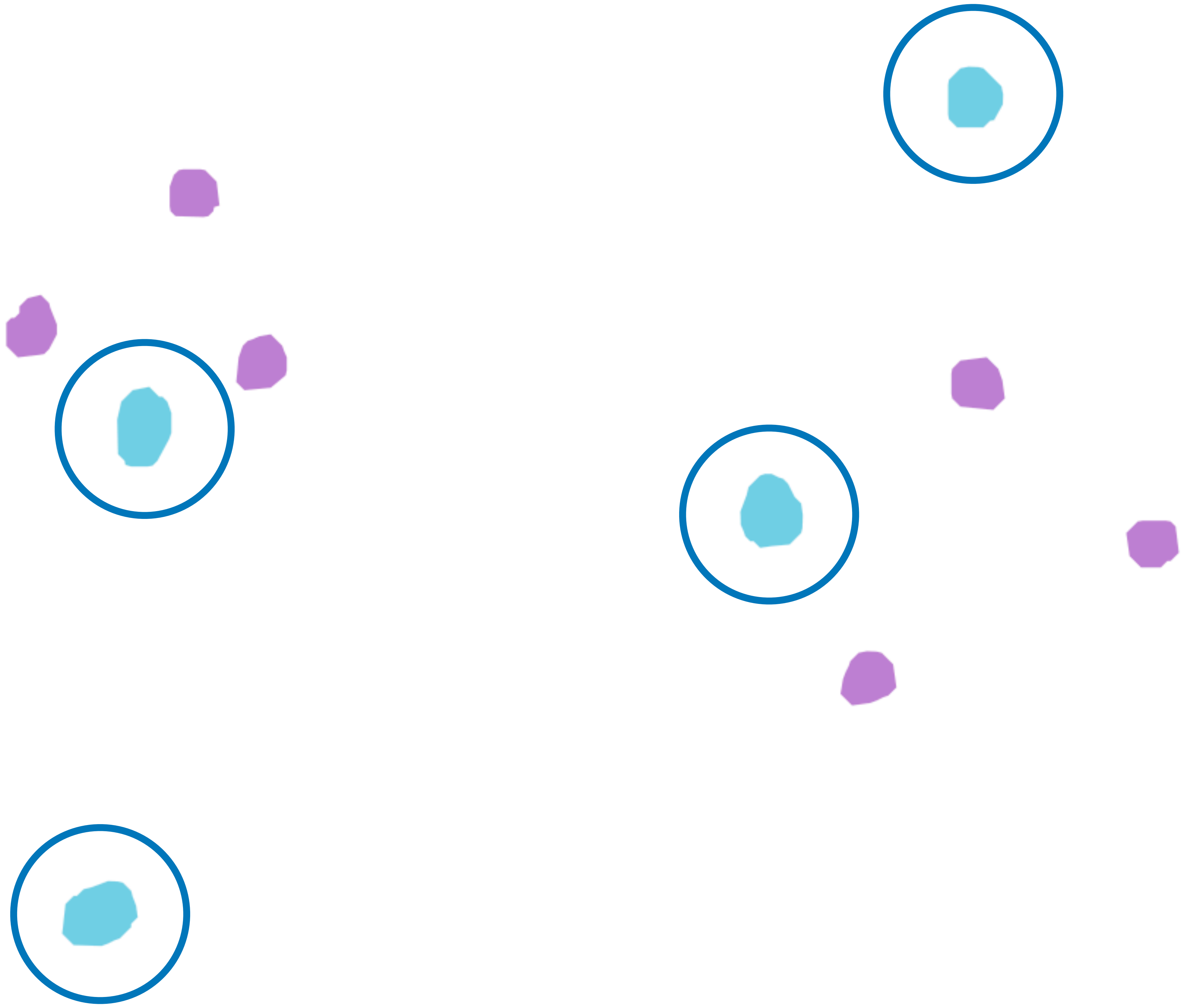


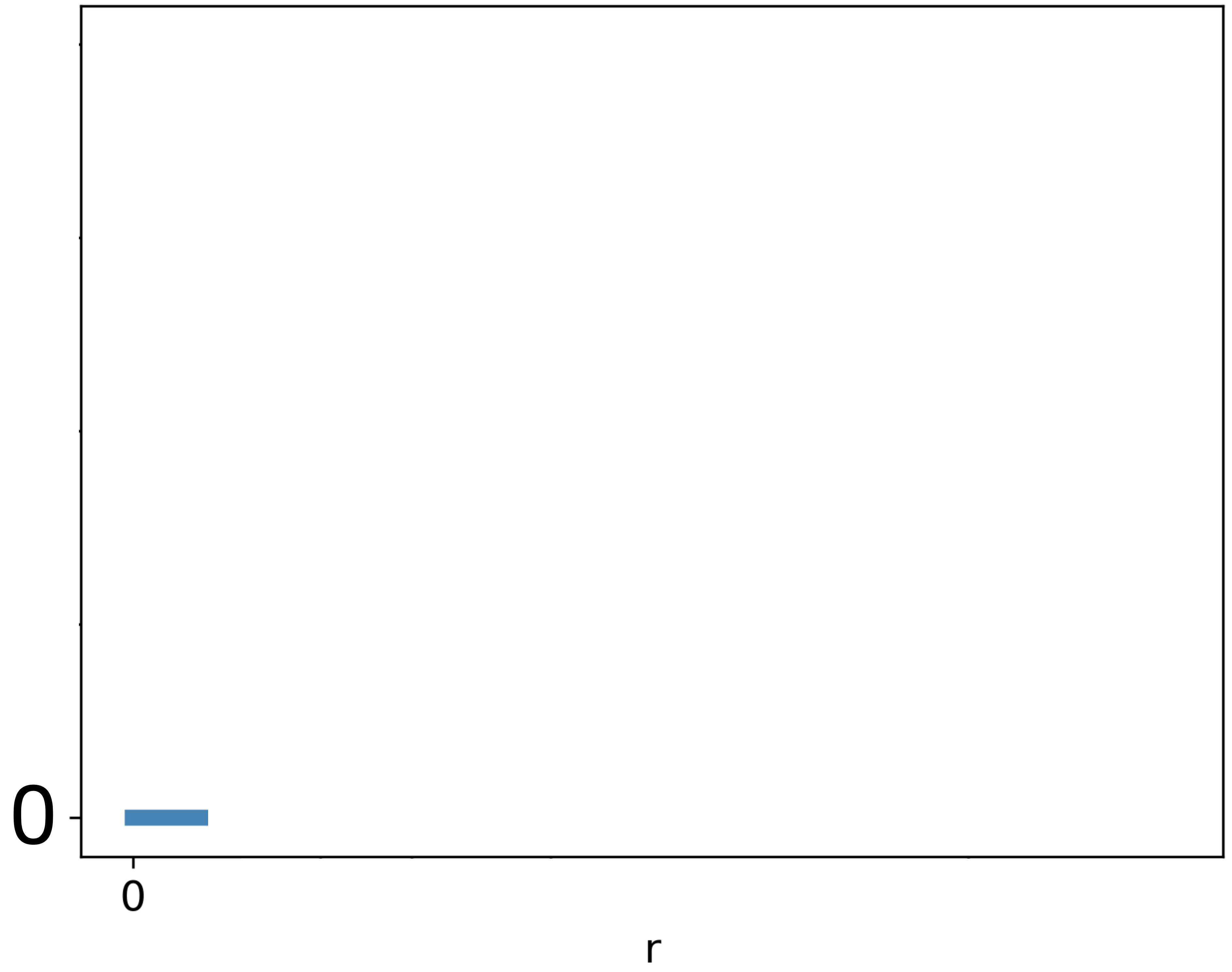
$$n_1 = 4$$



Nearest neighbor function

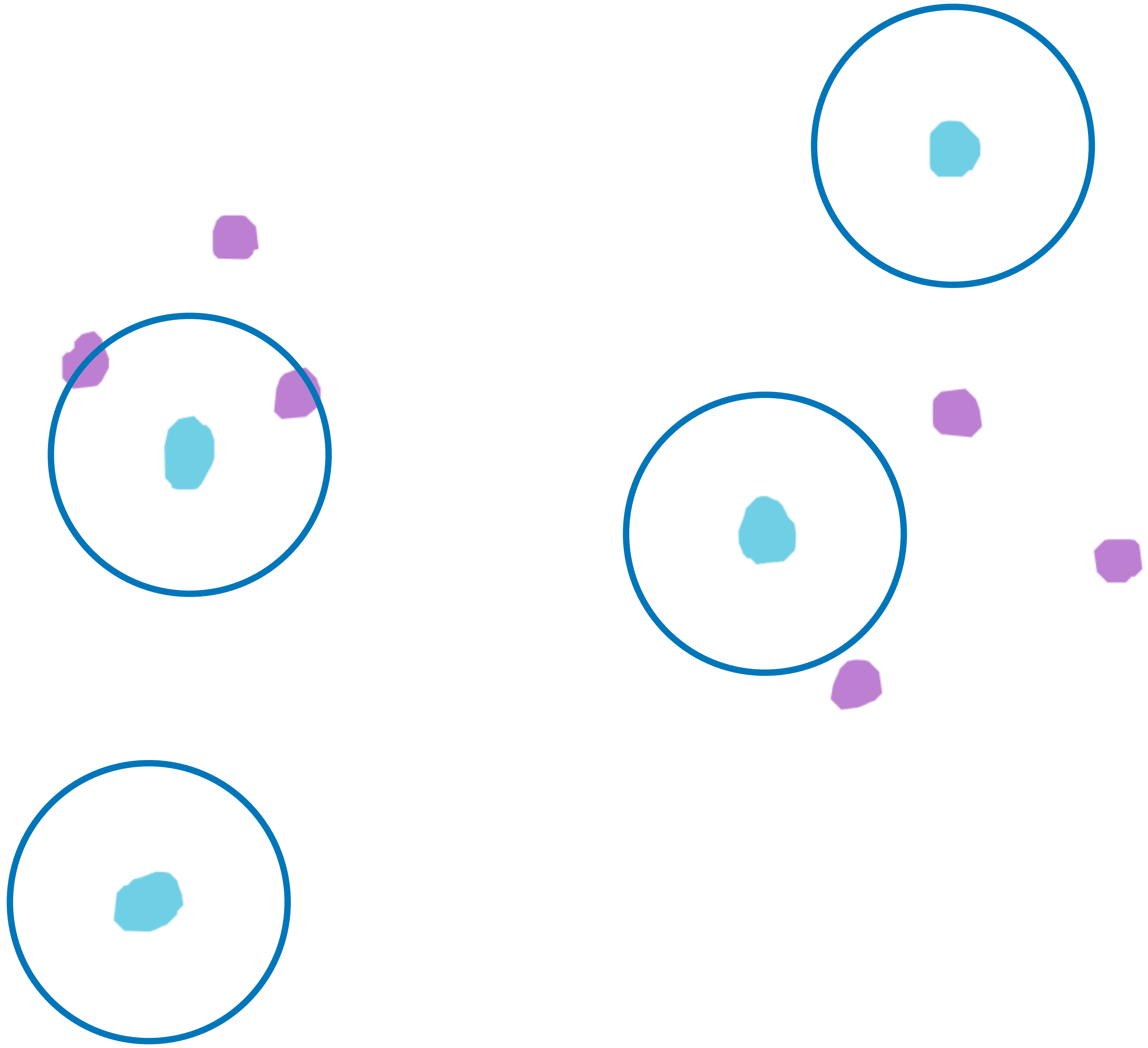
$$S(r) = \frac{1}{n_1}\sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$

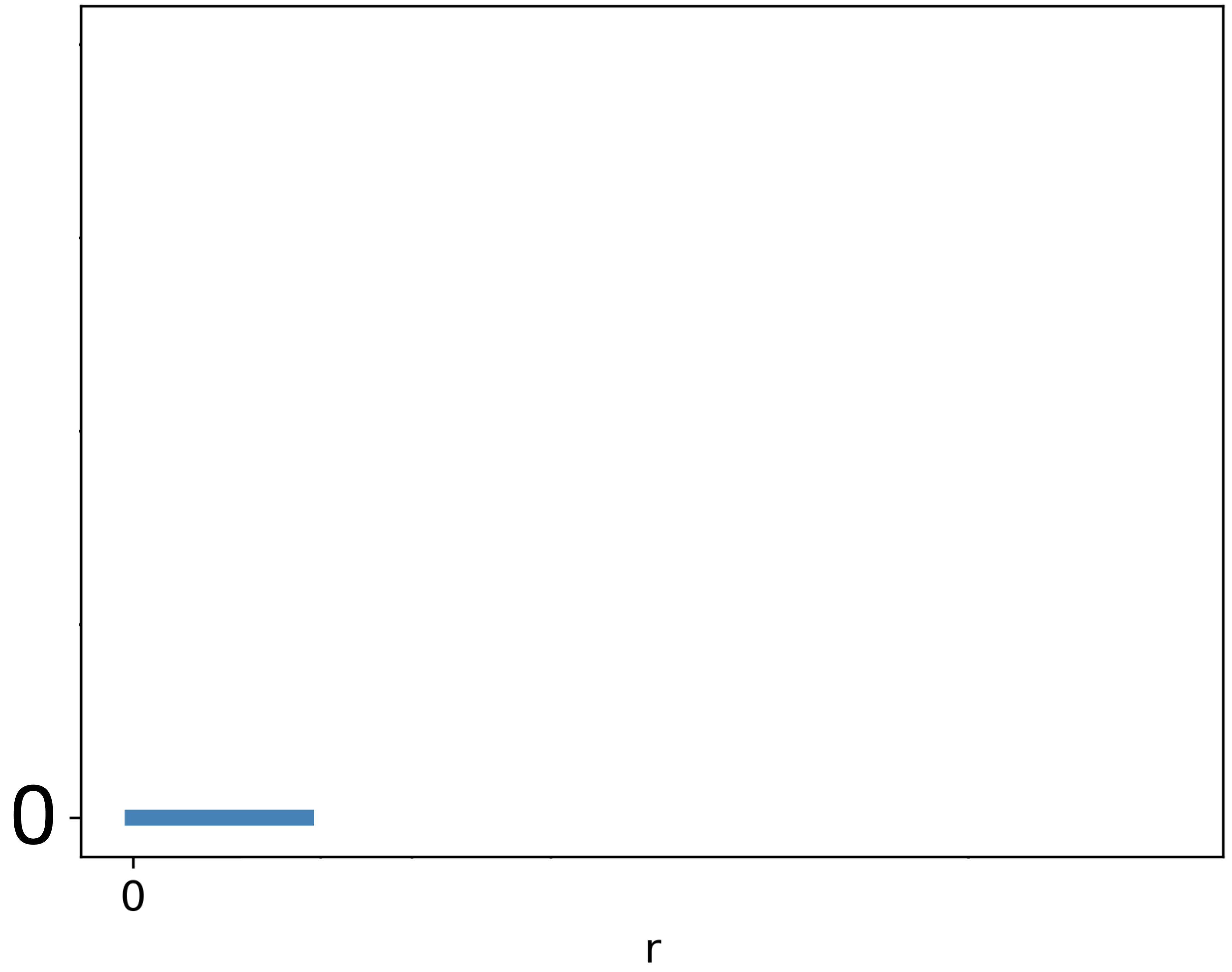




Nearest neighbor function

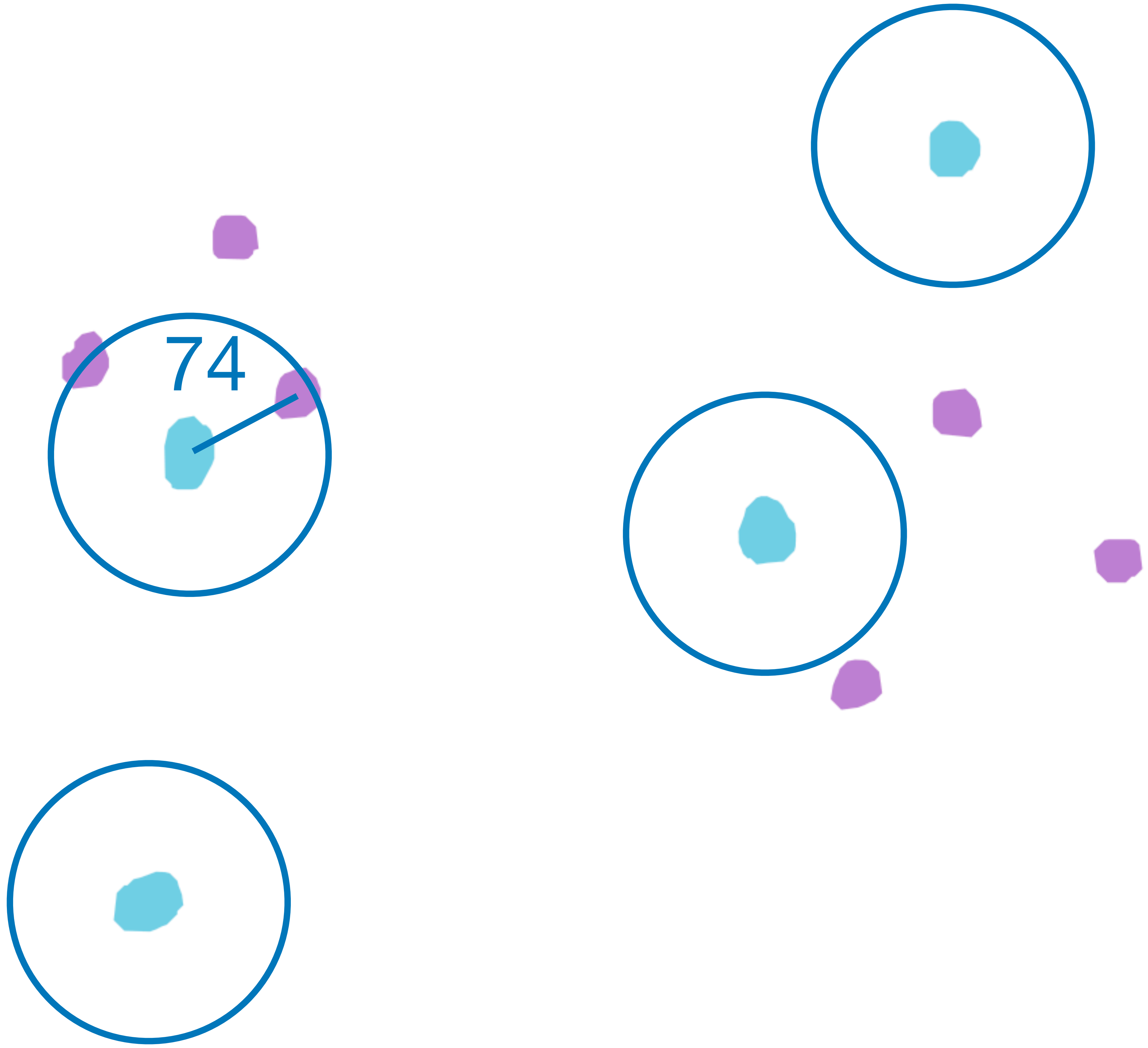
$$S(r) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$

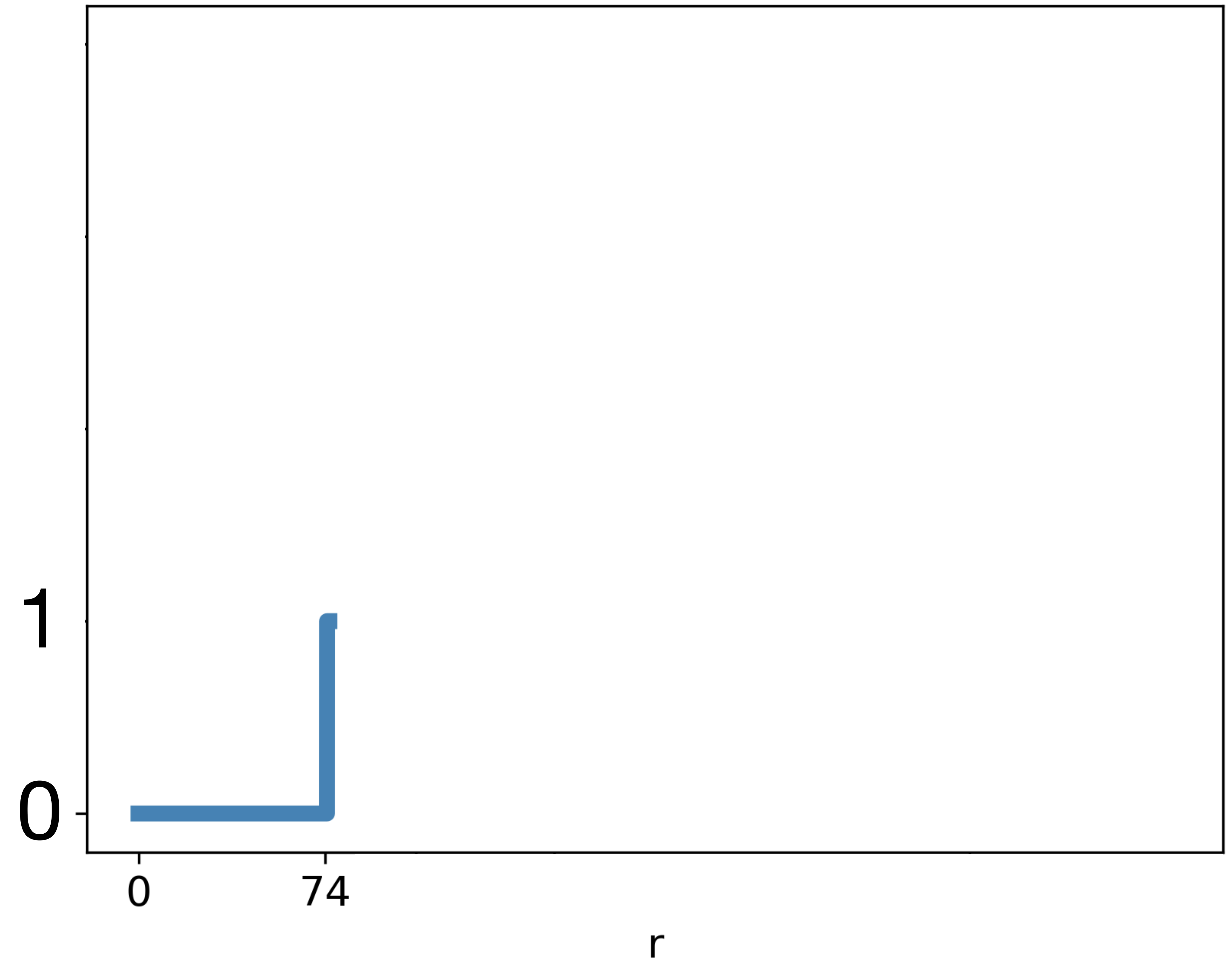




Nearest neighbor function

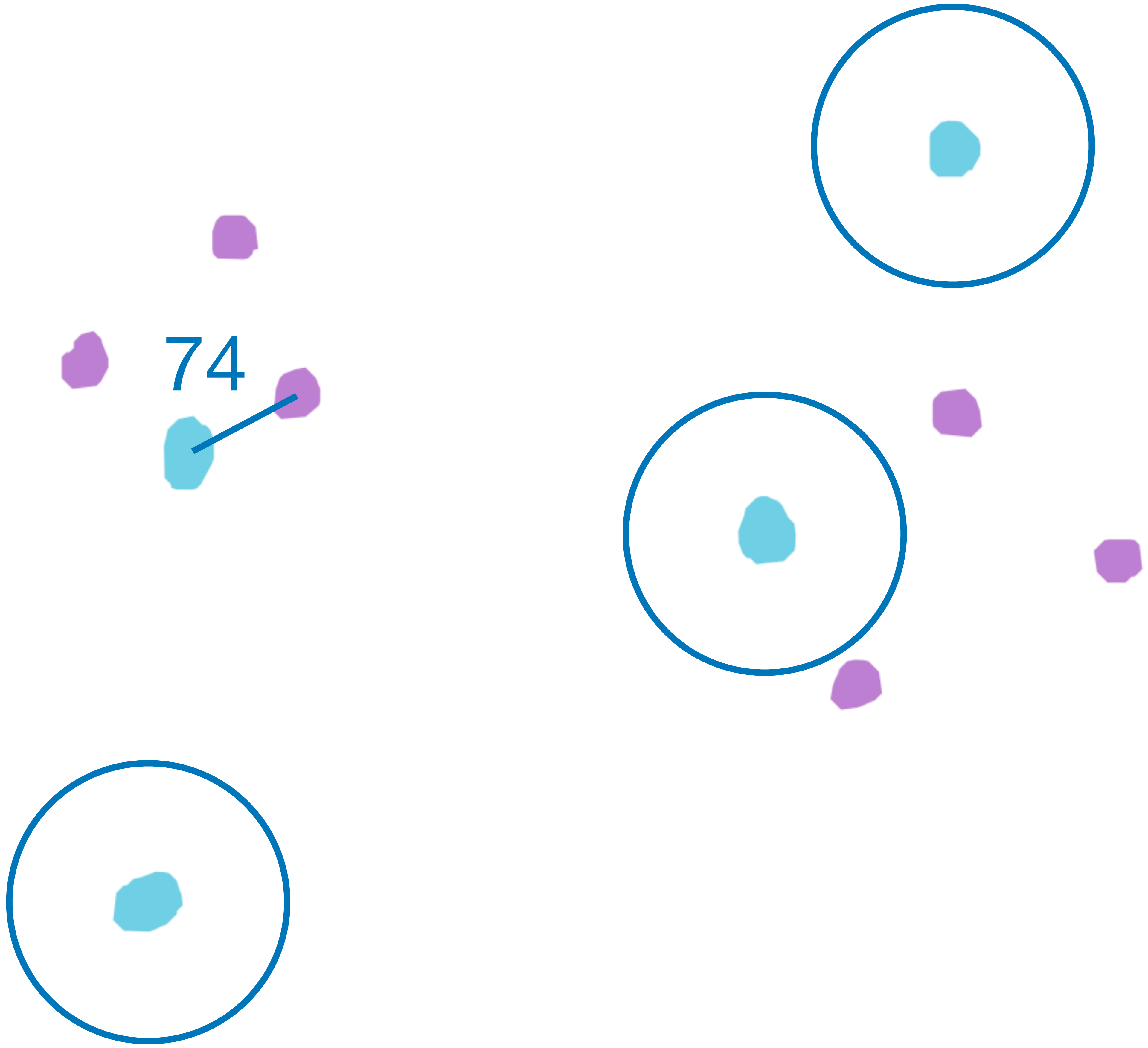
$$S(r) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$

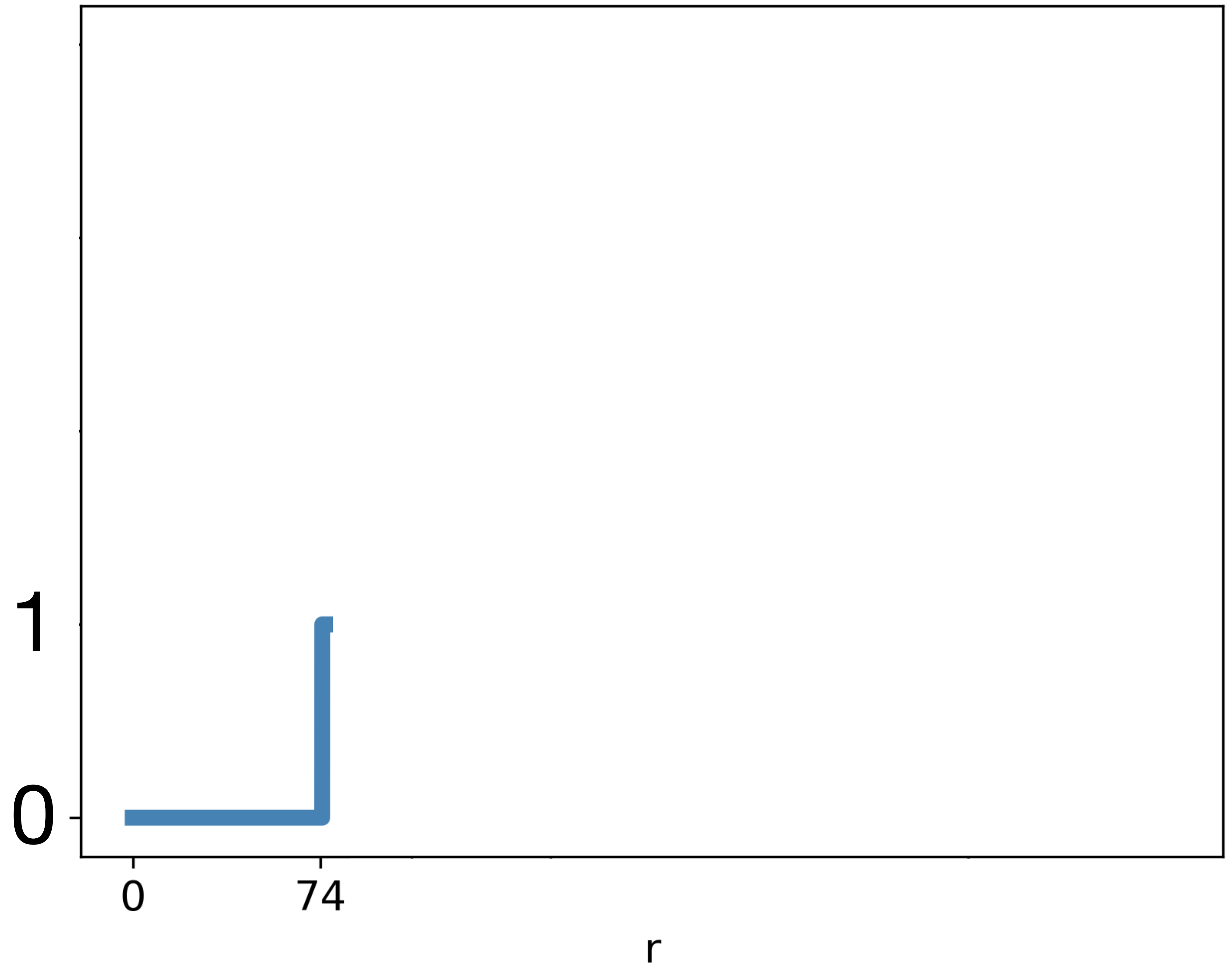




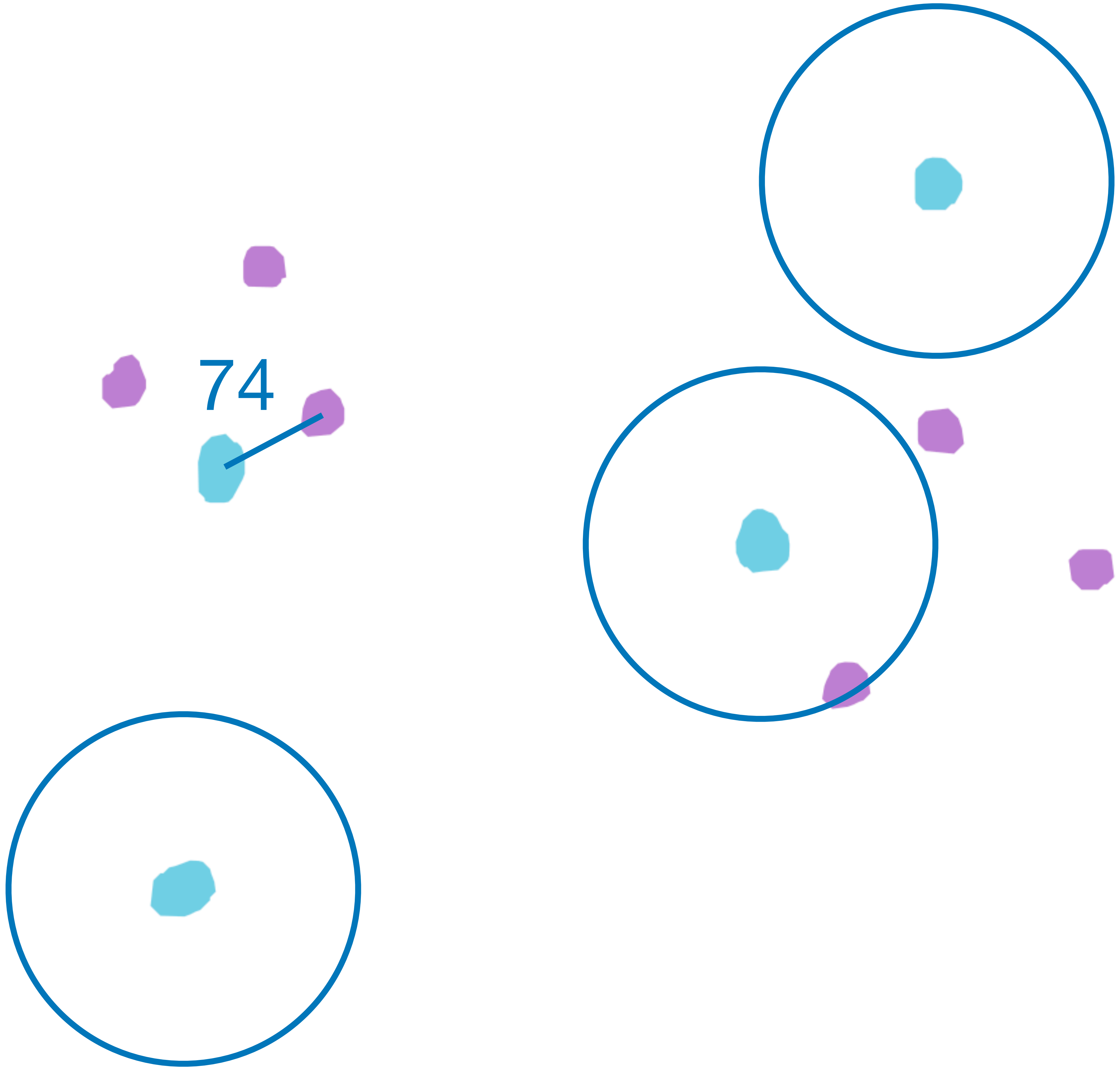
Nearest neighbor function

$$S(r) = \frac{1}{n_1}\sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$

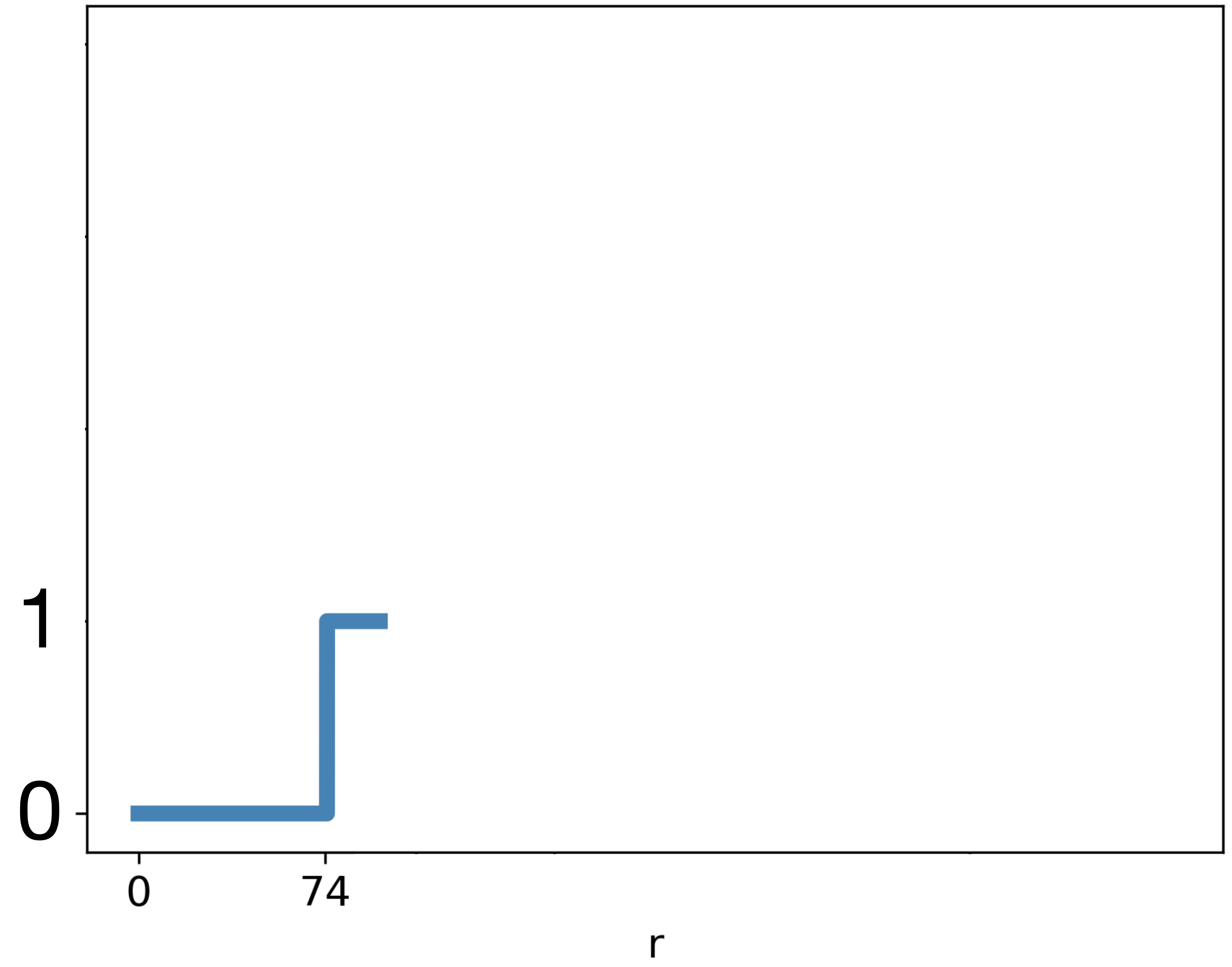




Nearest neighbor function

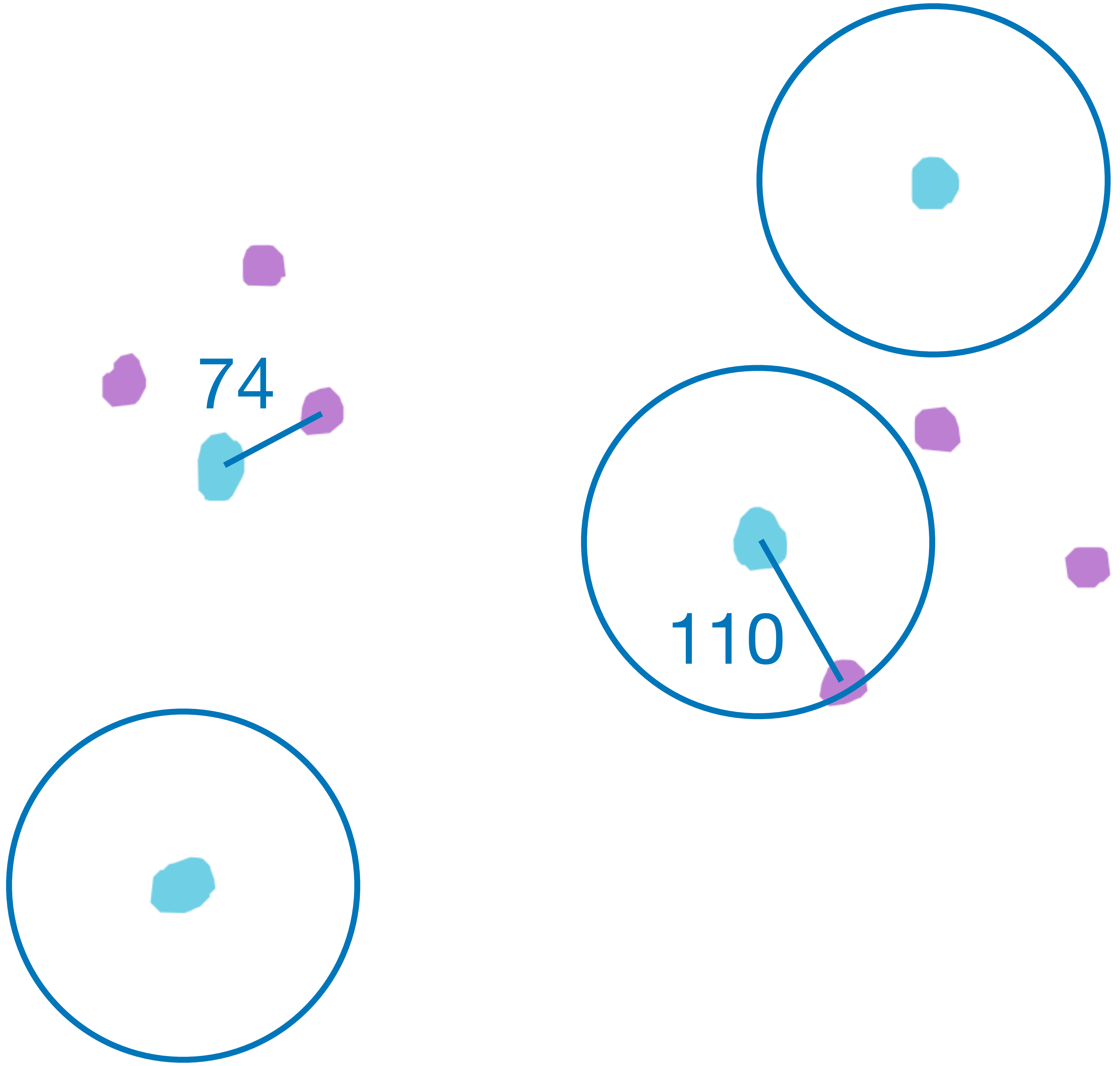


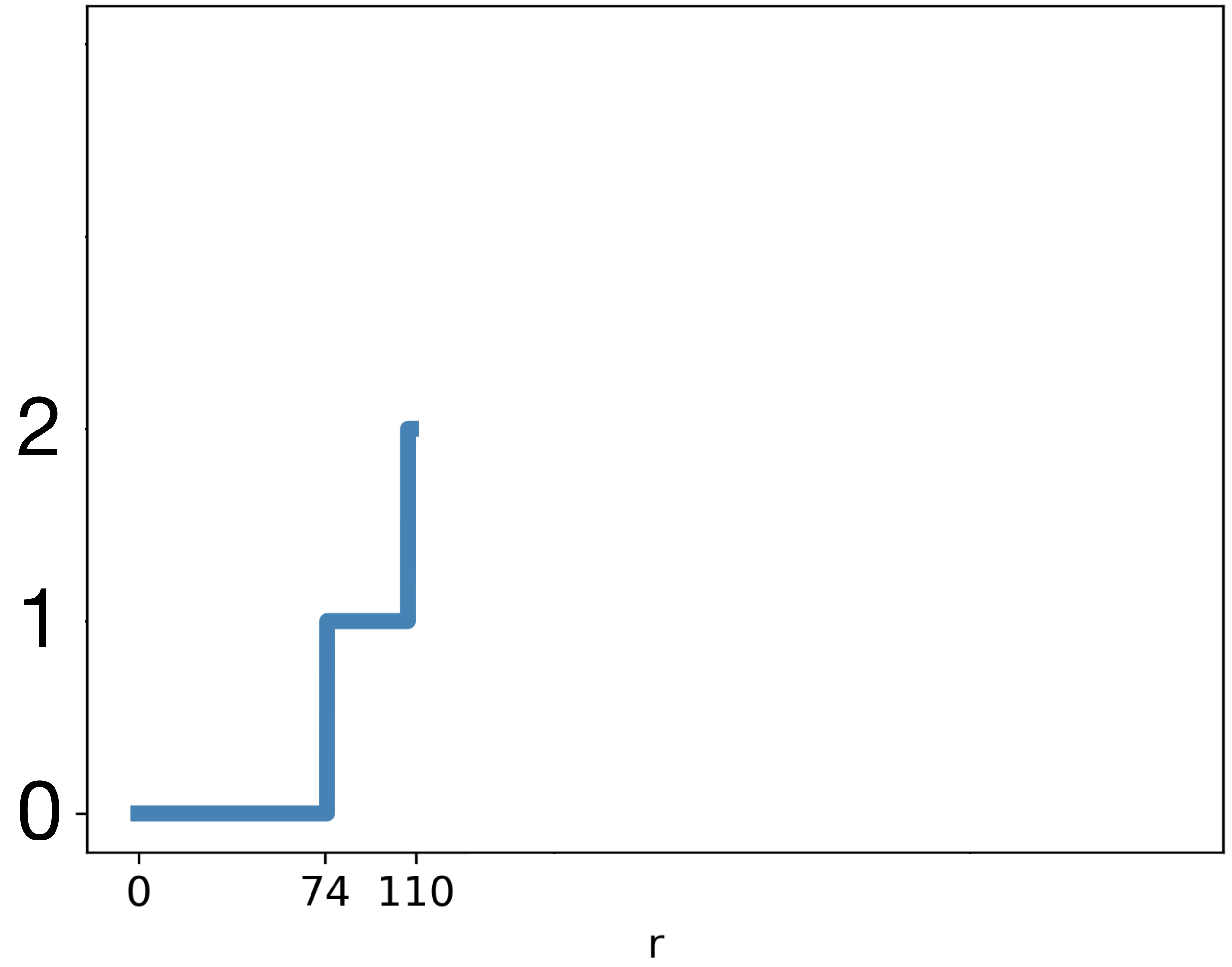
$$S(r) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$



Nearest neighbor function

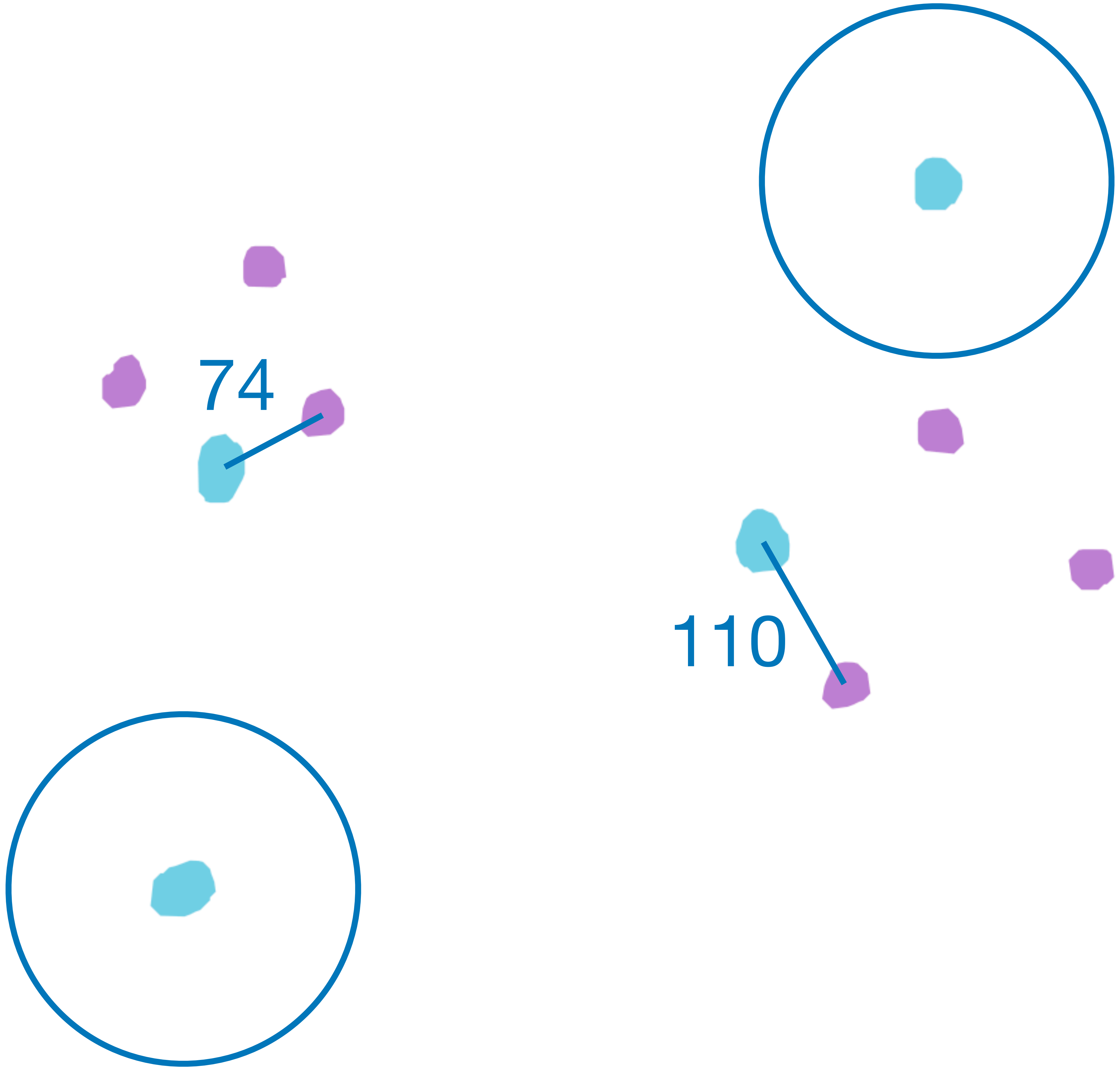
$$S(r) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$

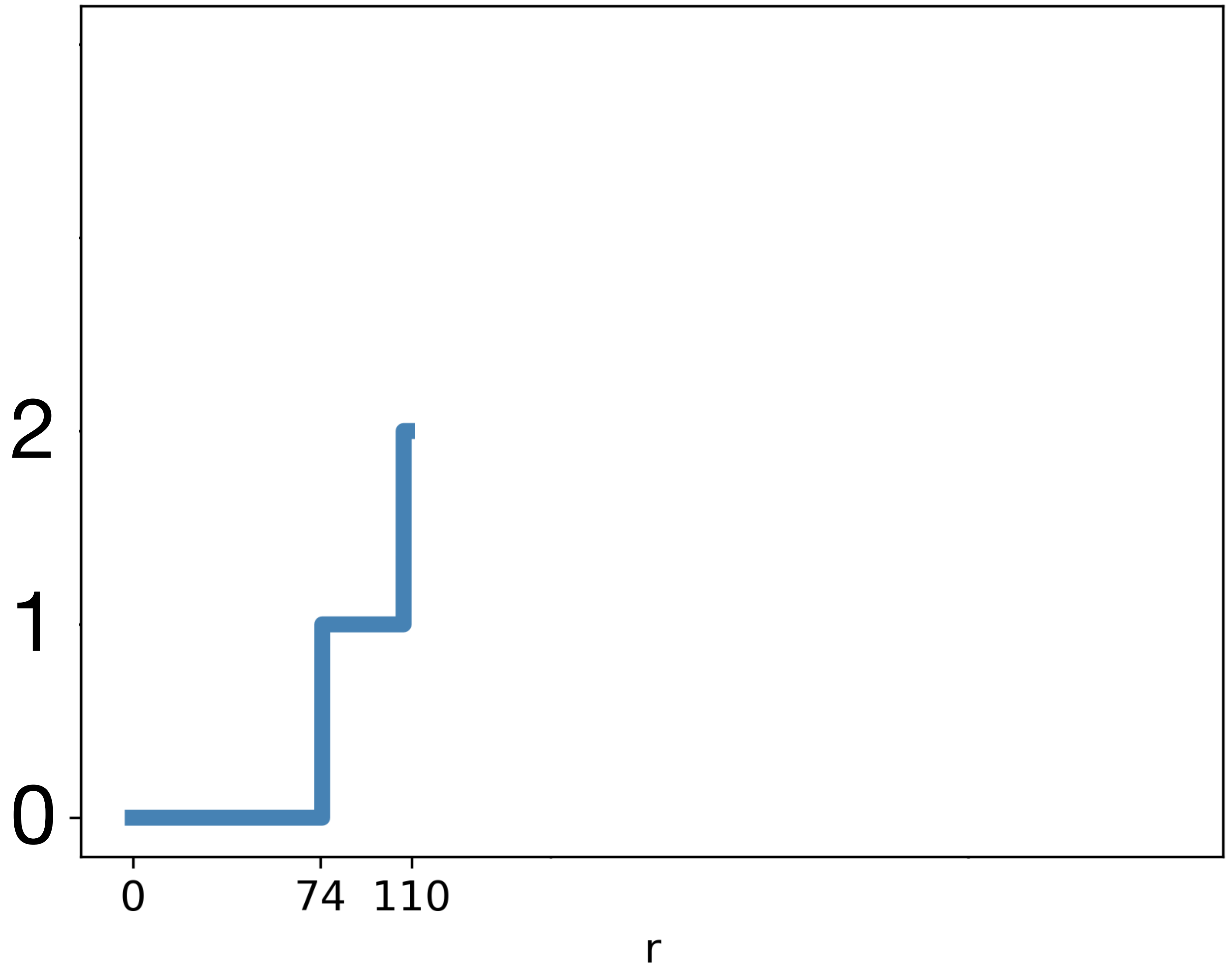




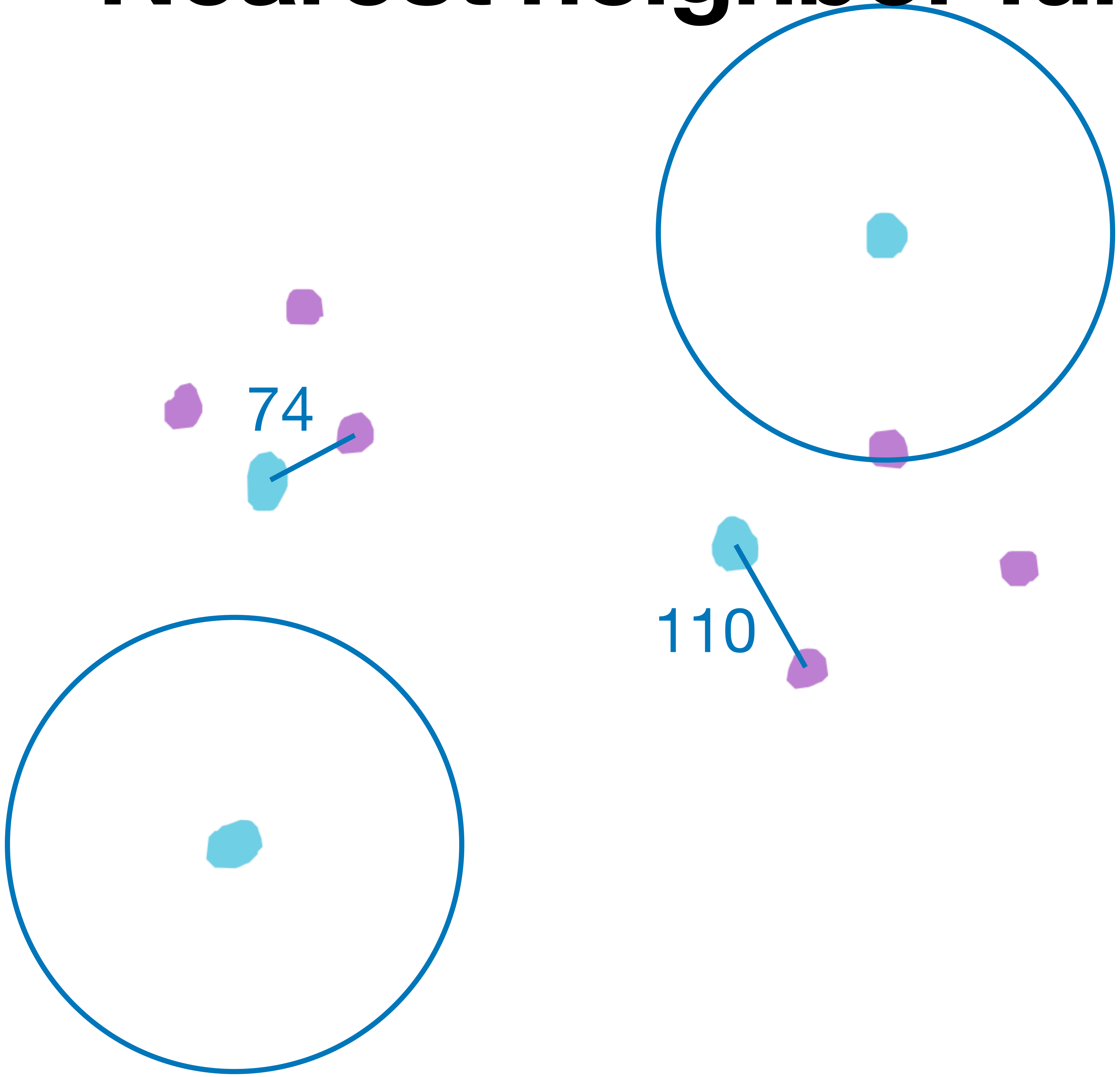
Nearest neighbor function

$$S(r) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$

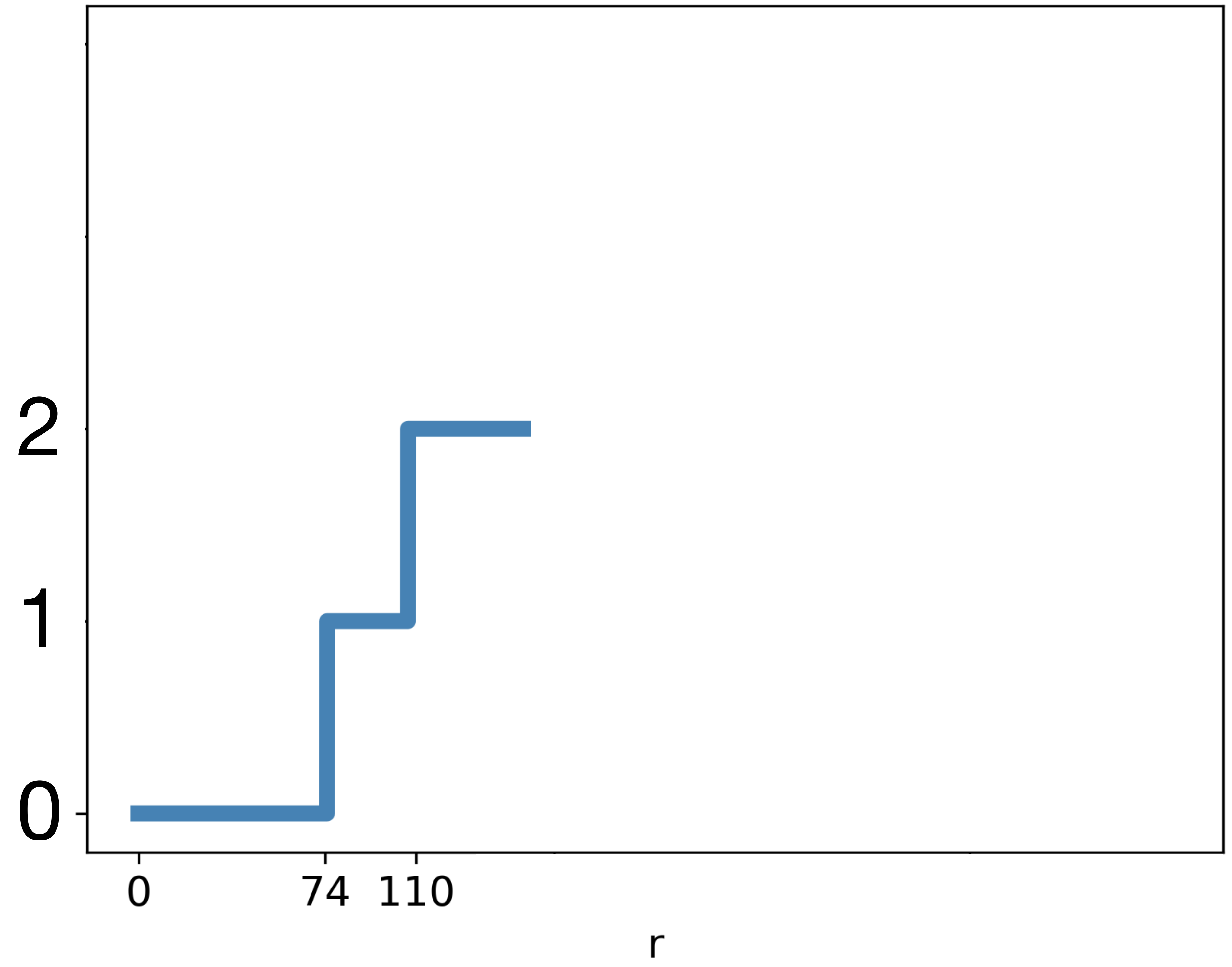




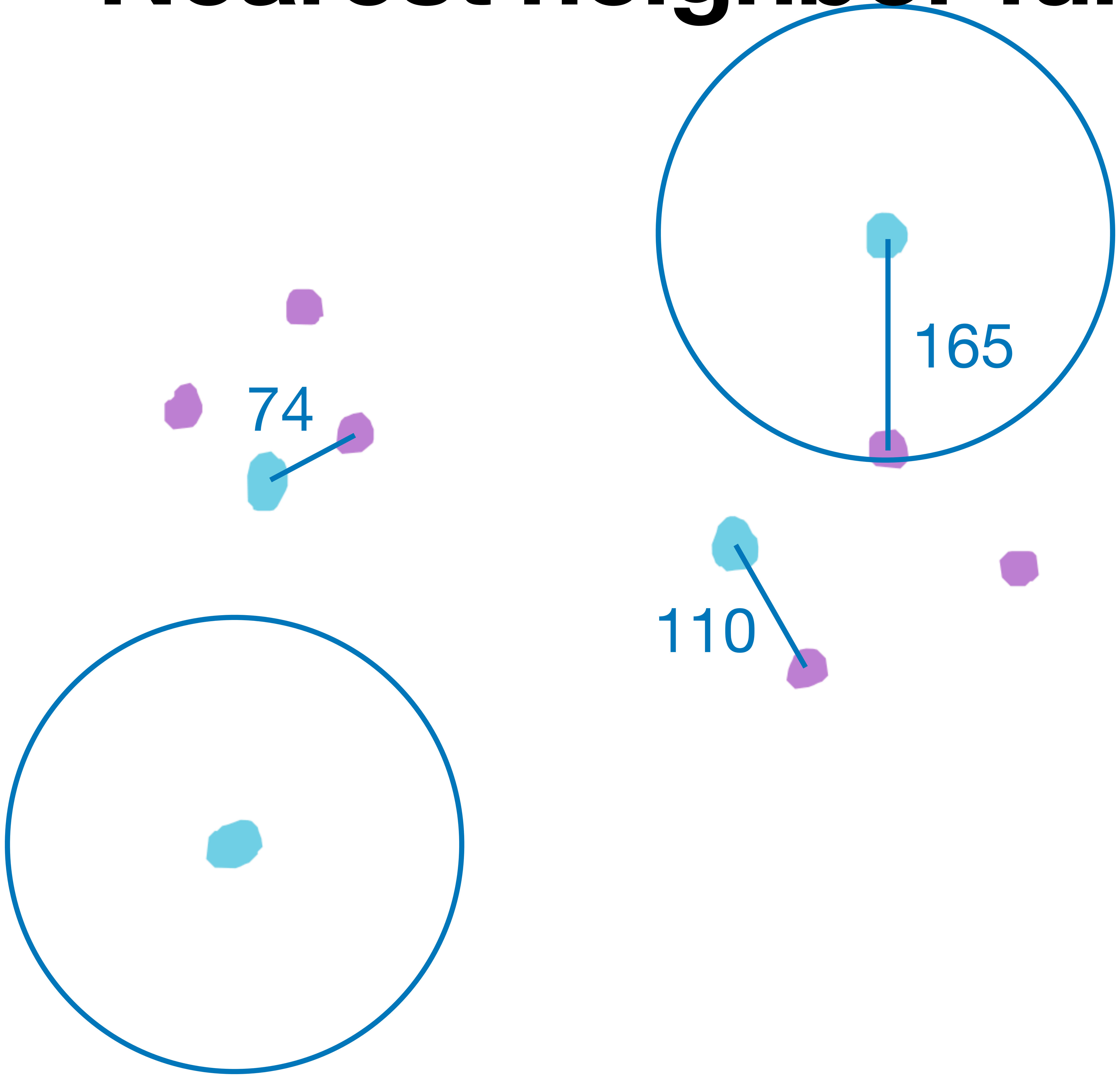
Nearest neighbor function



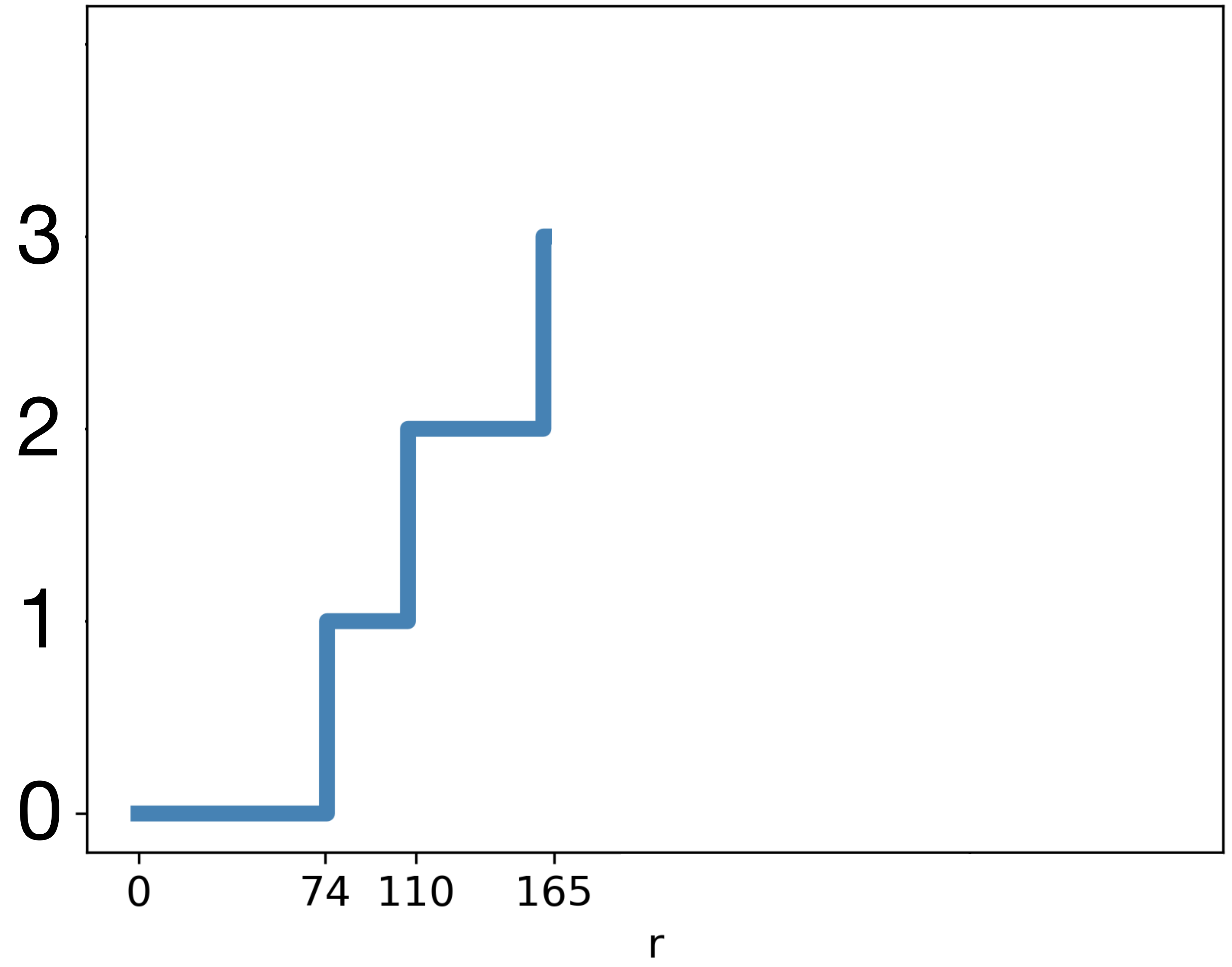
$$S(r) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$



Nearest neighbor function

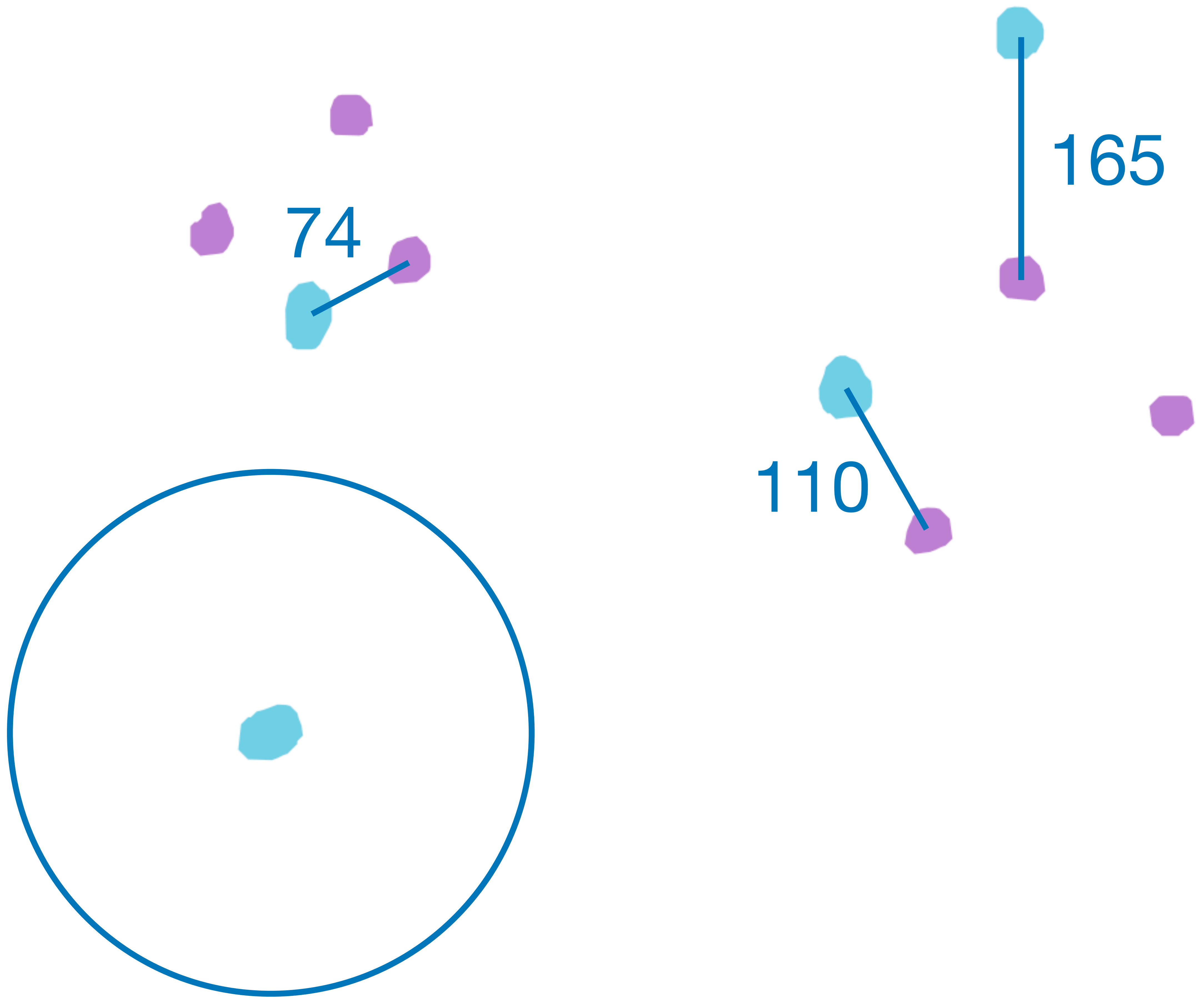


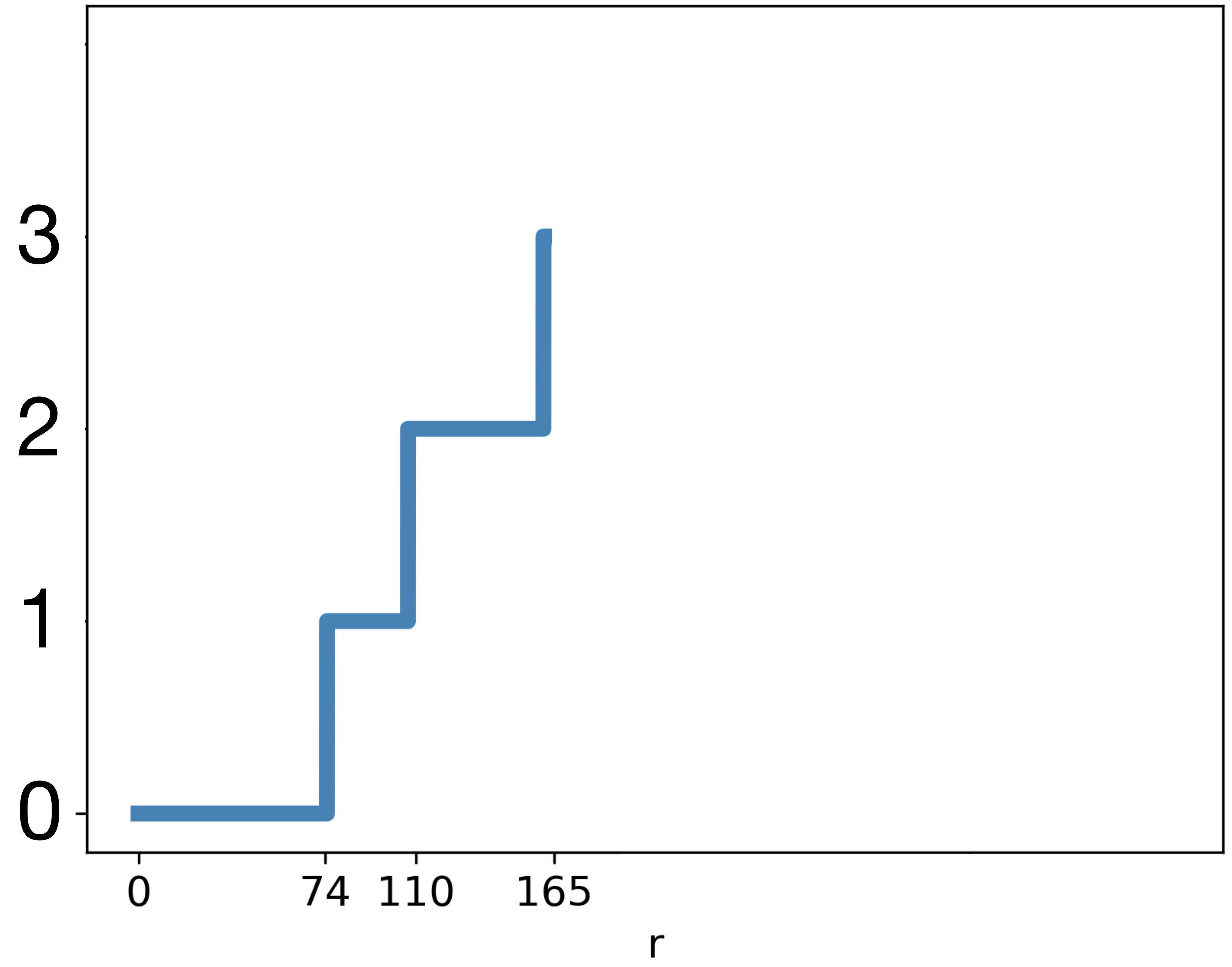
$$S(r) = \frac{1}{n_1}\sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$



Nearest neighbor function

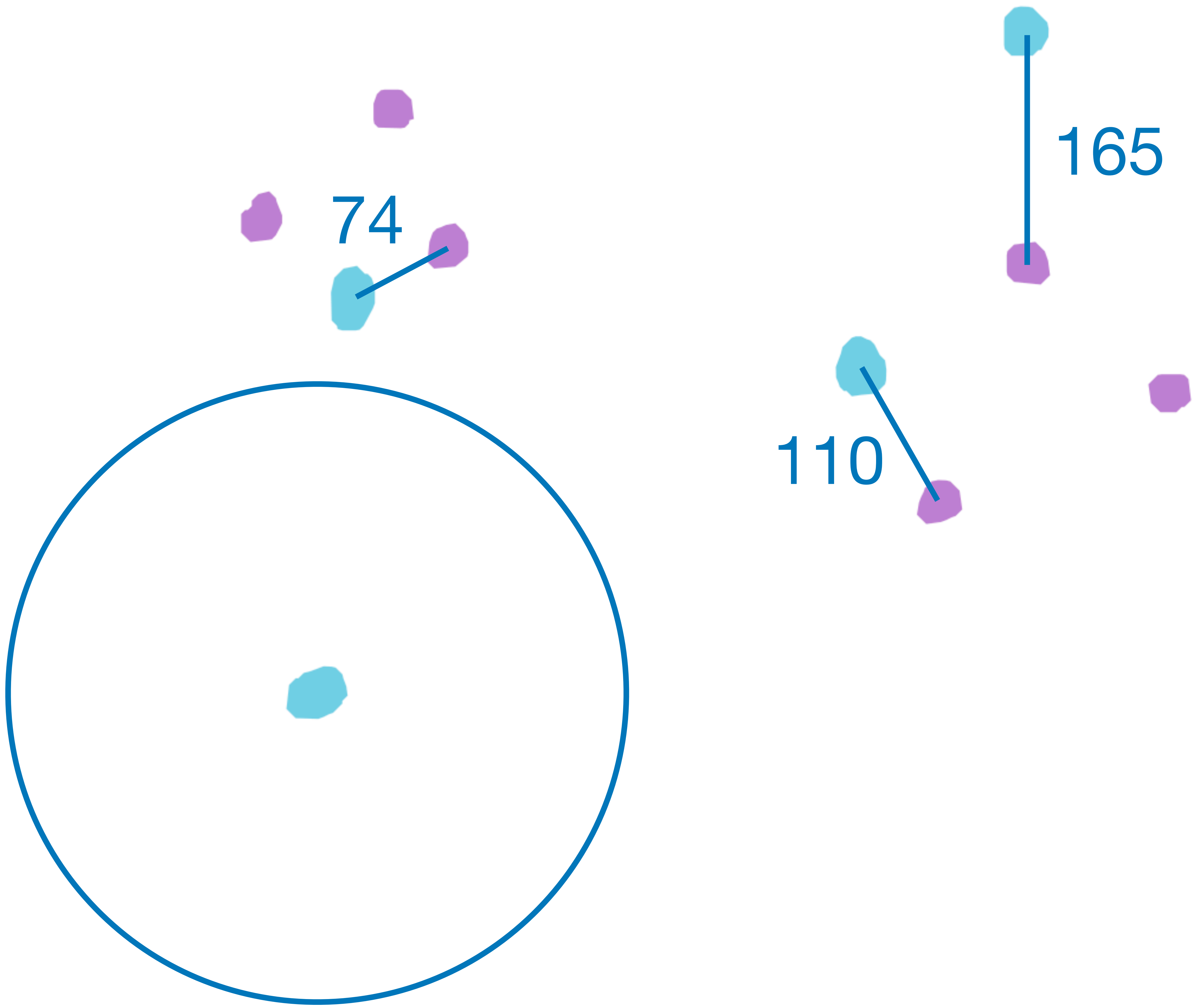
$$S(r) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$

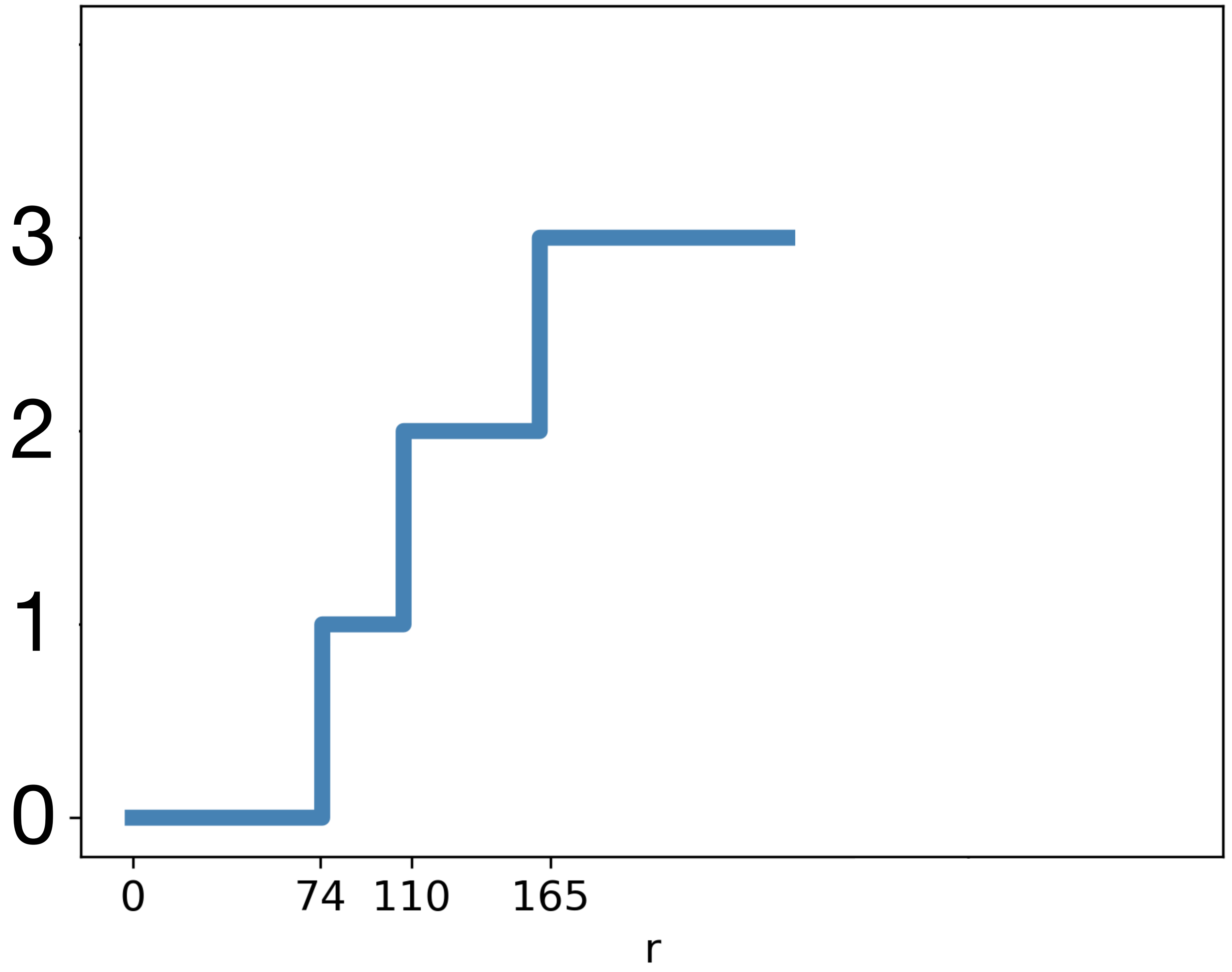




Nearest neighbor function

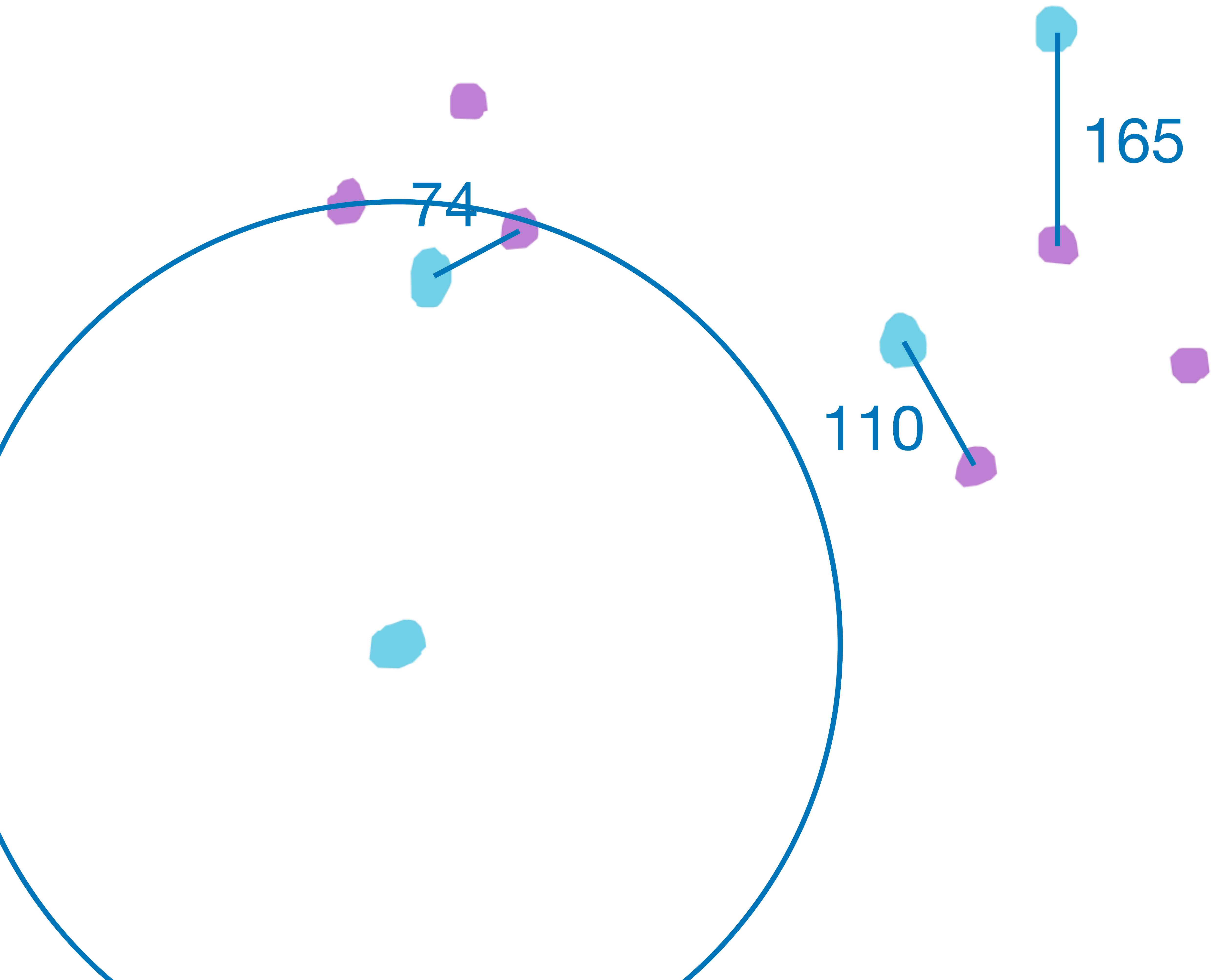
$$S(r) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$

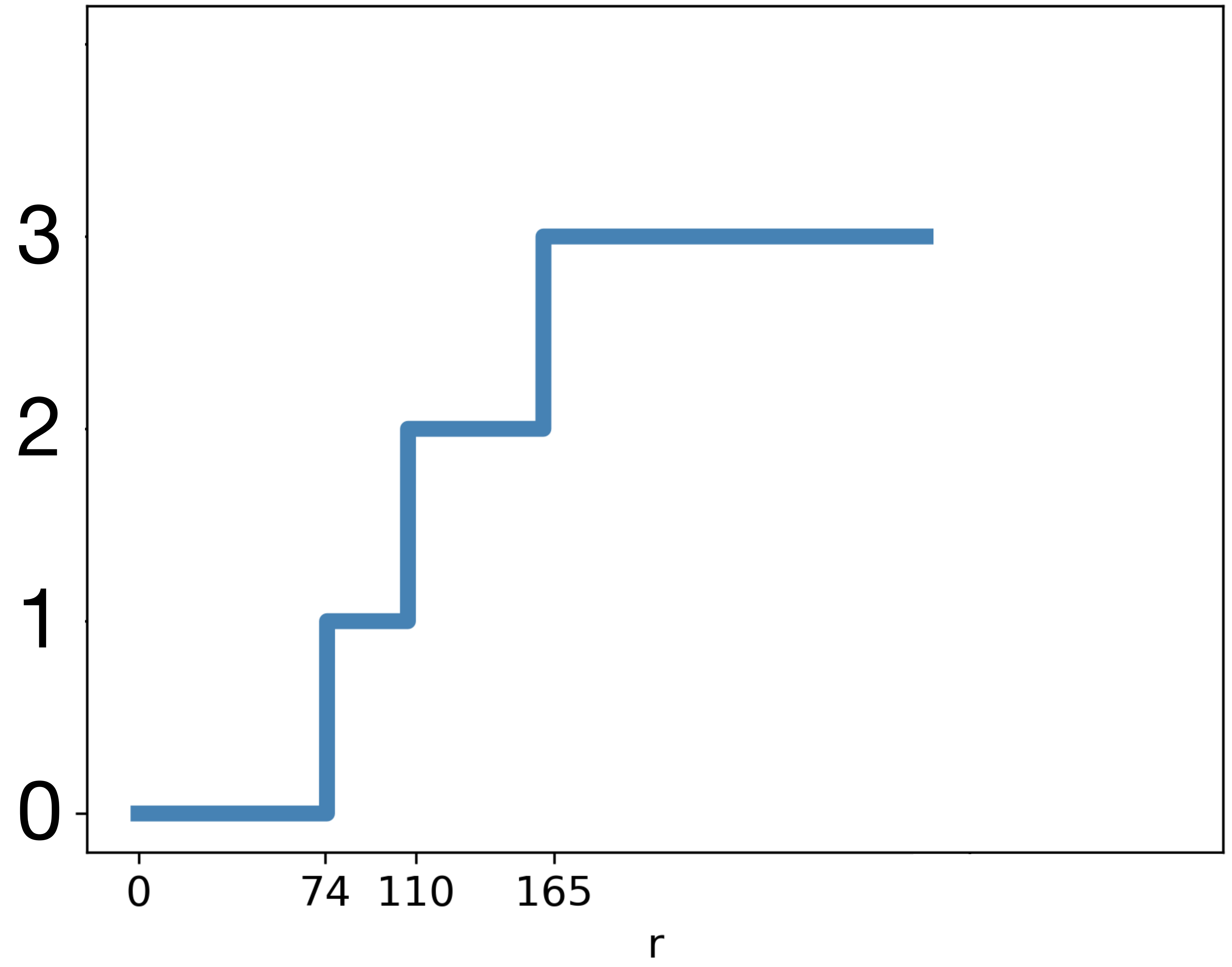




Nearest neighbor function

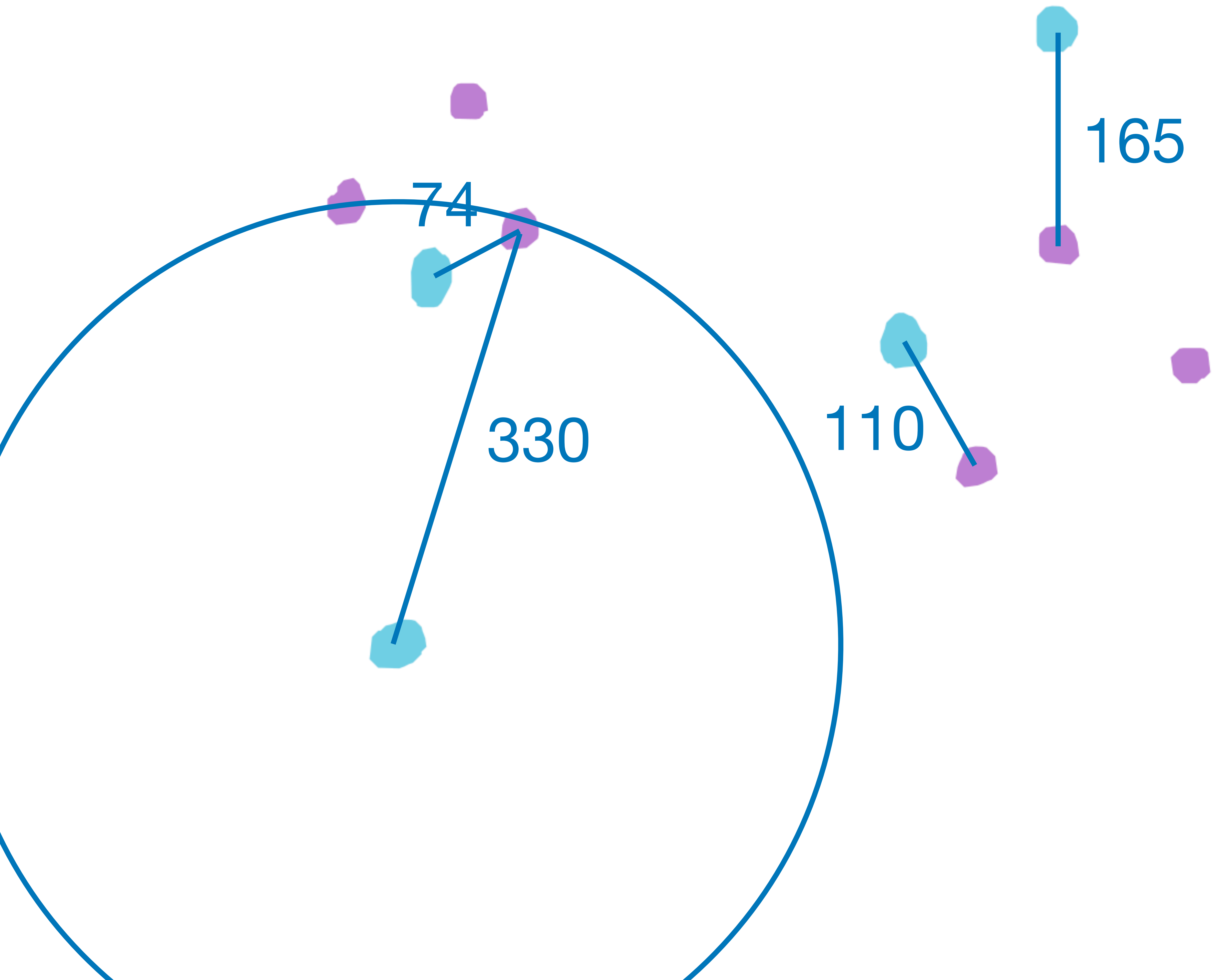
$$S(r) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$

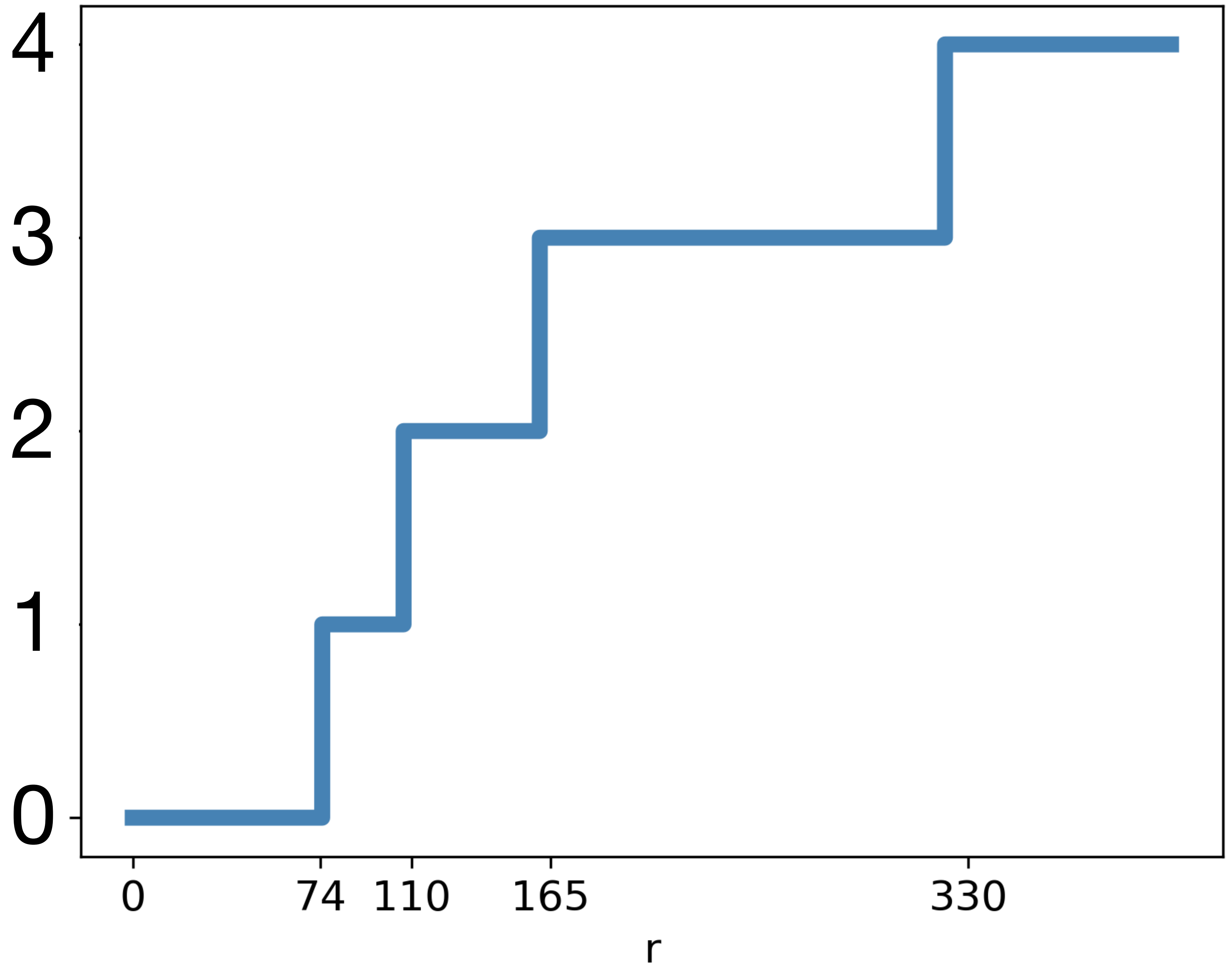




Nearest neighbor function

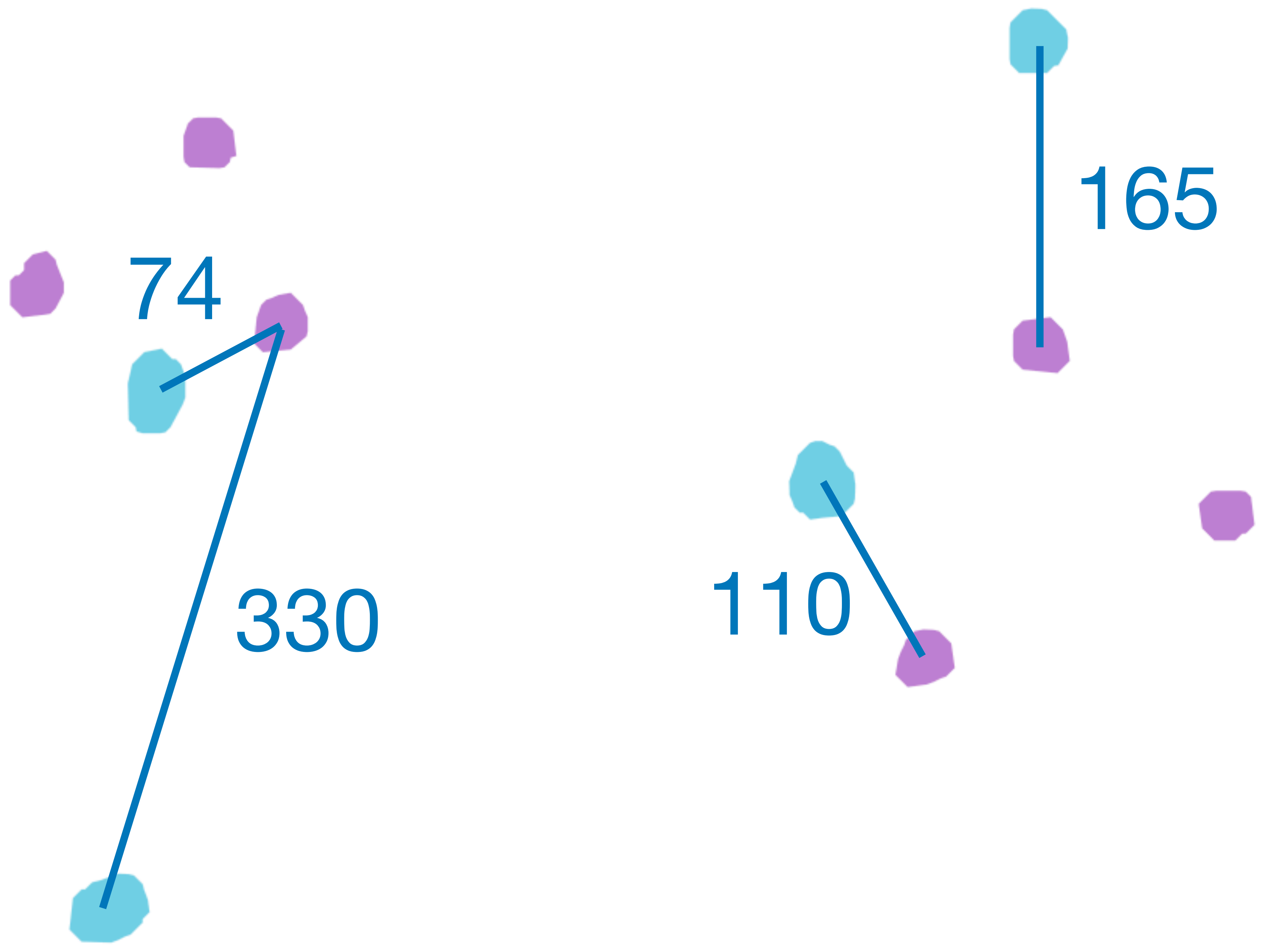
$$S(r) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$

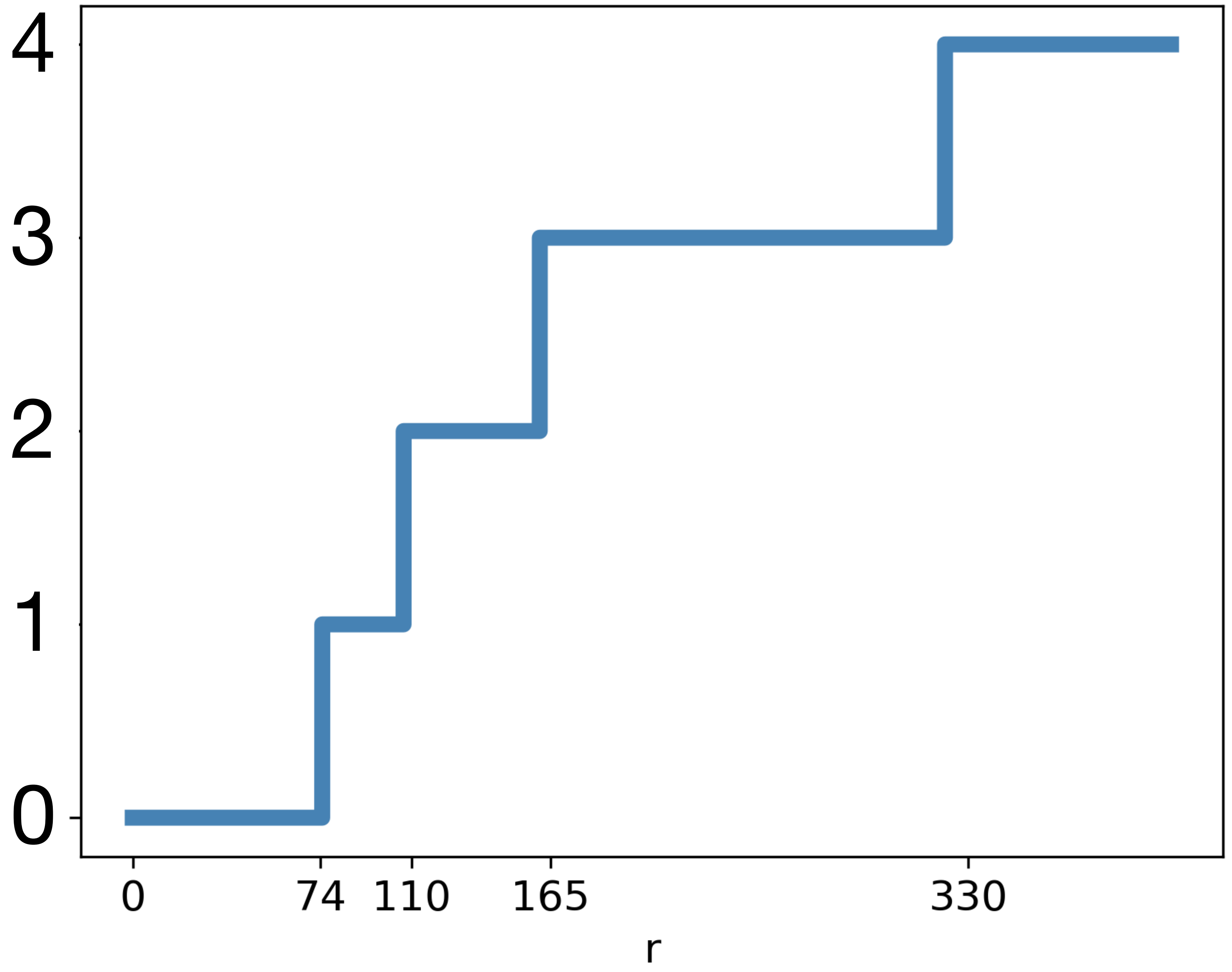




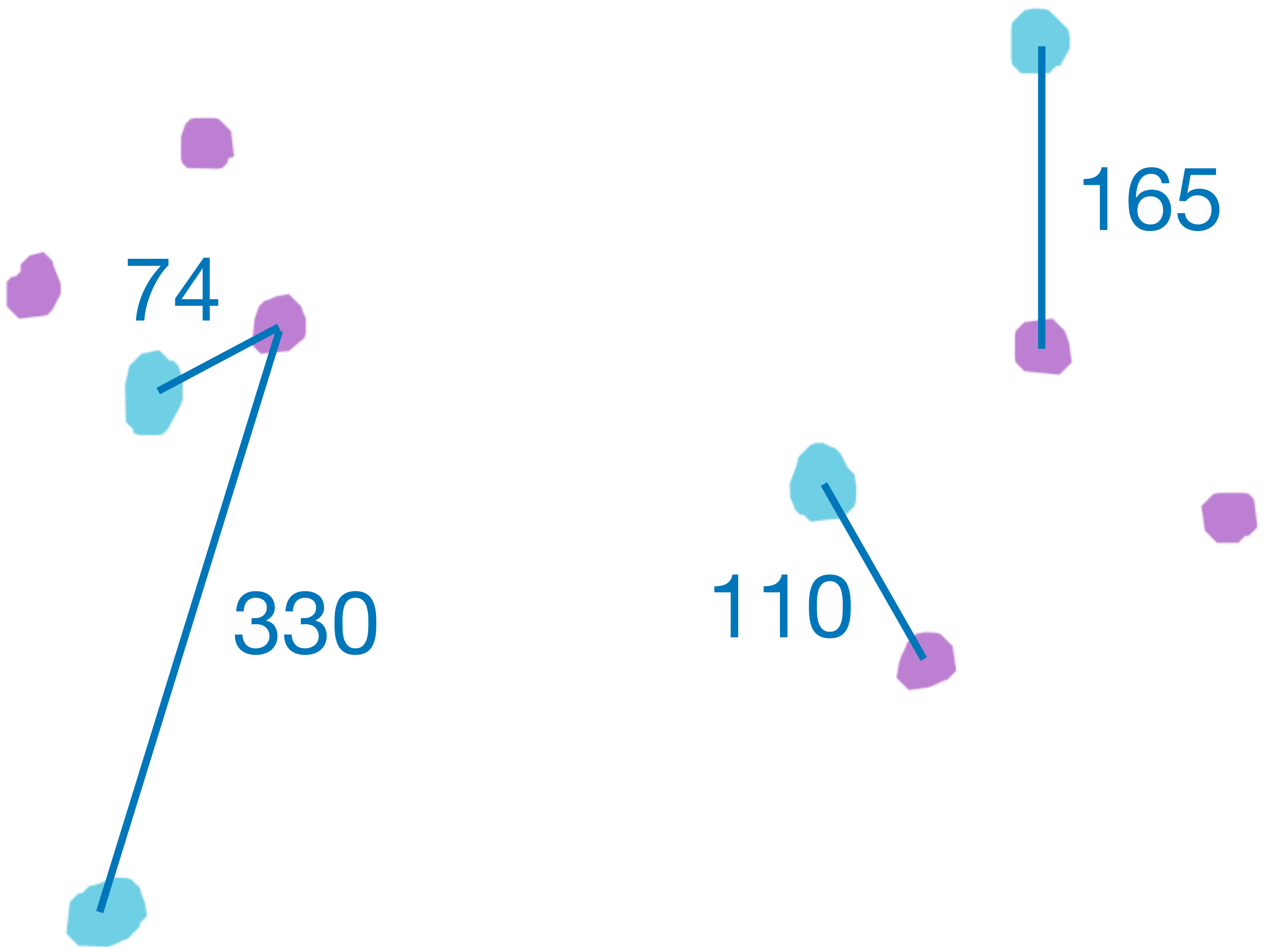
Nearest neighbor function

$$S(r) = \frac{1}{n_1}\sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$

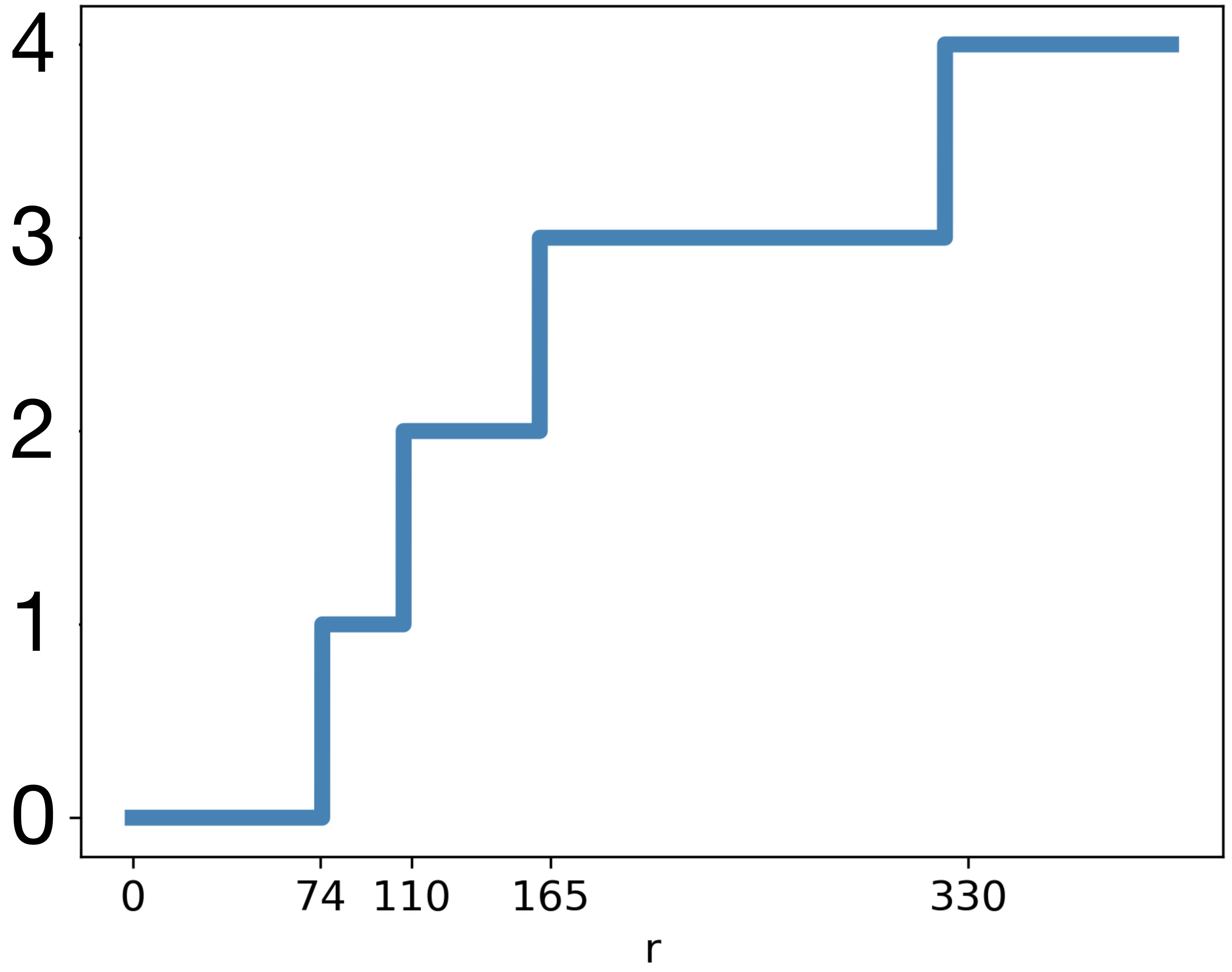




Nearest neighbor function

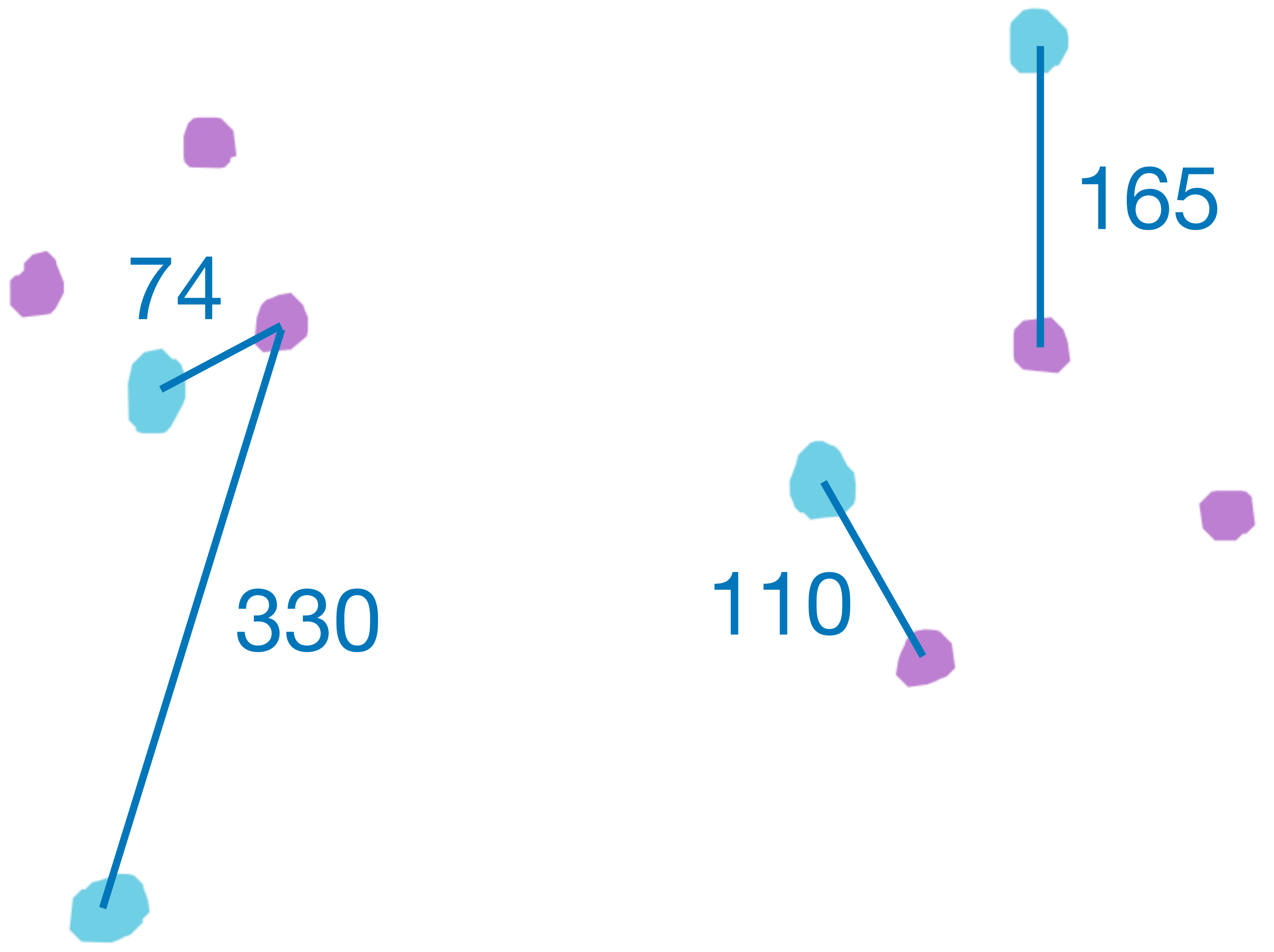


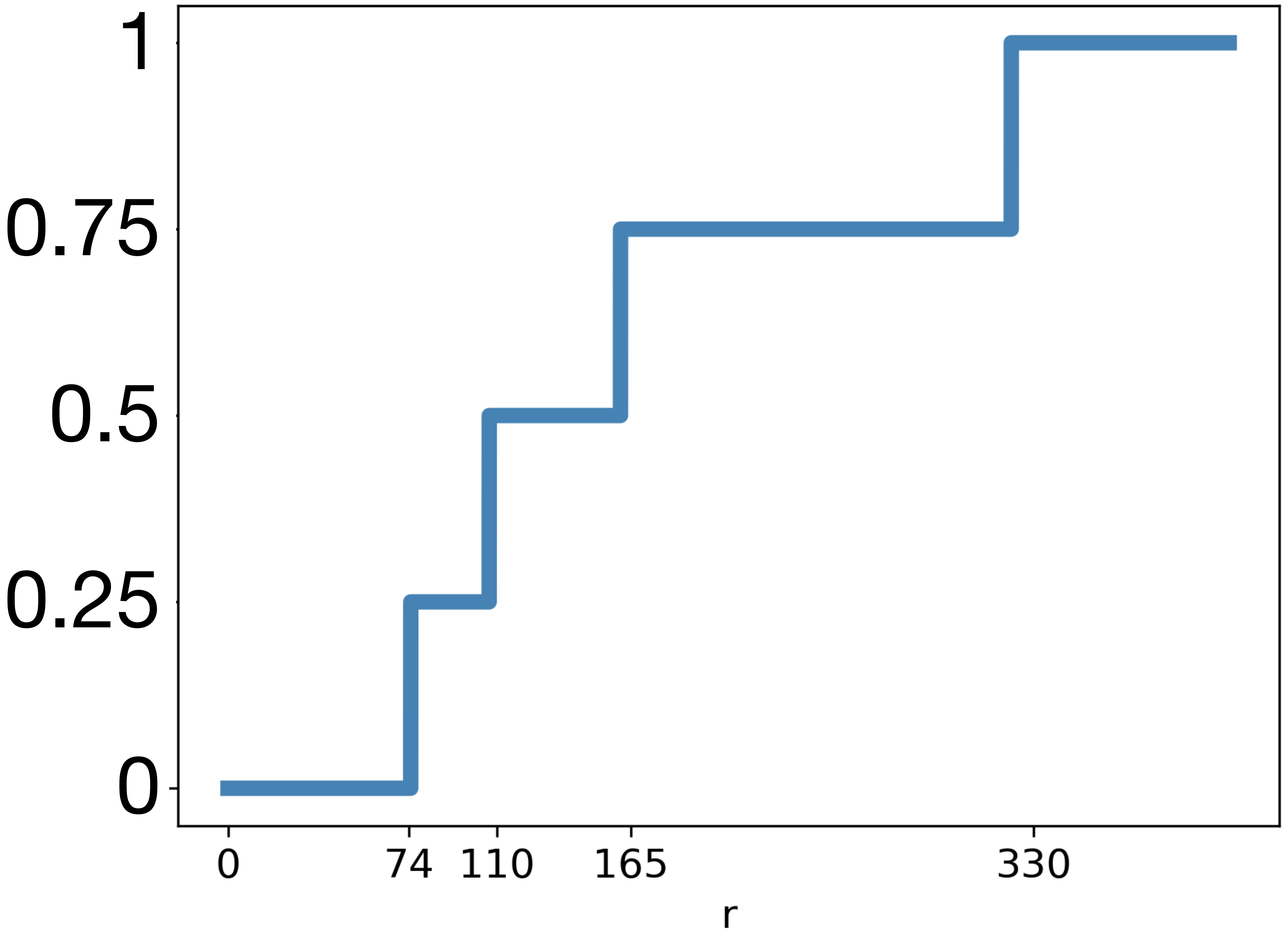
$$S(r) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$



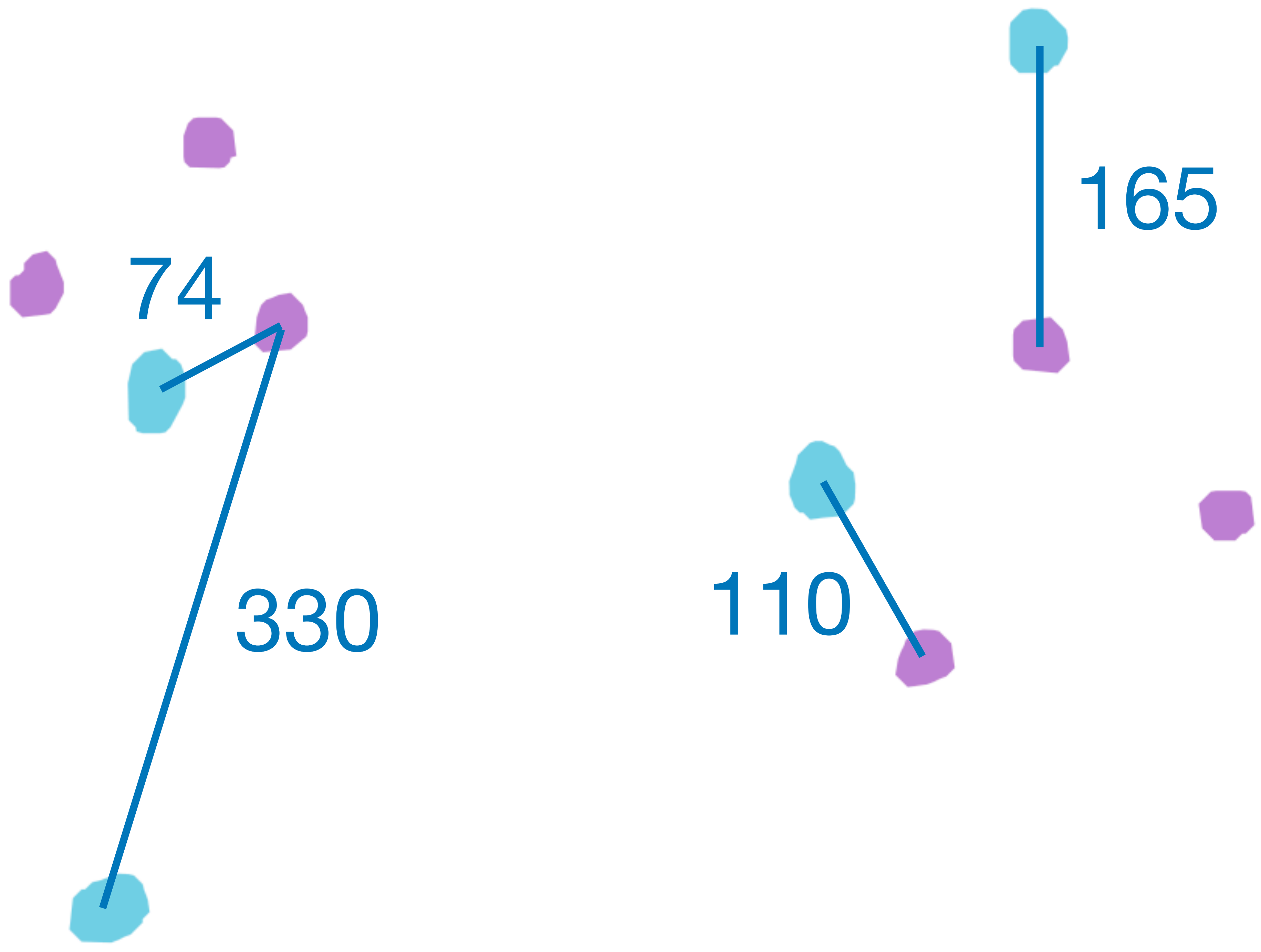
Nearest neighbor function

$$S(r) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$

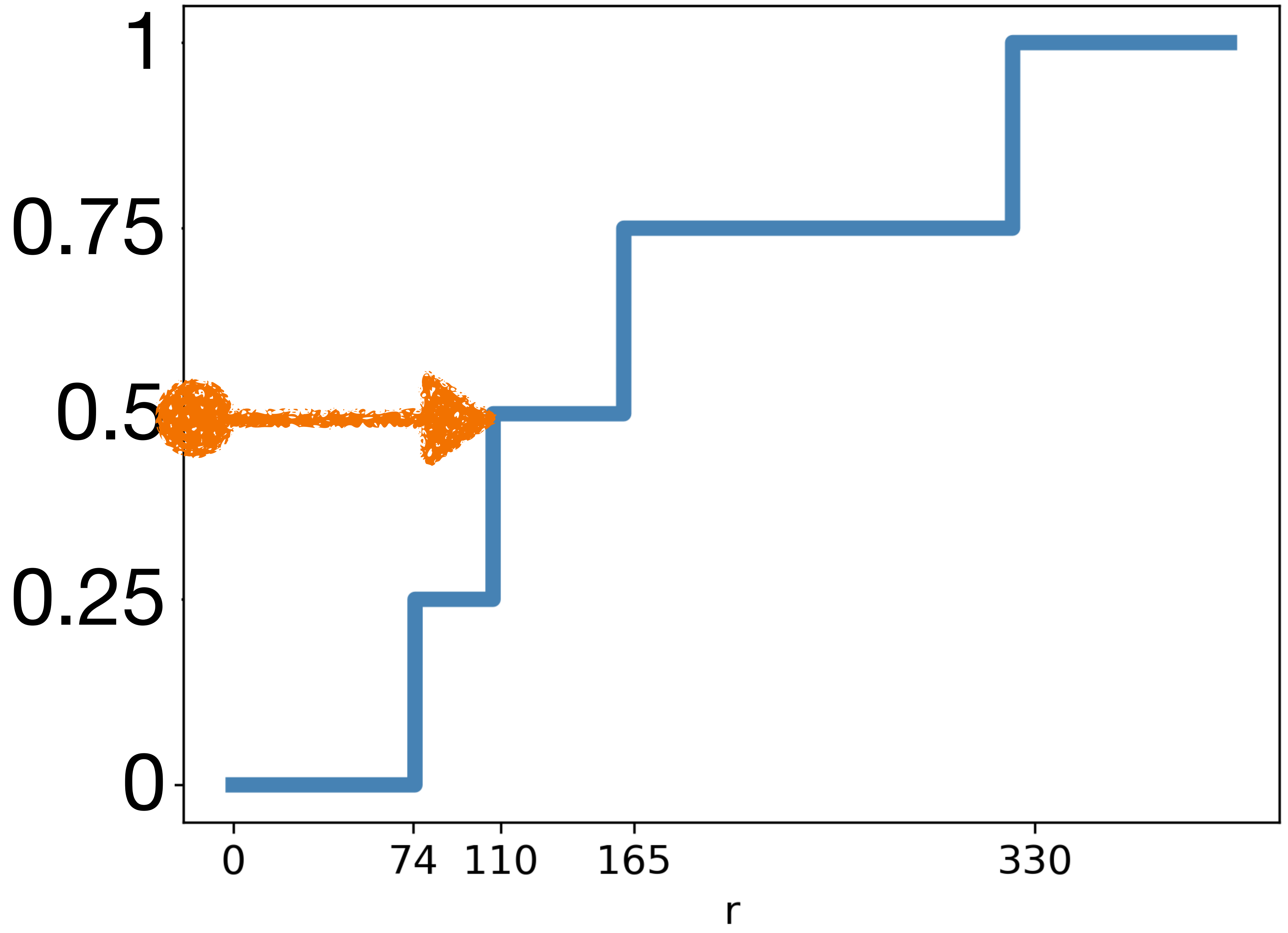




Nearest neighbor function

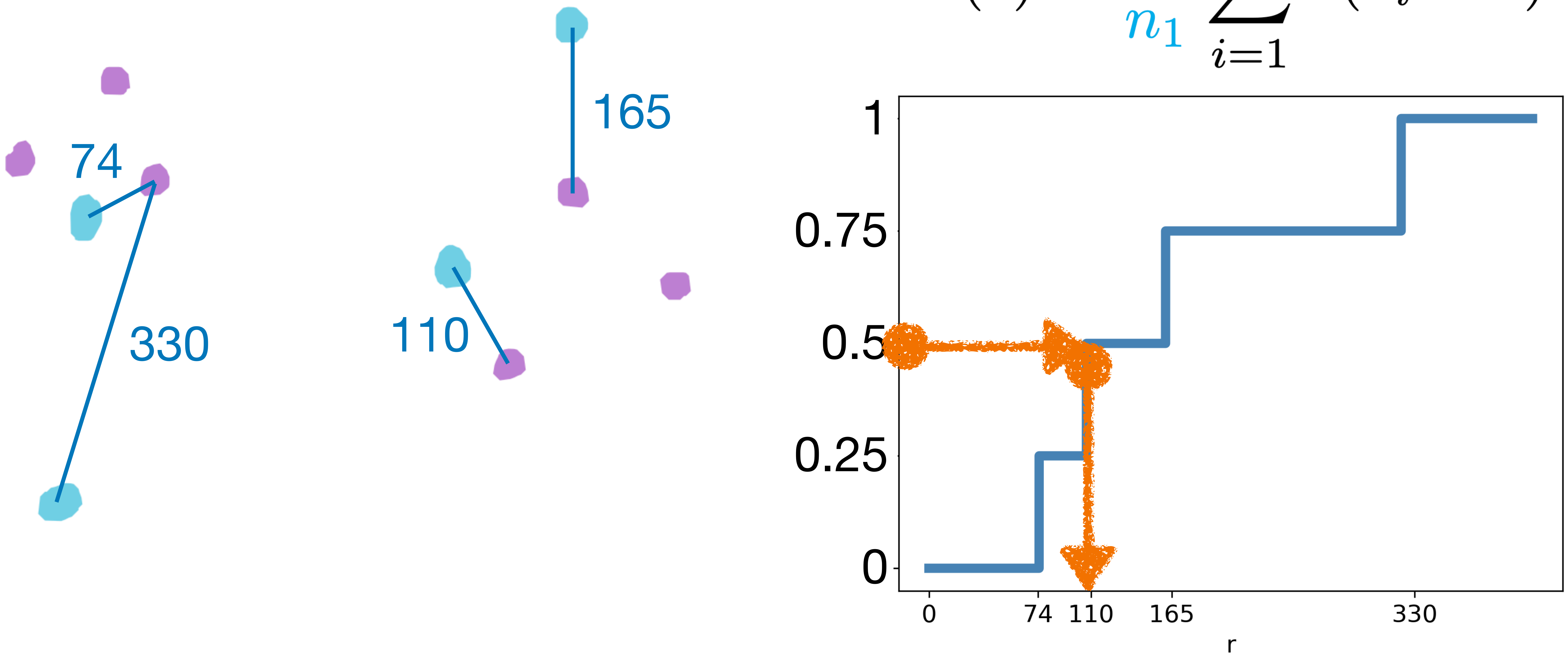


$$S(r) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$



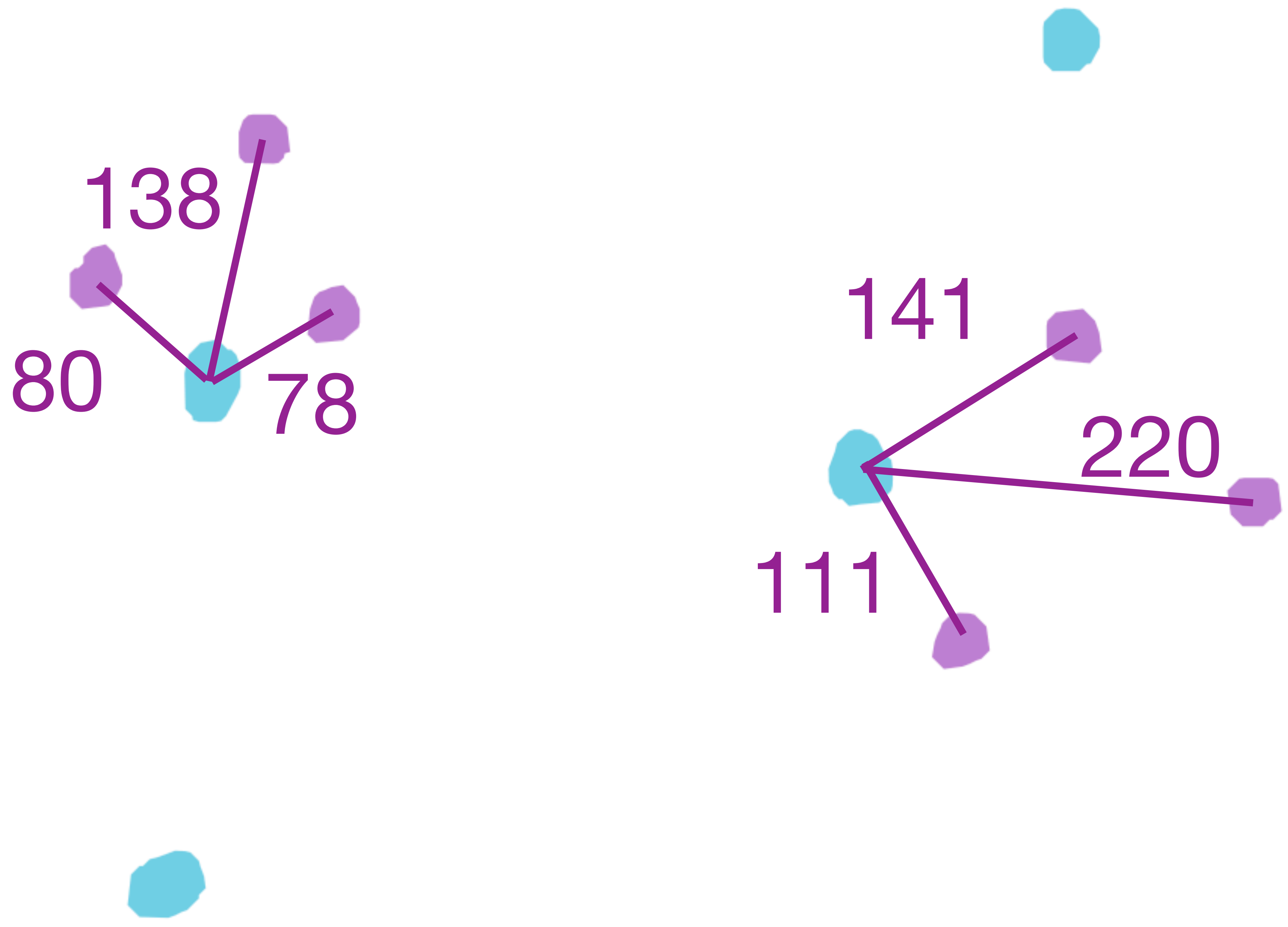
Nearest neighbor function

$$S(r) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$

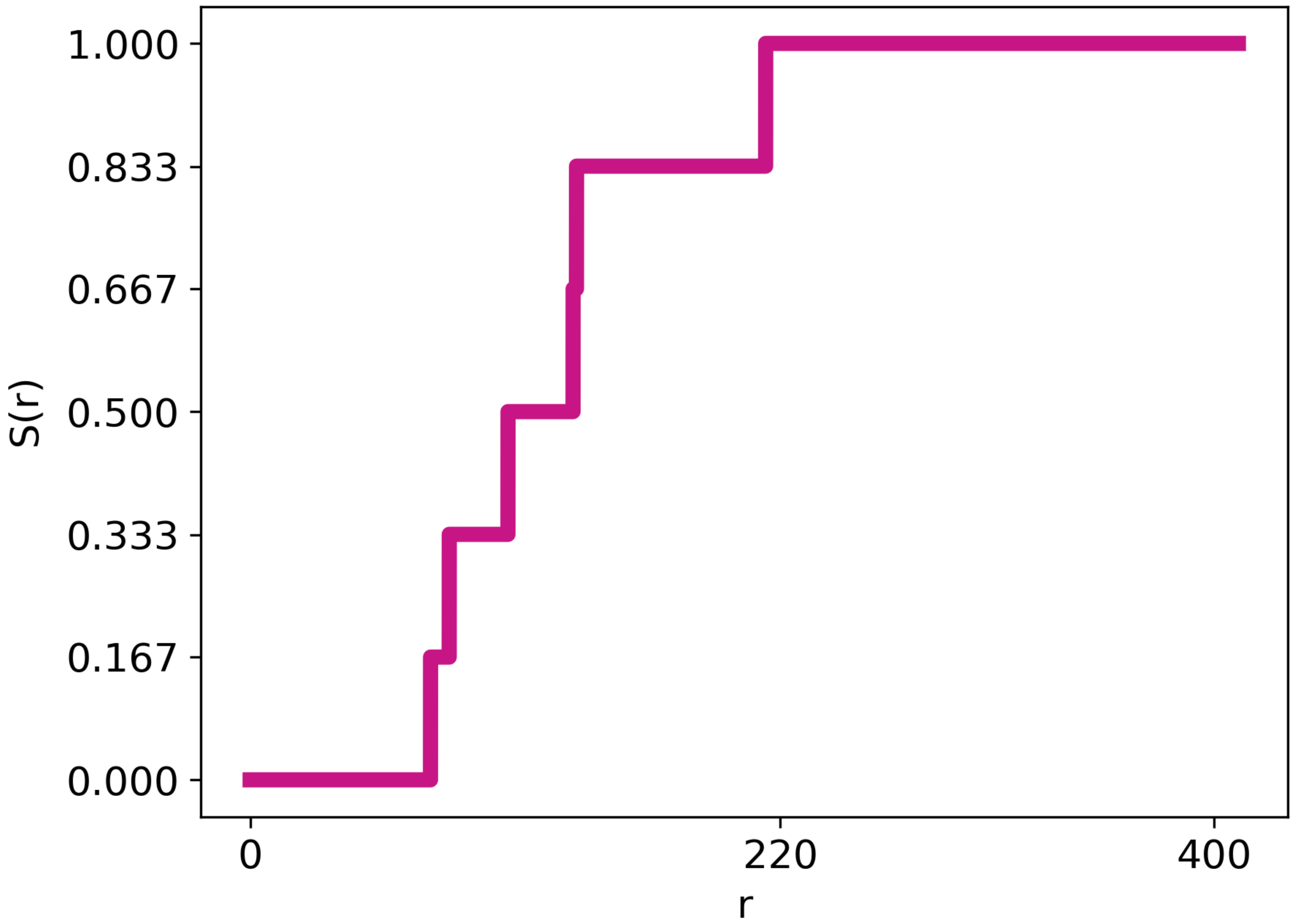


Nearest neighbor function

$$S(r) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{1}(d_i < r)$$



$$n_1 = 6$$

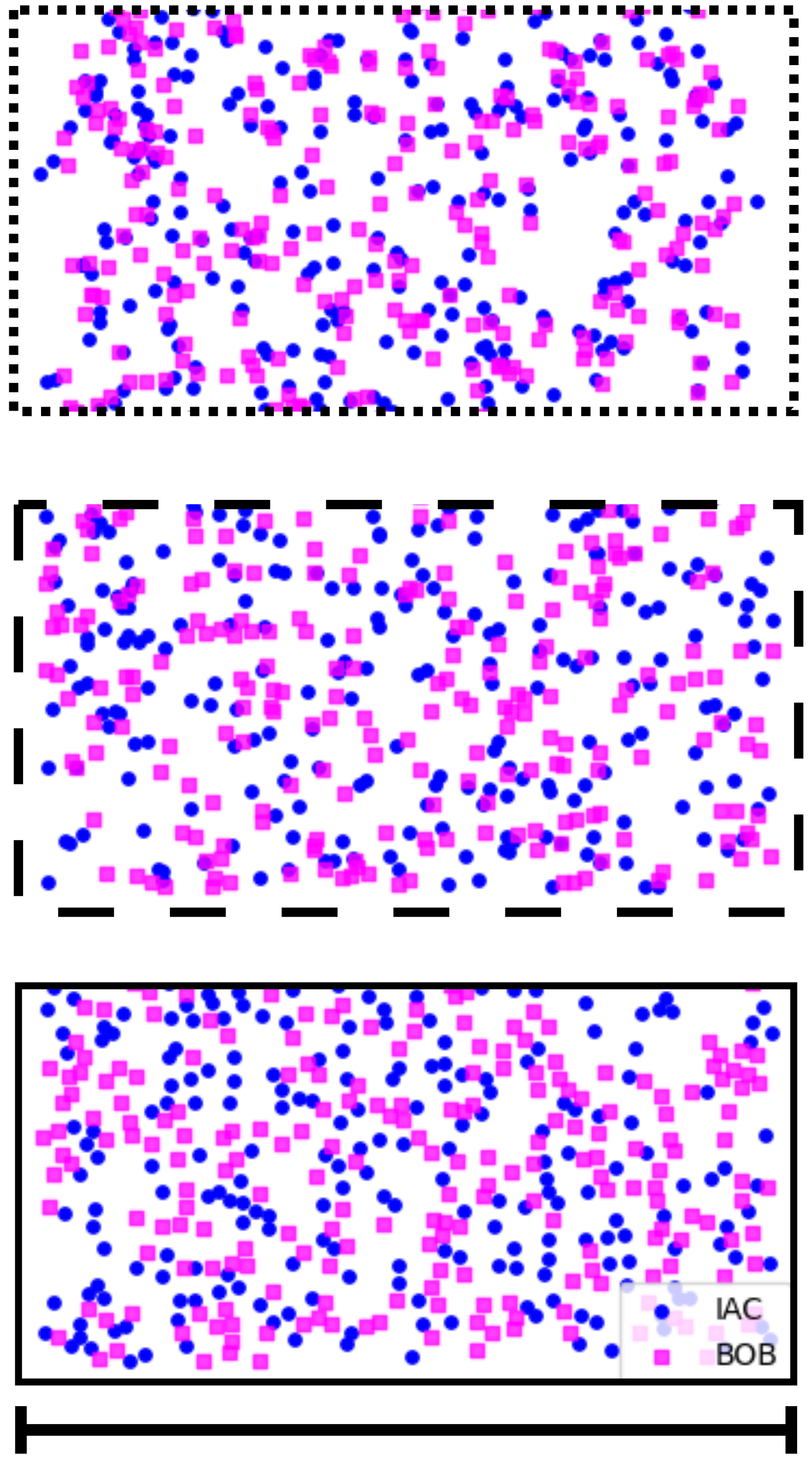


Results: Nearest neighbor function

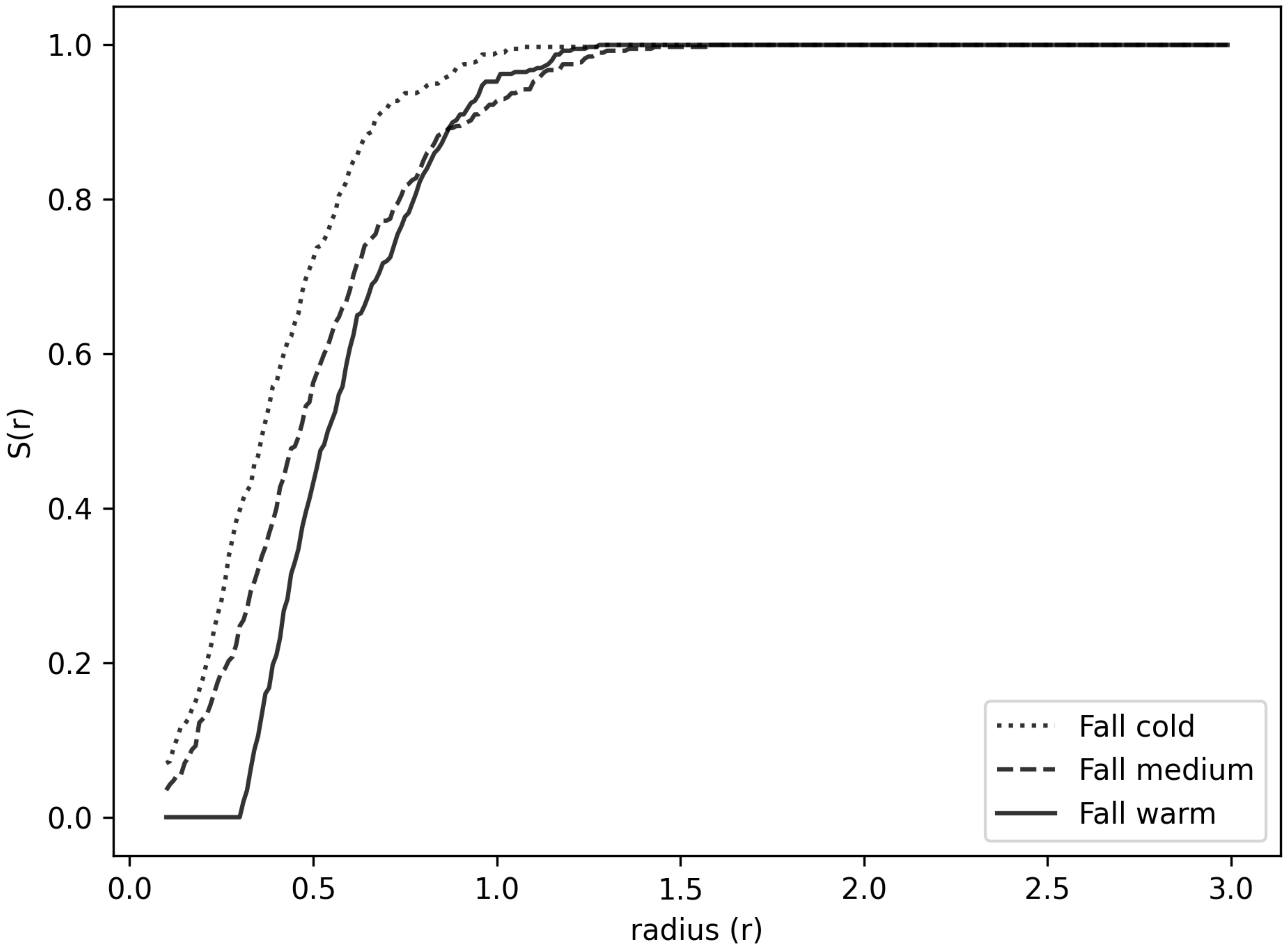
Fall — cold

Fall — medium

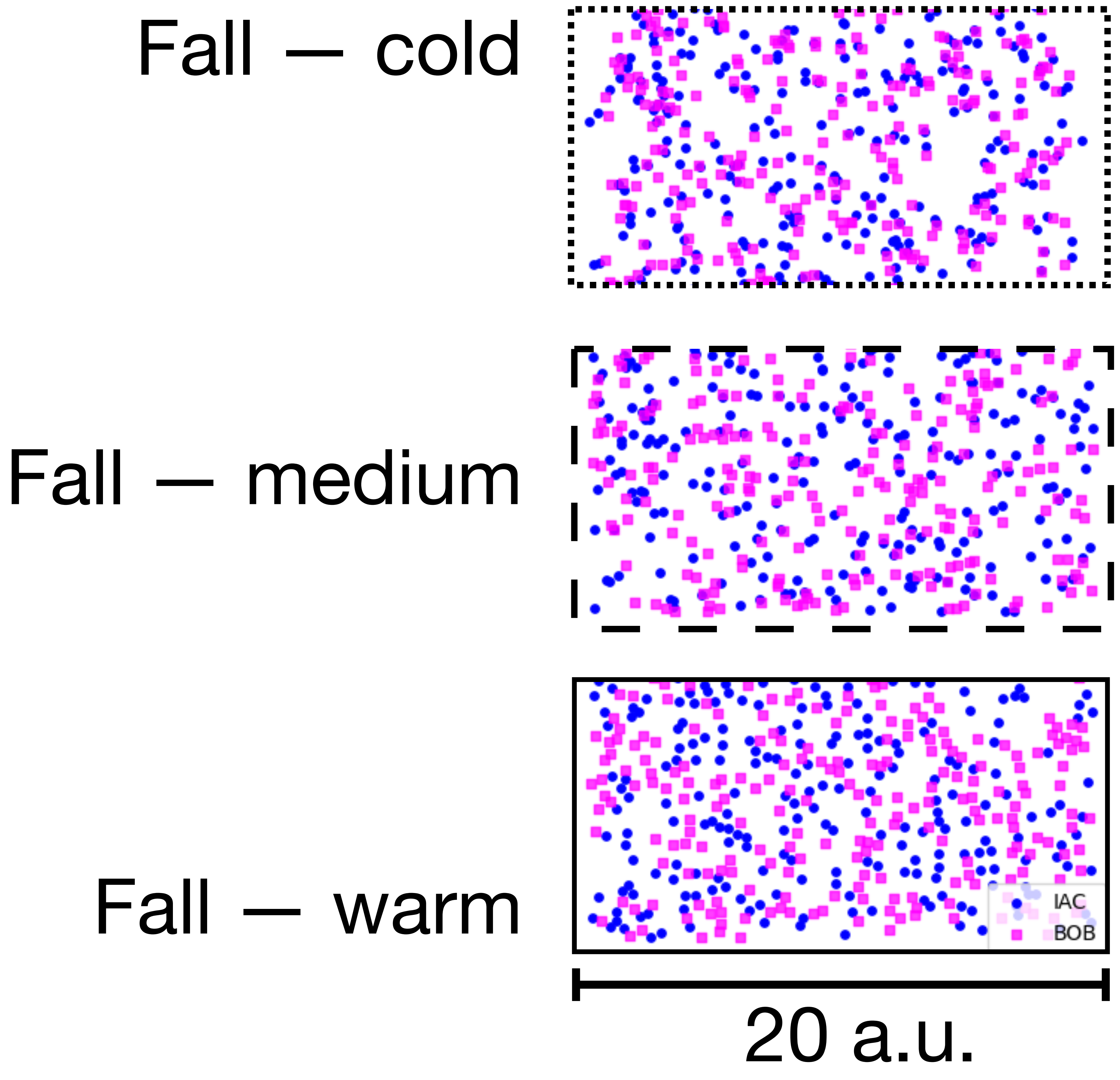
Fall — warm

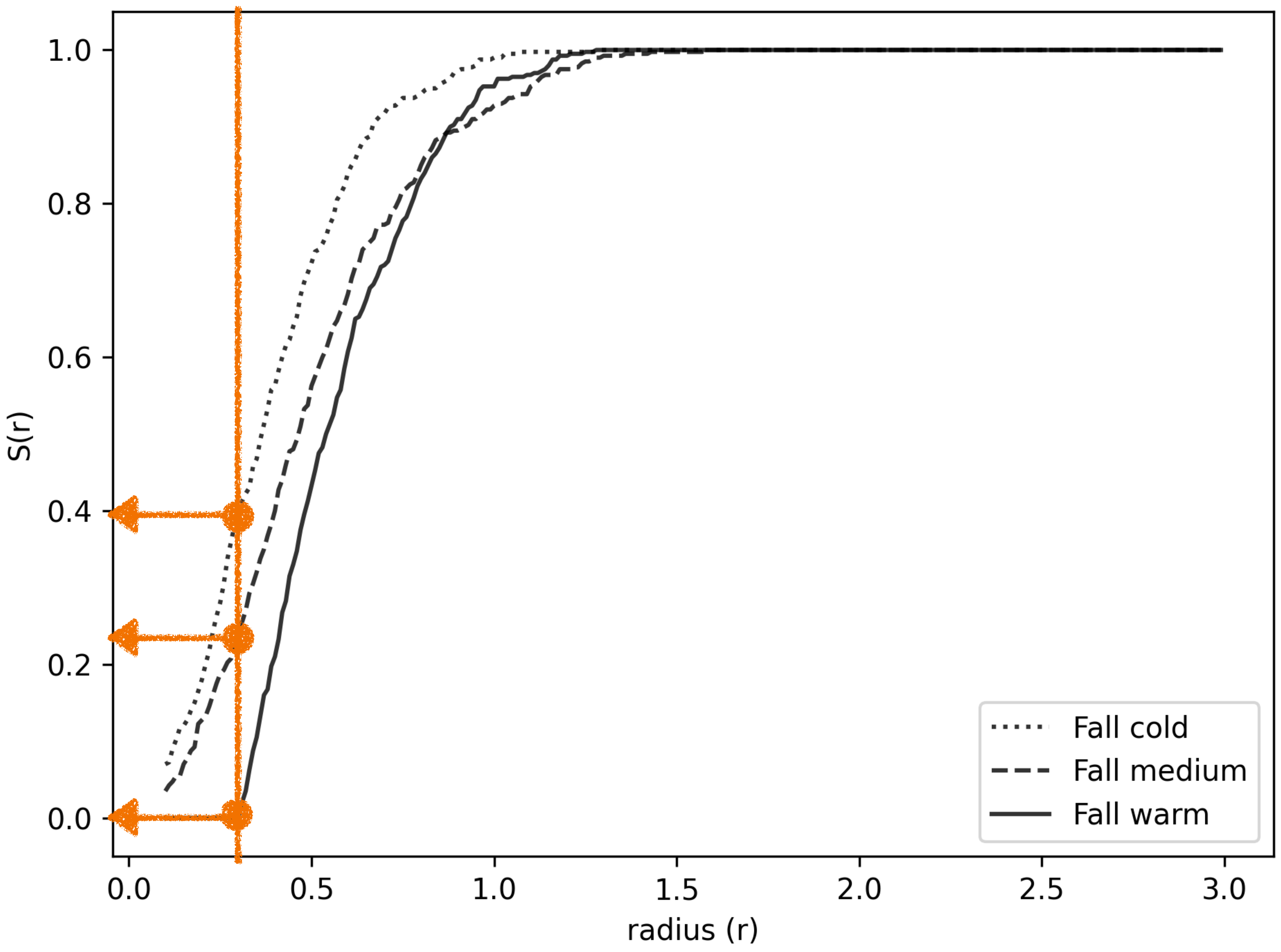


20 a.u.



Results: Nearest neighbor function



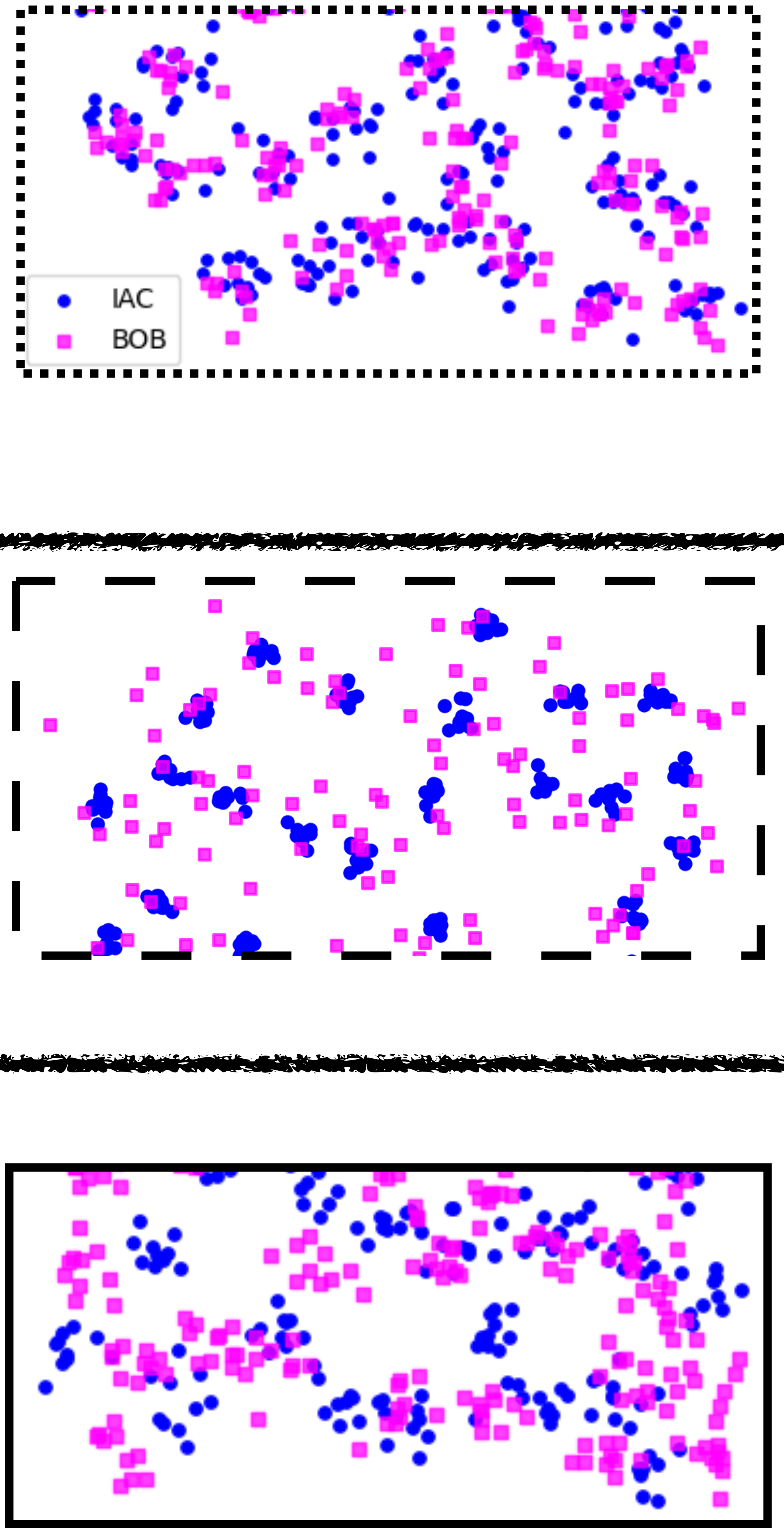


Results: Nearest neighbor function

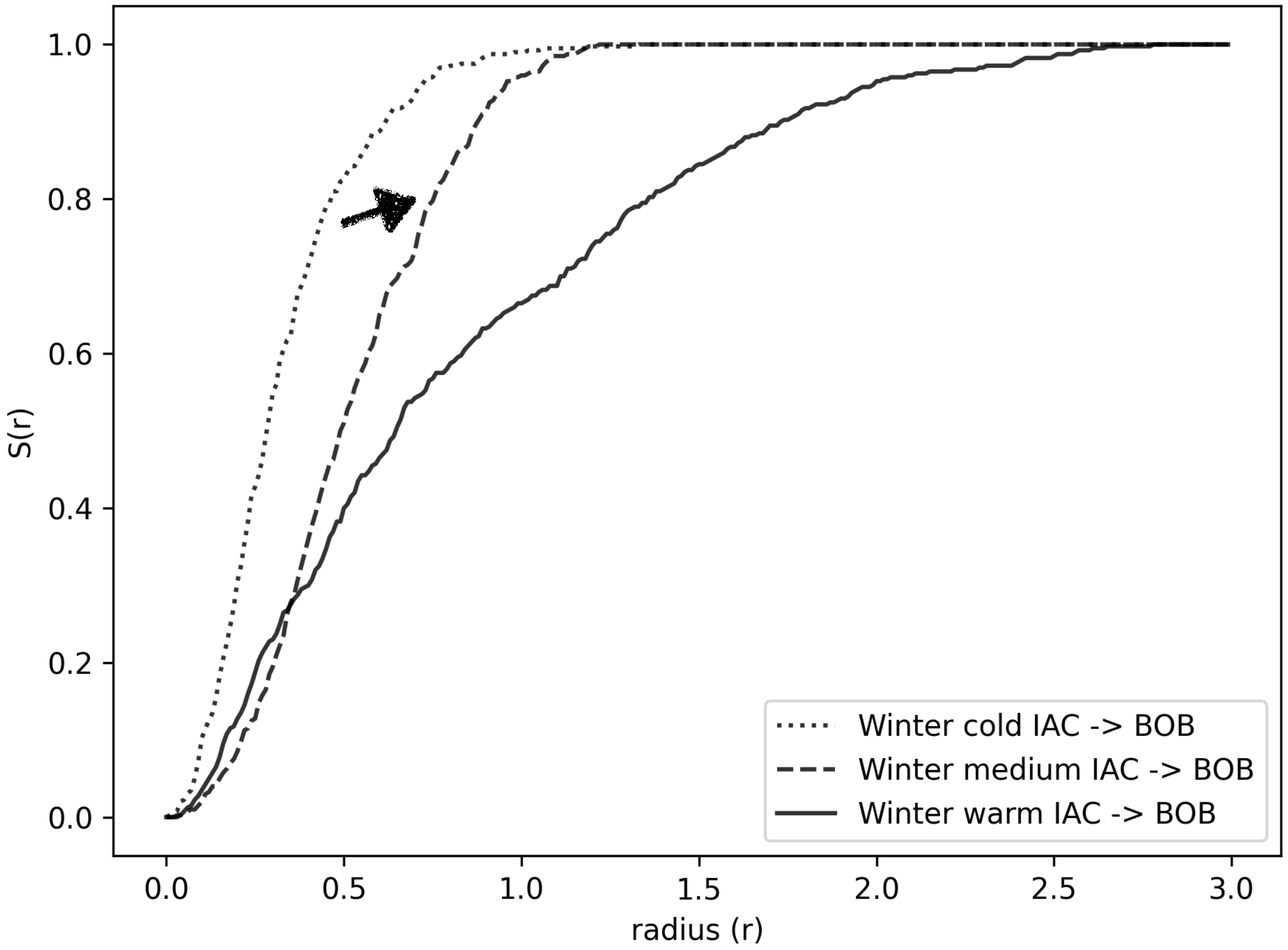
Winter — cold

Winter — medium

Winter — warm

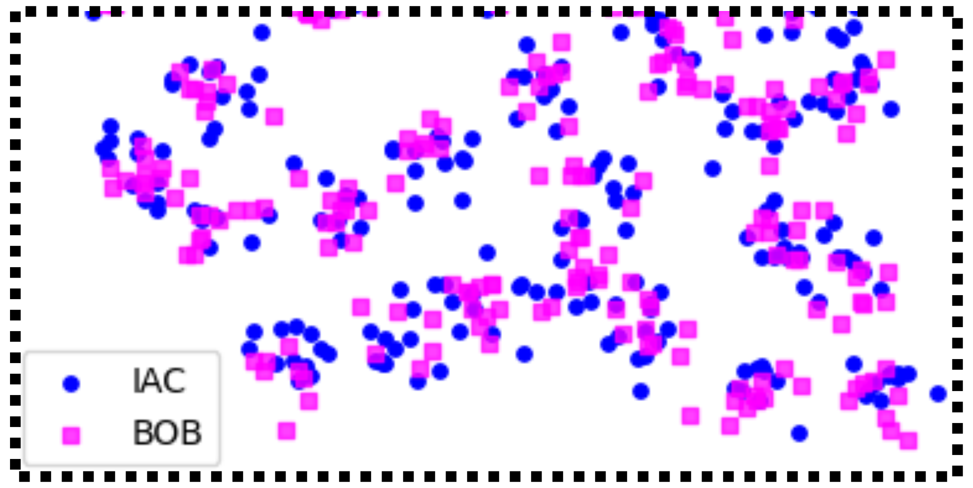


IAC → BOB

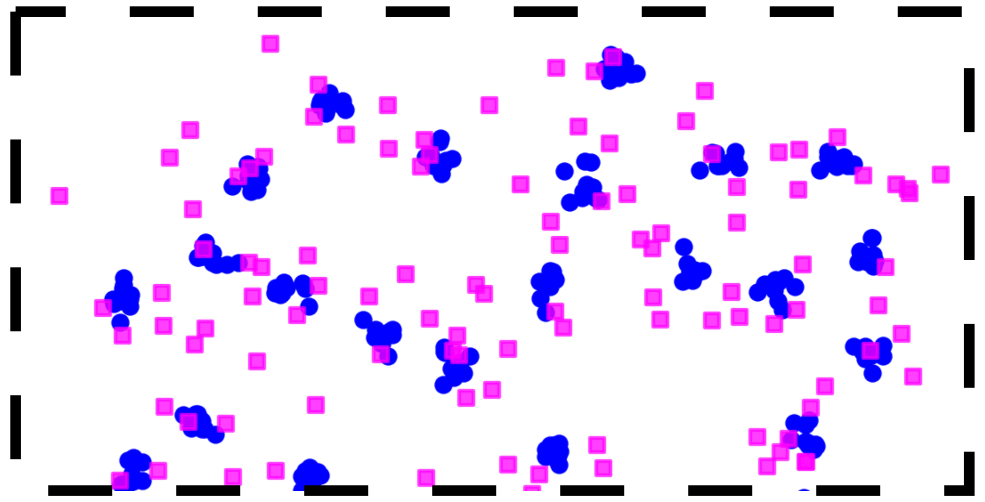


Results: Nearest neighbor function

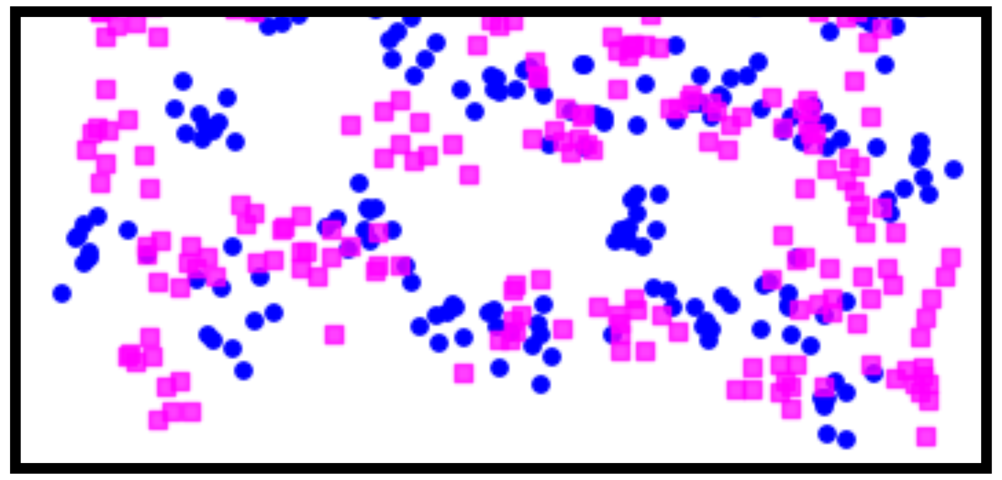
Winter — cold



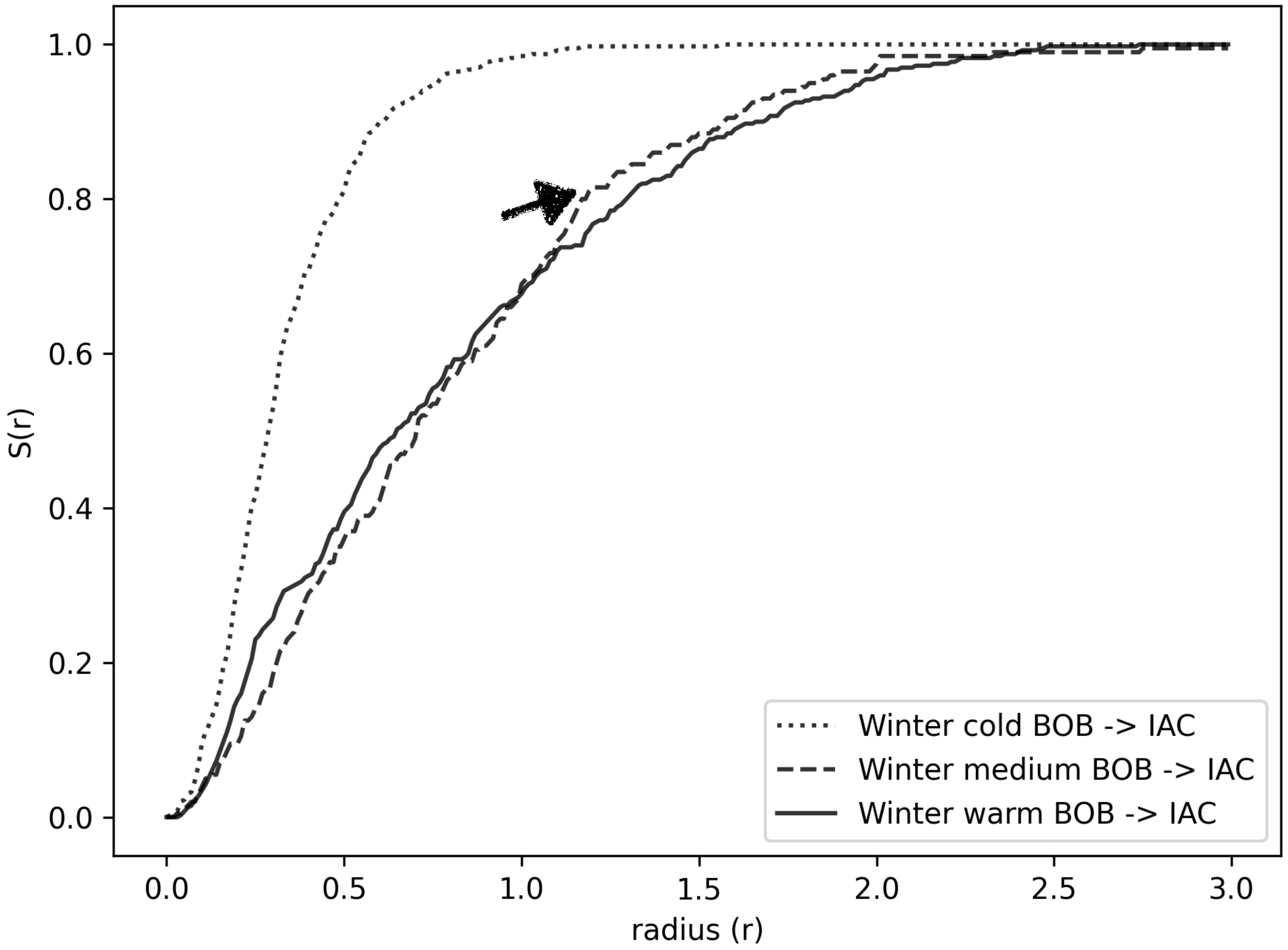
Winter — medium



Winter — warm



BOB → IAC

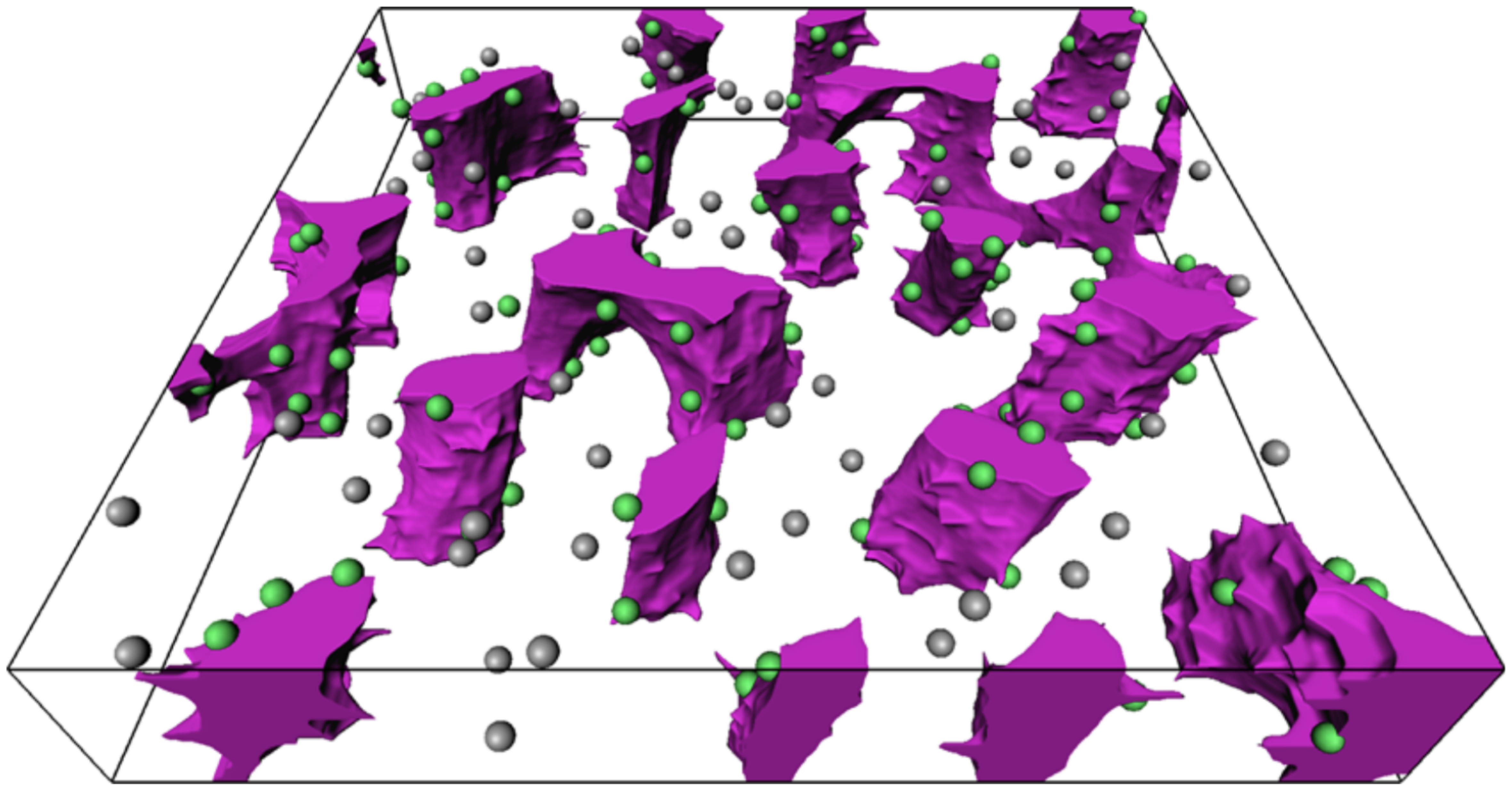


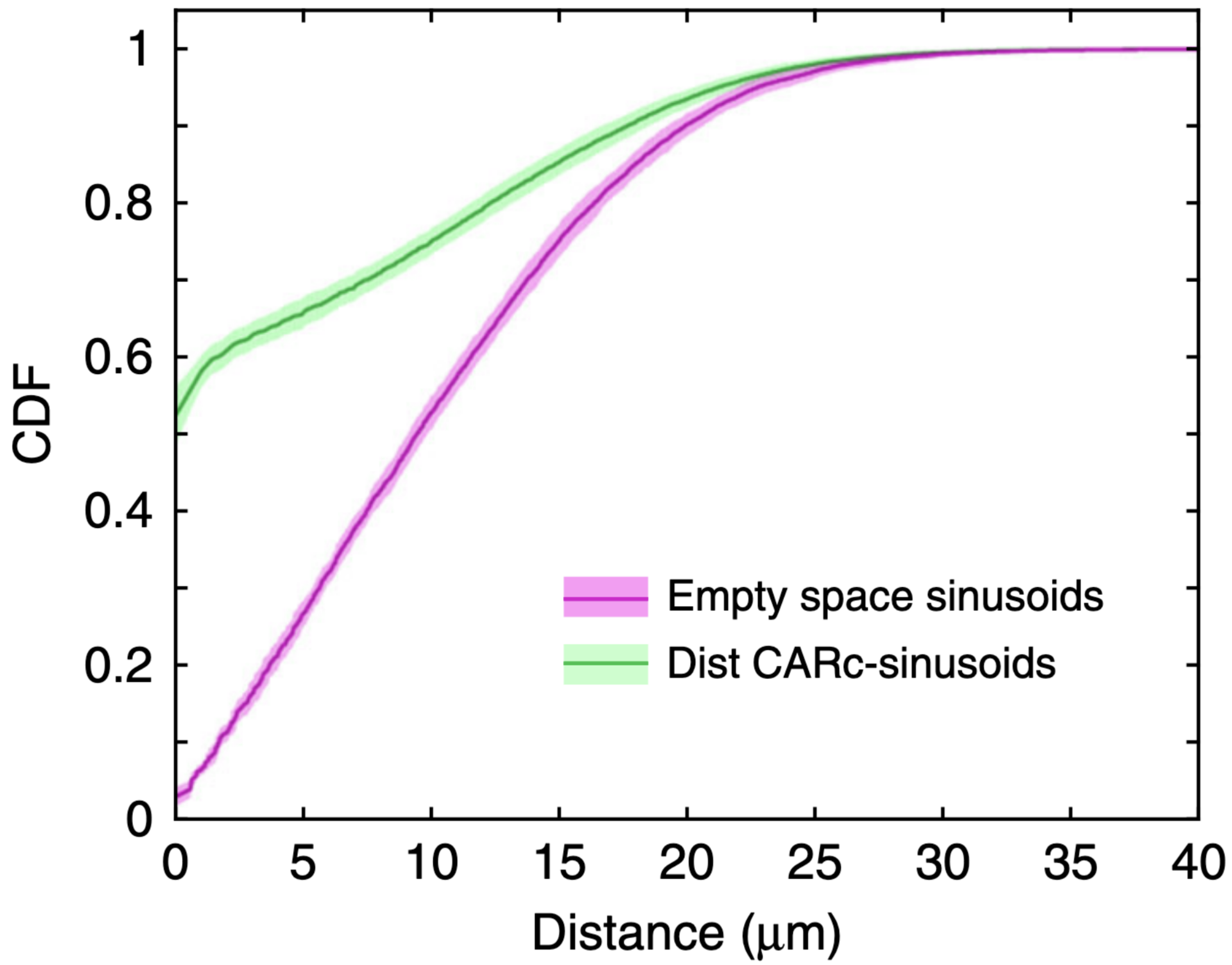
Nearest neighbor function

- Asymmetric: BOB → IAC ≠ IAC → BOB
- Returns: A number for each radius
- Range: Short

Beyond the nearest neighbor function

Green < 5 µm   Grey > 5 µm

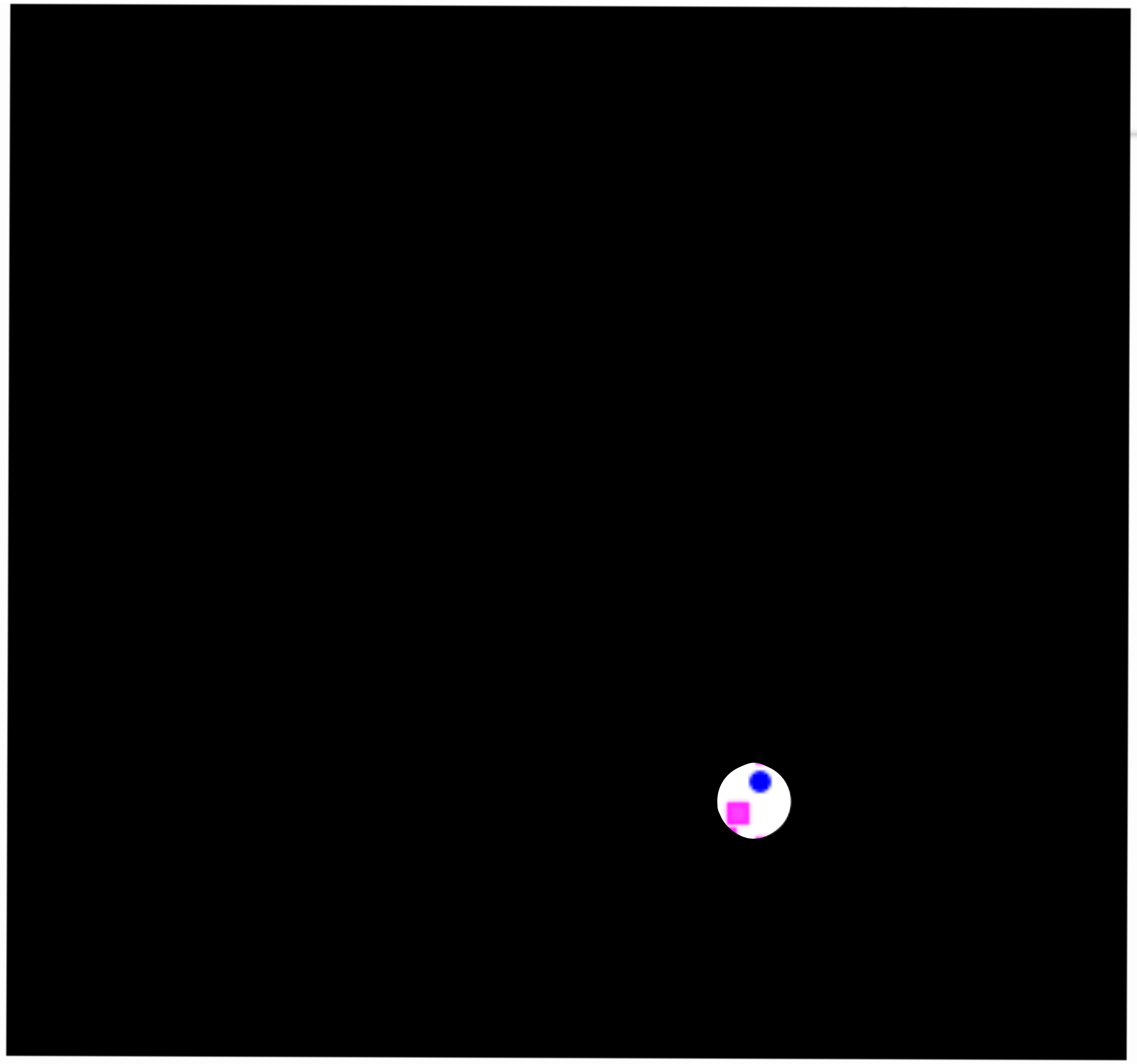




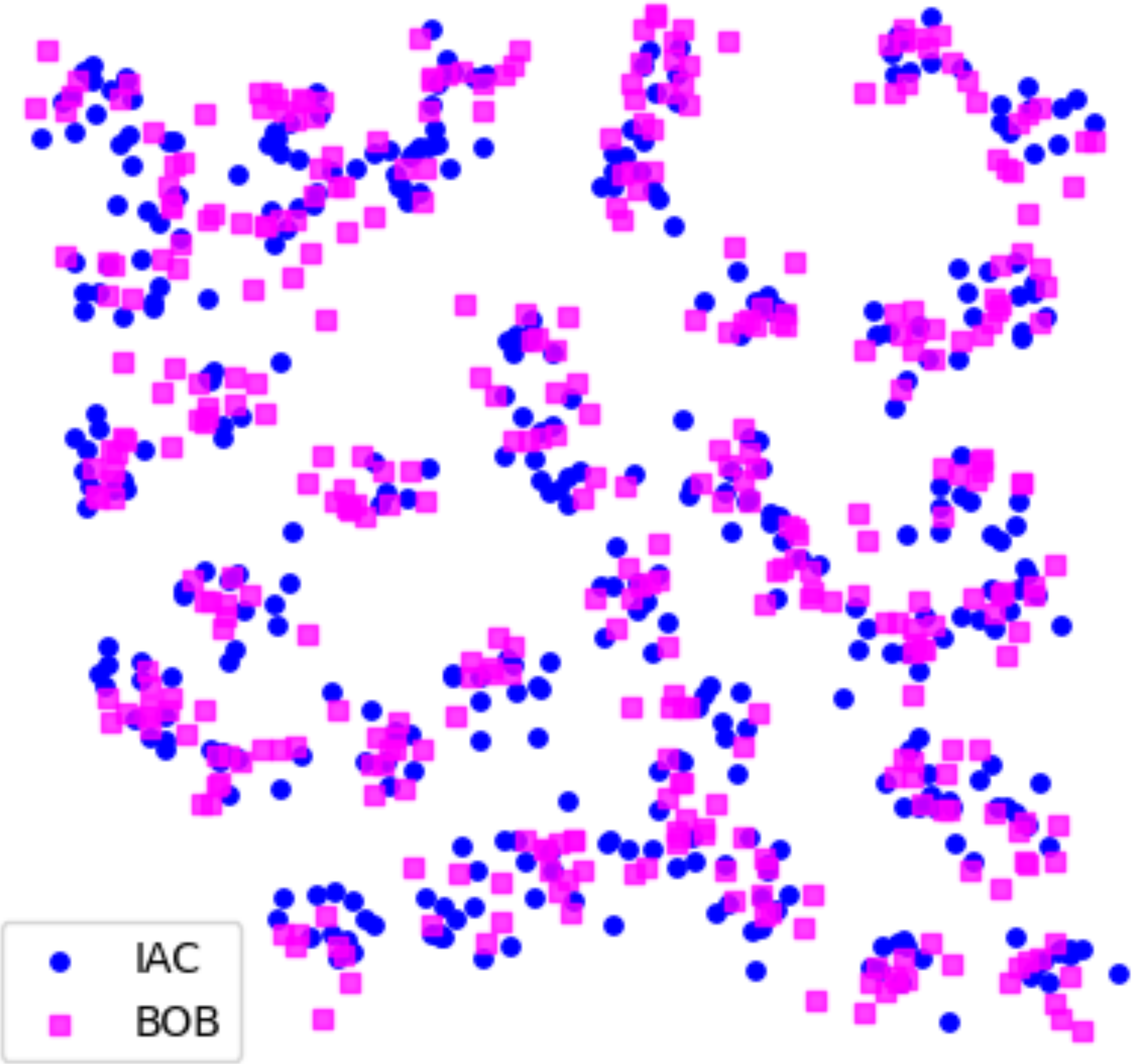
CARcs   Sinusoidal vessel wall

Modified from: Gomariz, A., Helbling, P.M., Isringhausen, S. *et al.* Quantitative spatial analysis of haematopoiesis-regulating stromal cells in the bone marrow microenvironment by 3D microscopy. *Nat Commun* **9**, 2532 (2018). https://doi.org/10.1038/s41467-018-04770-z

Ripley’s K function

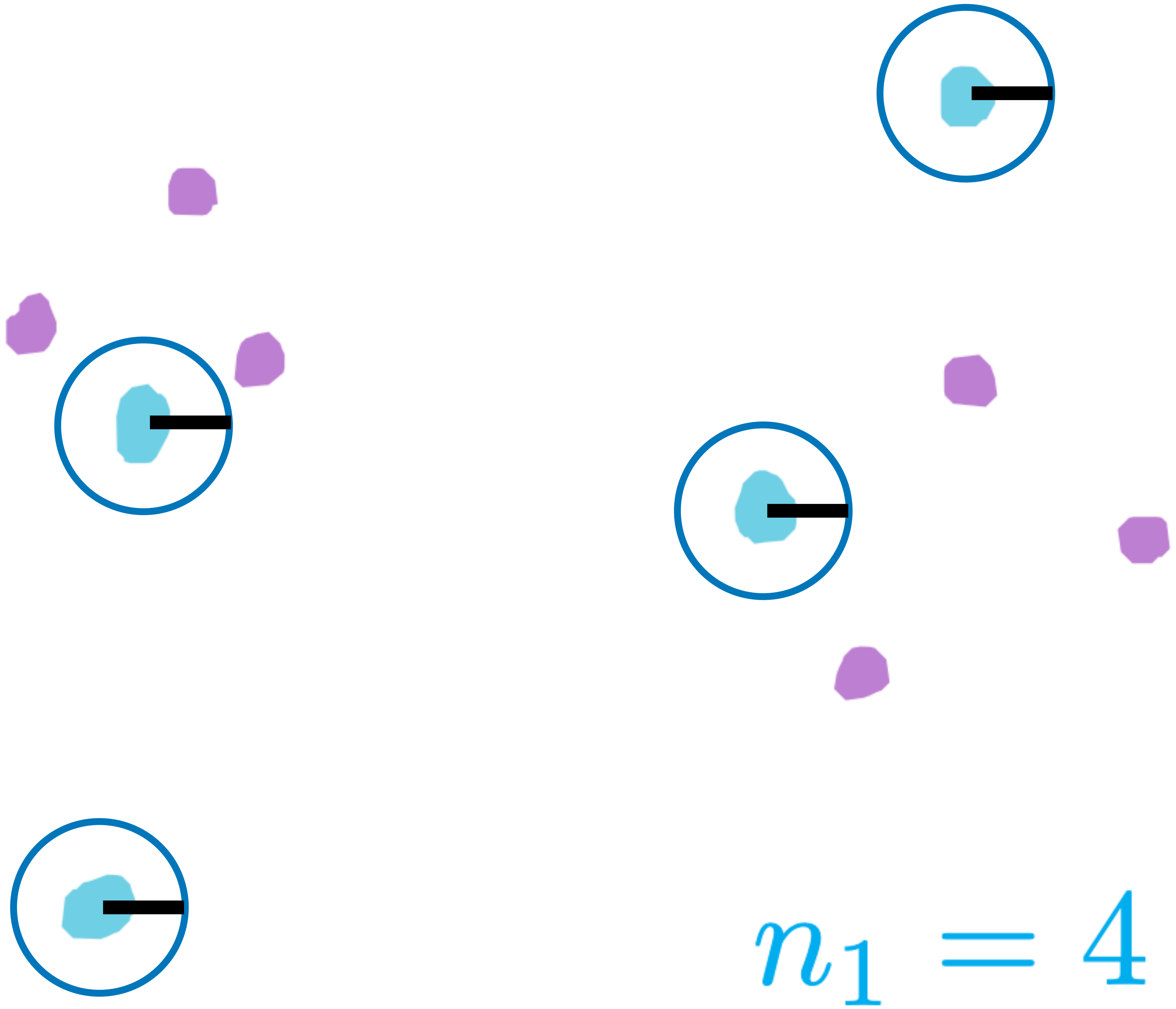


Ripley’s K function



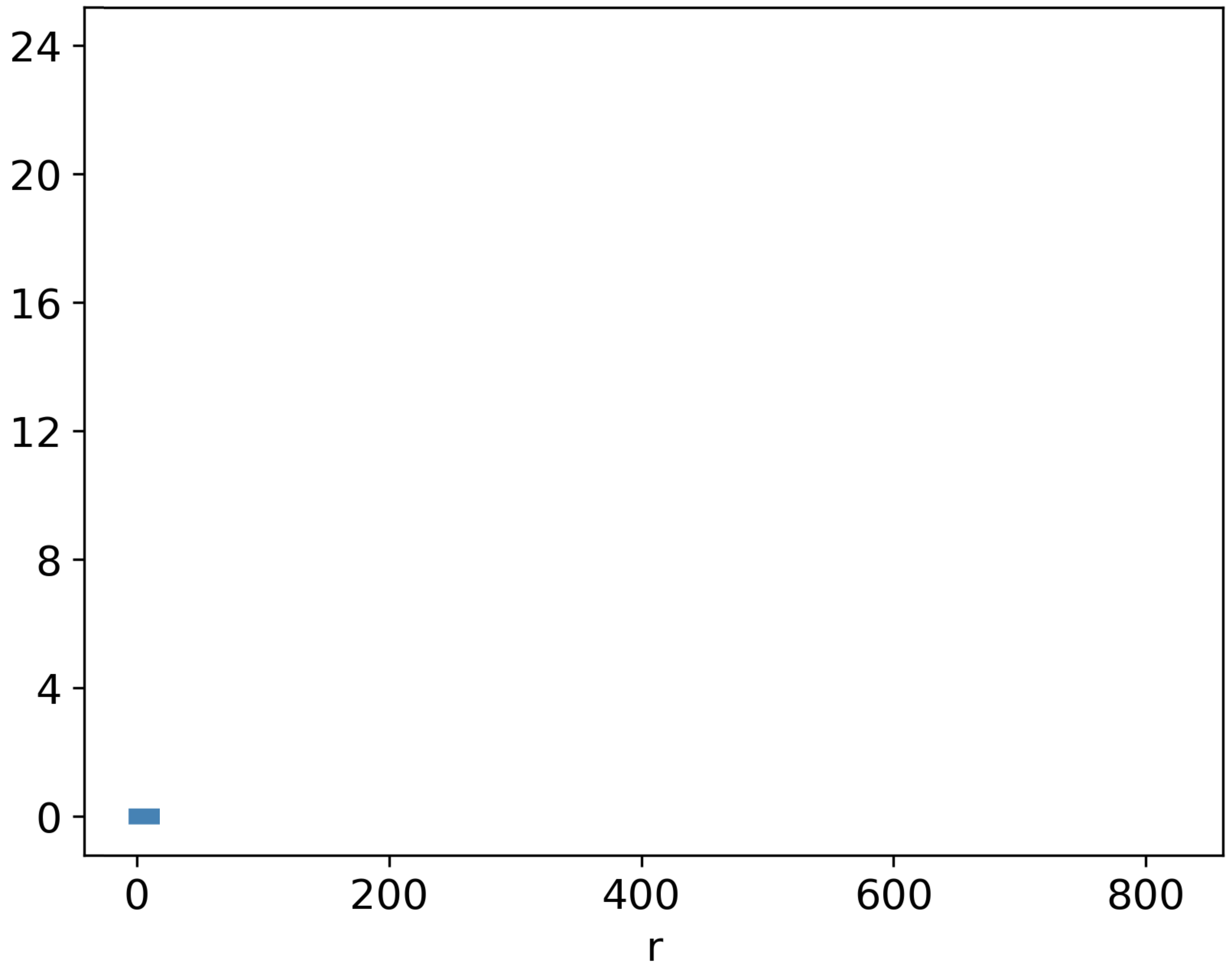
Ripley’s K function

$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r)\, b(i, j, r)$$



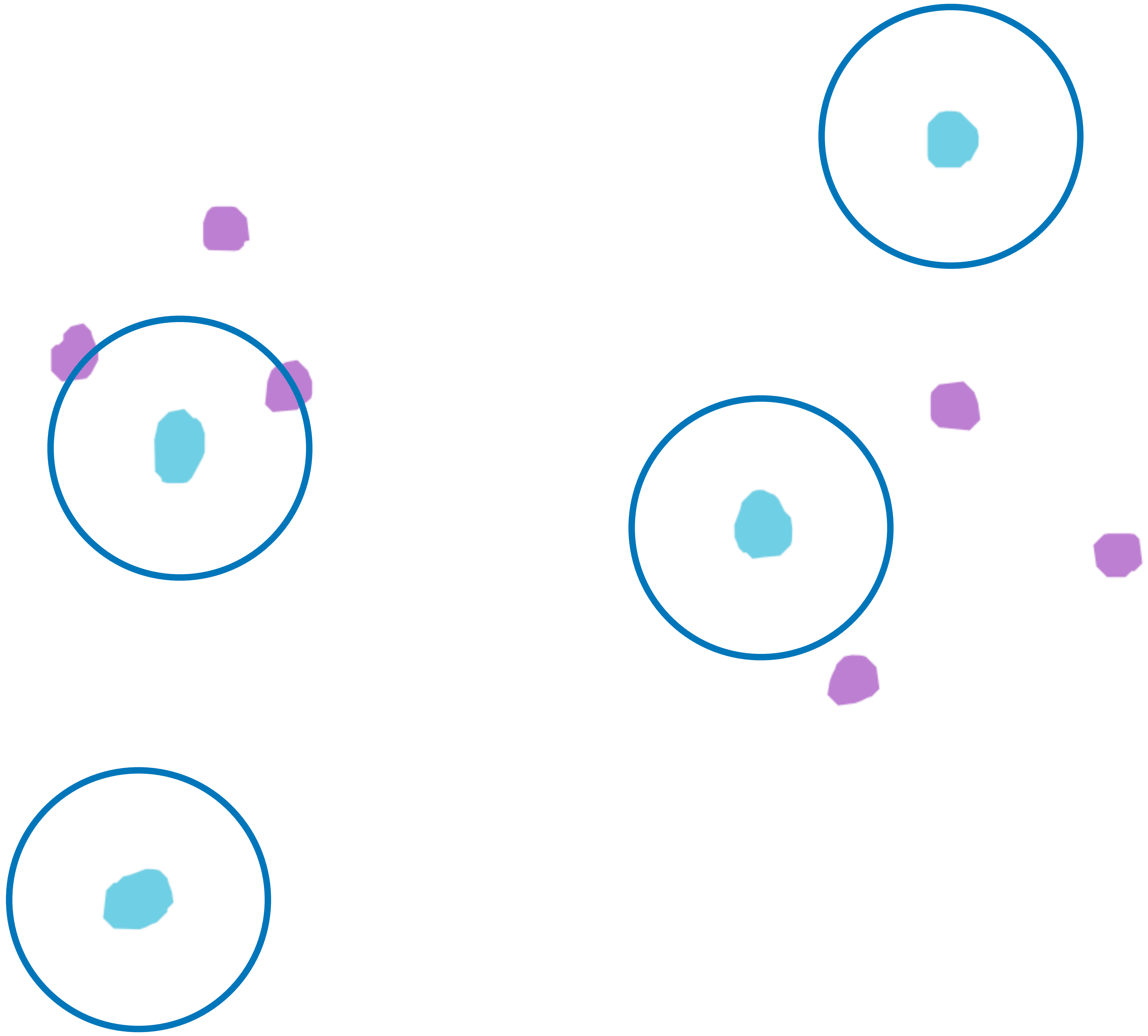
$n_1 = 4$

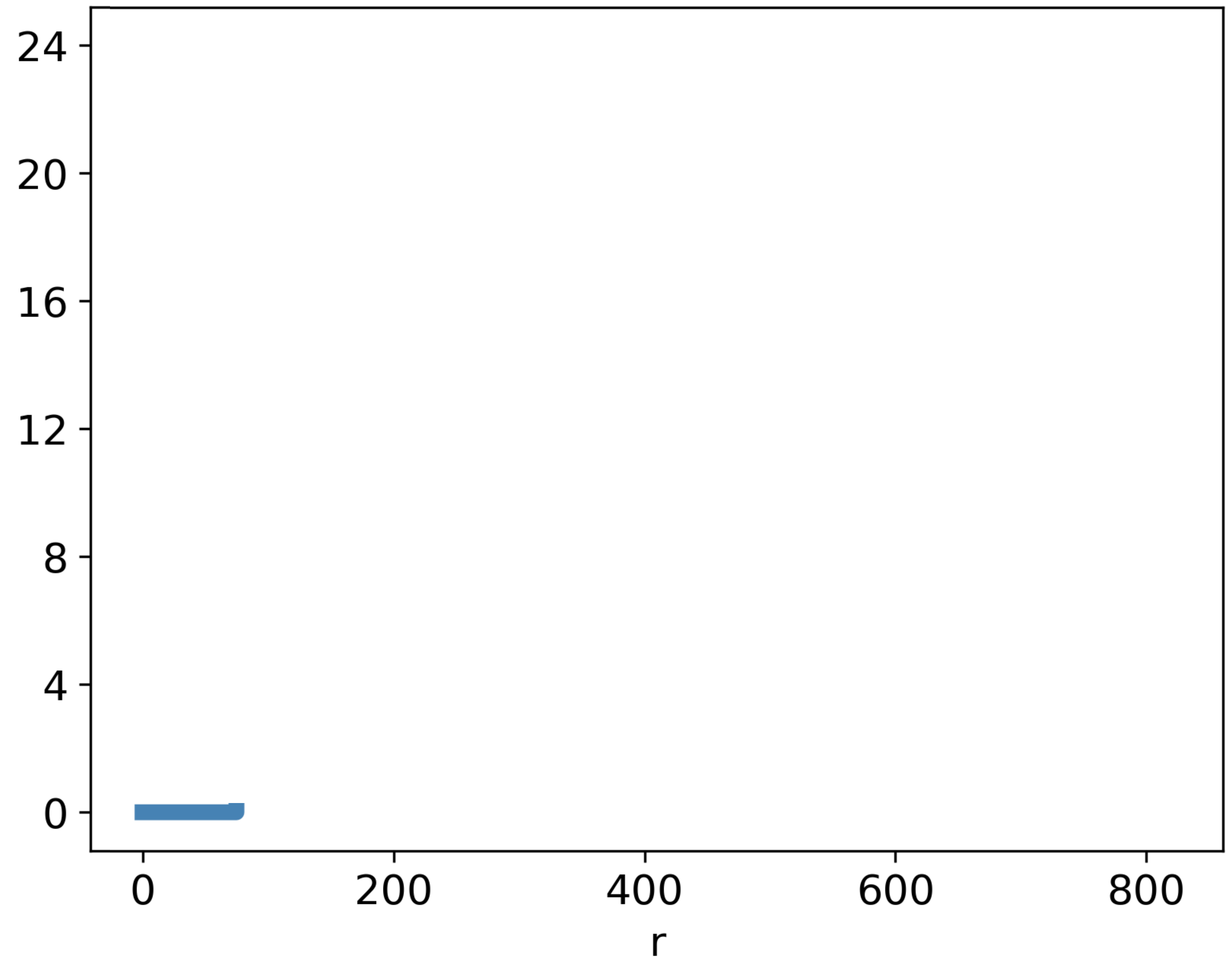
$n_2 = 6$



Ripley's K function

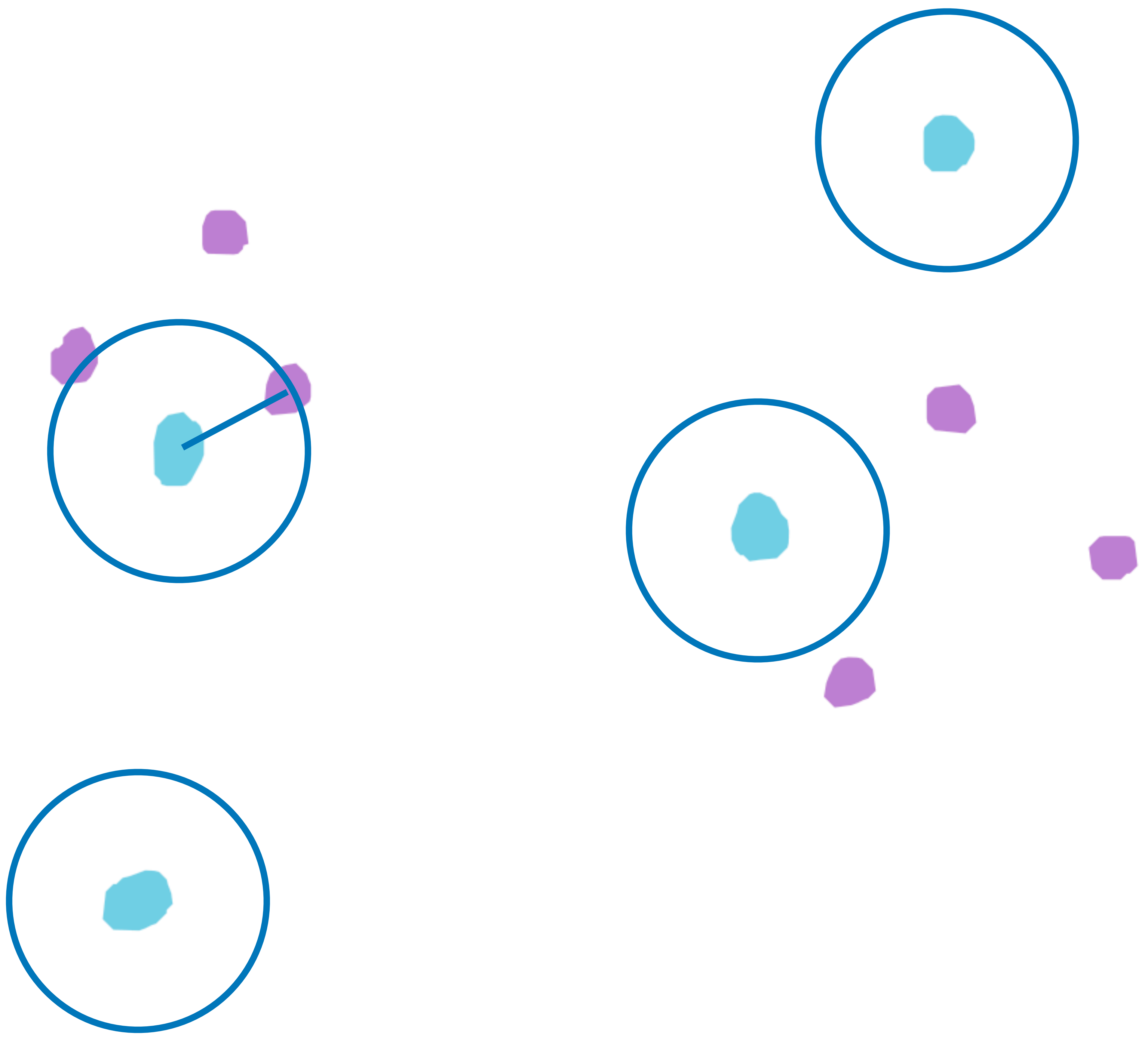
$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) b(i, j, r)$$

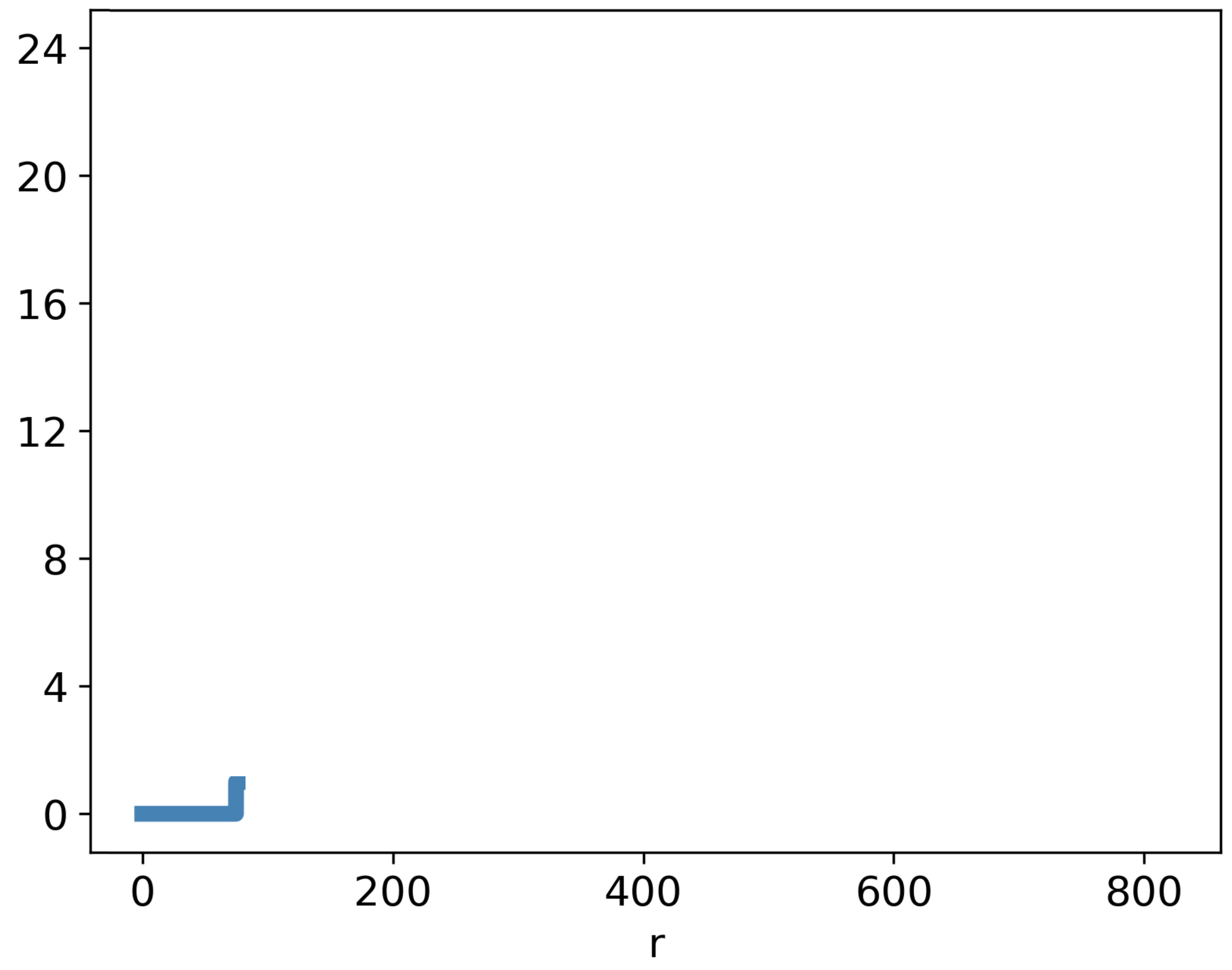




Ripley's K function

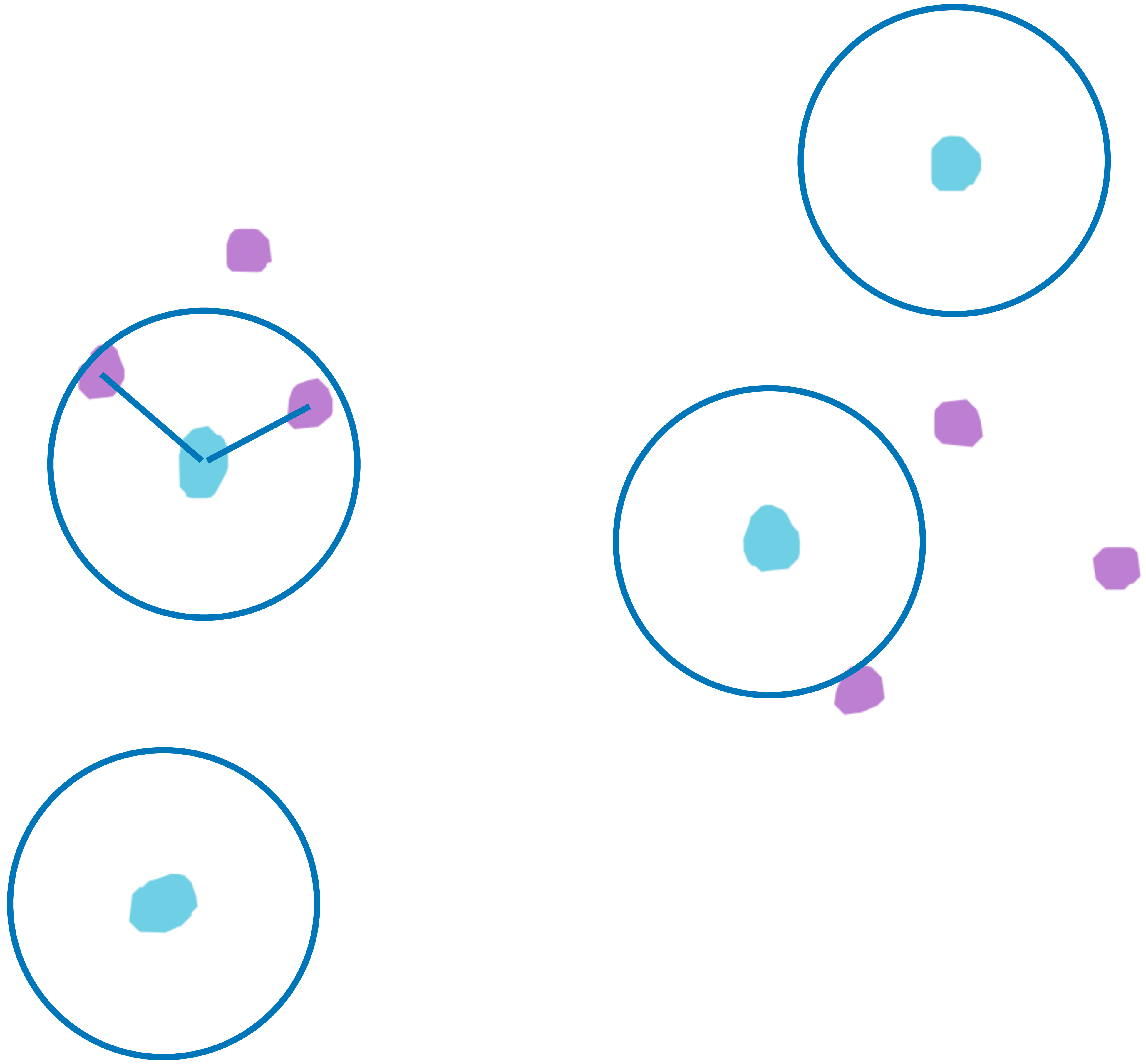
$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r)\, b(i, j, r)$$

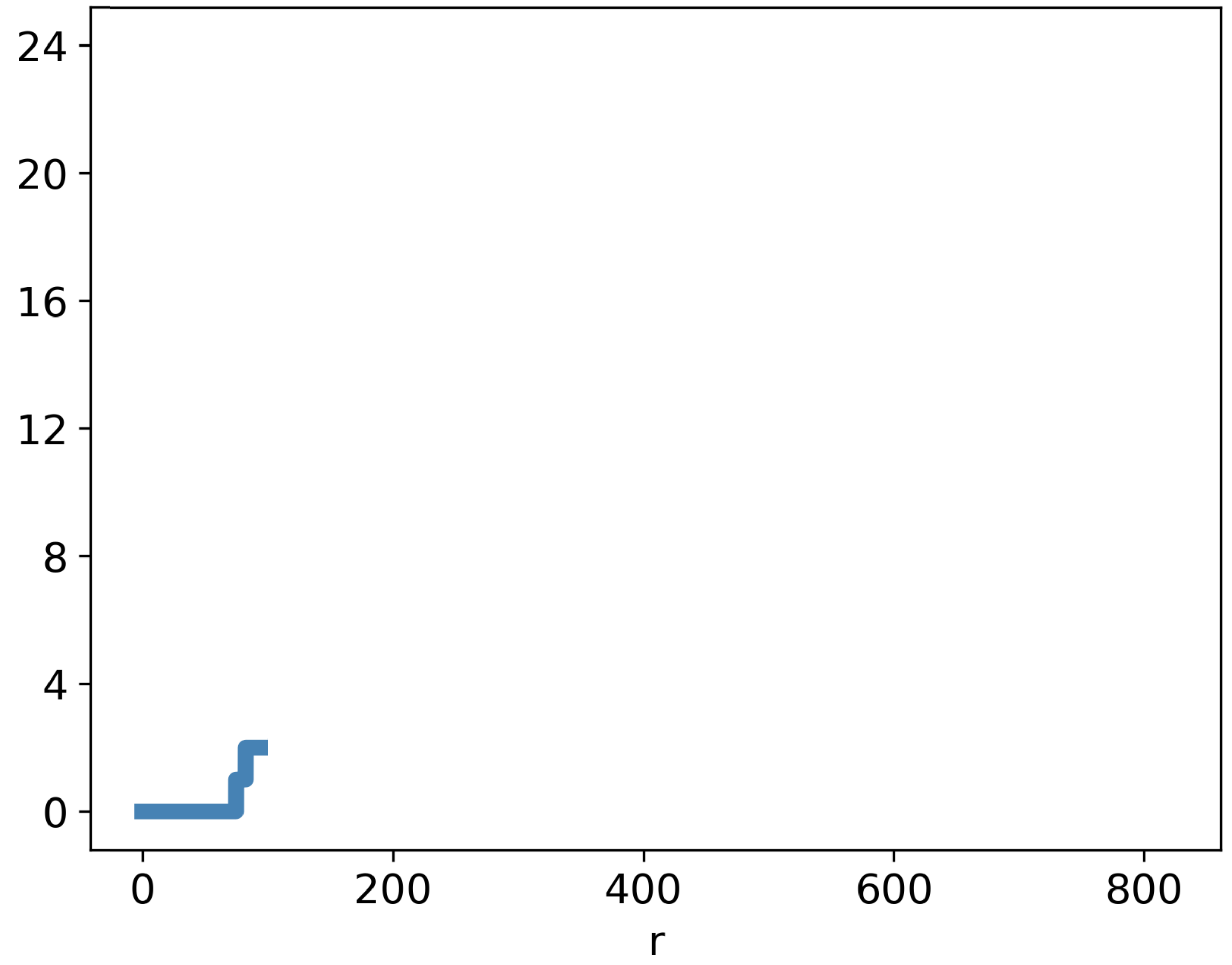




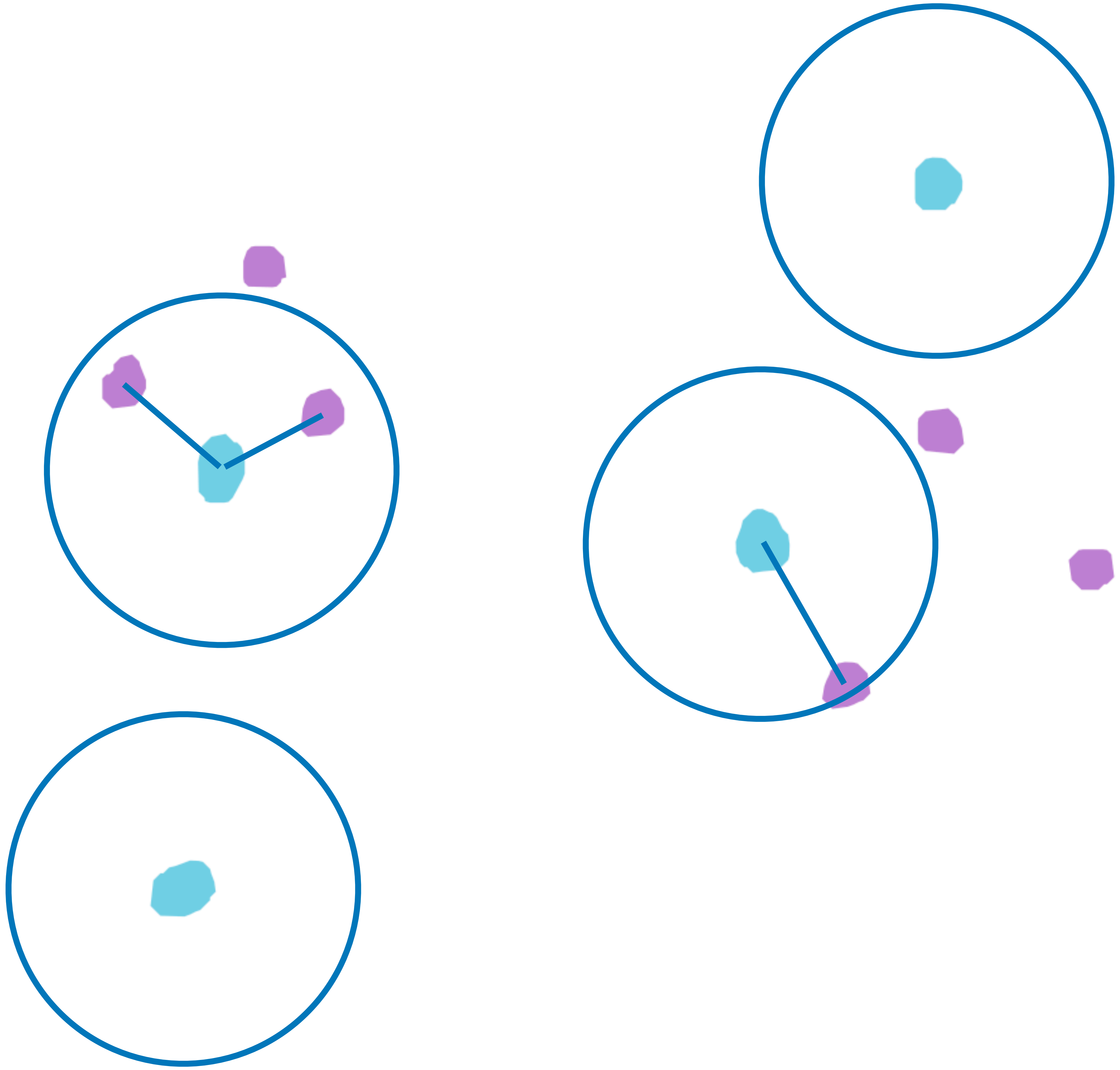
Ripley's K function

$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r)\, b(i, j, r)$$

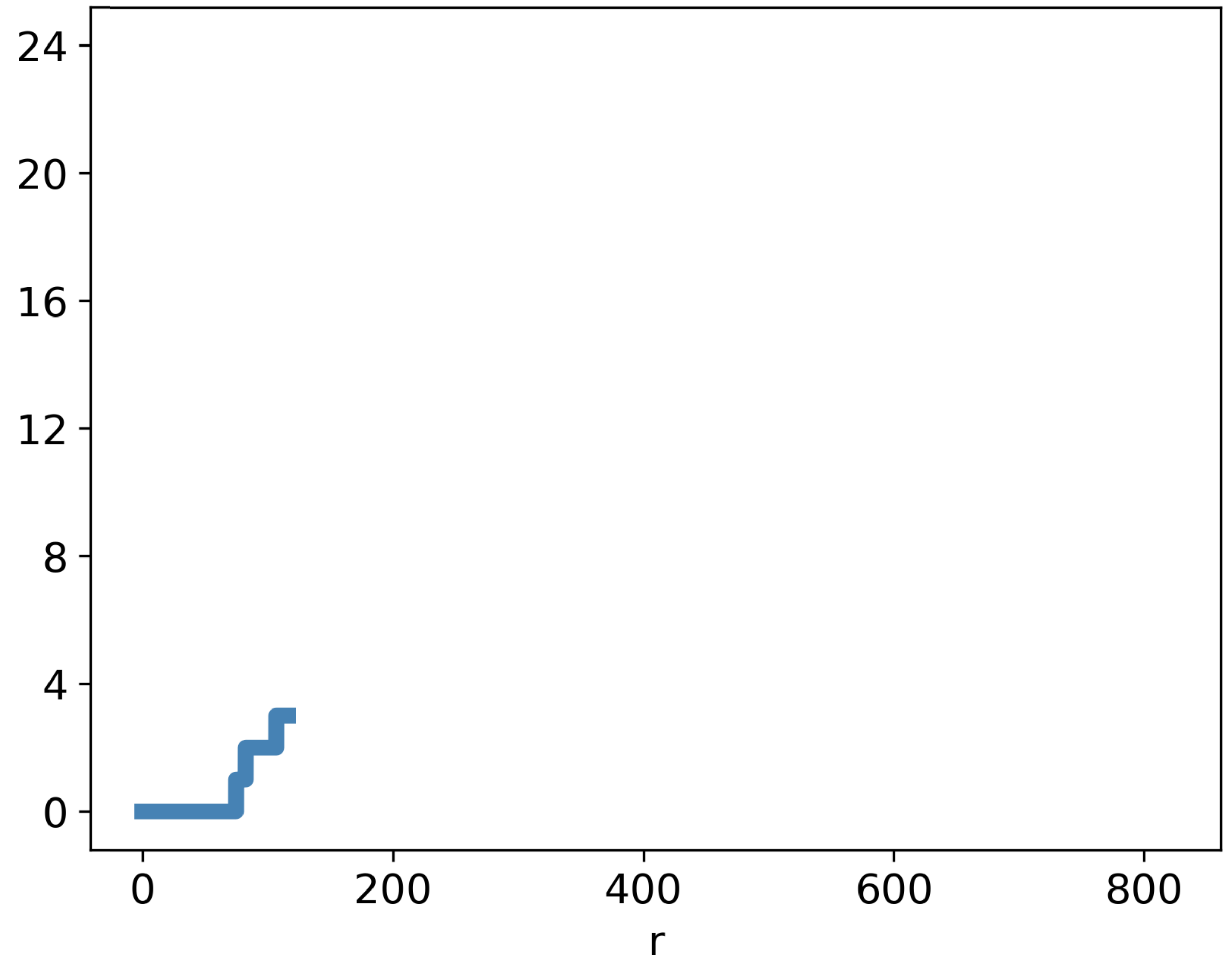




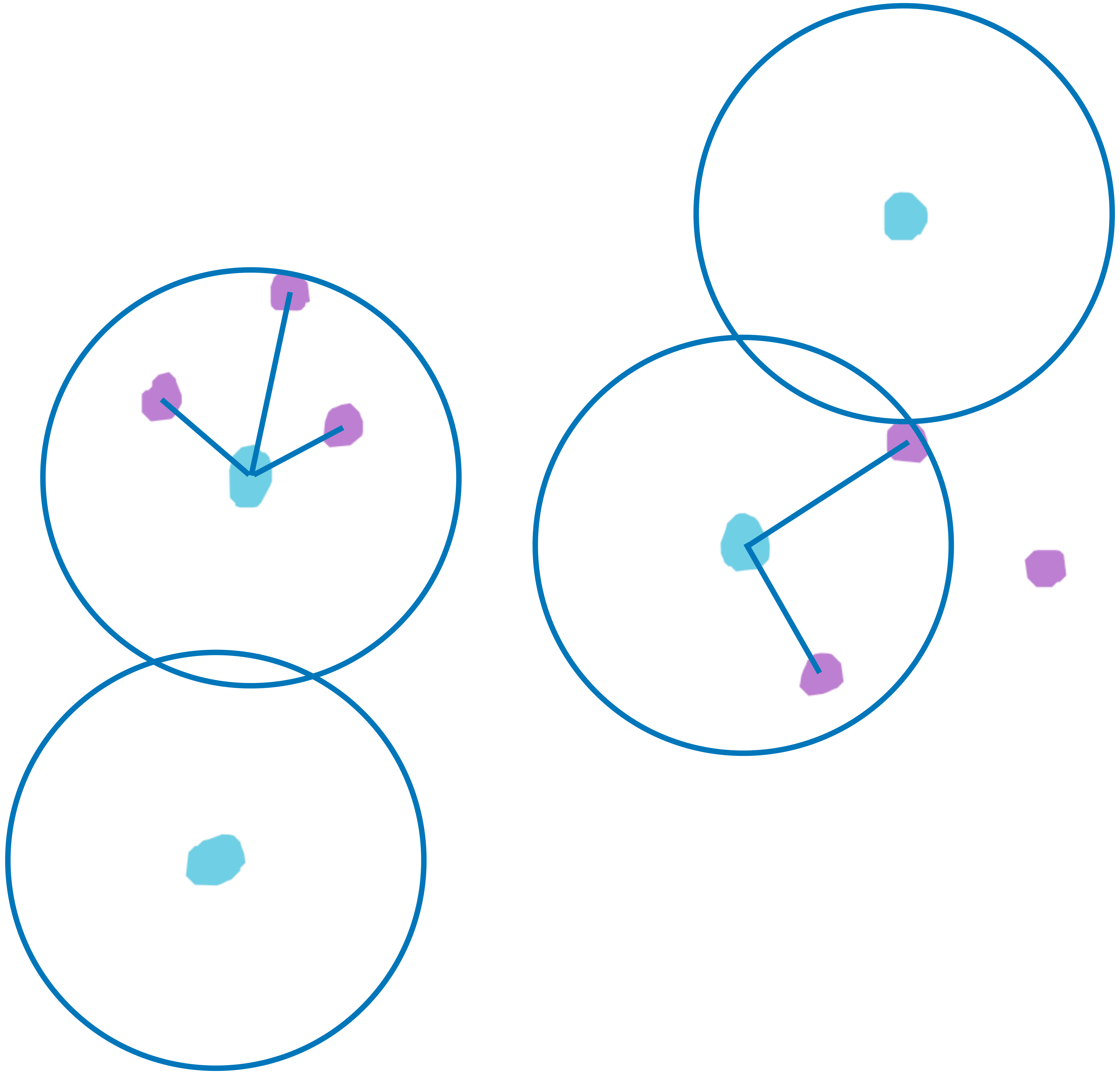
Ripley's K function



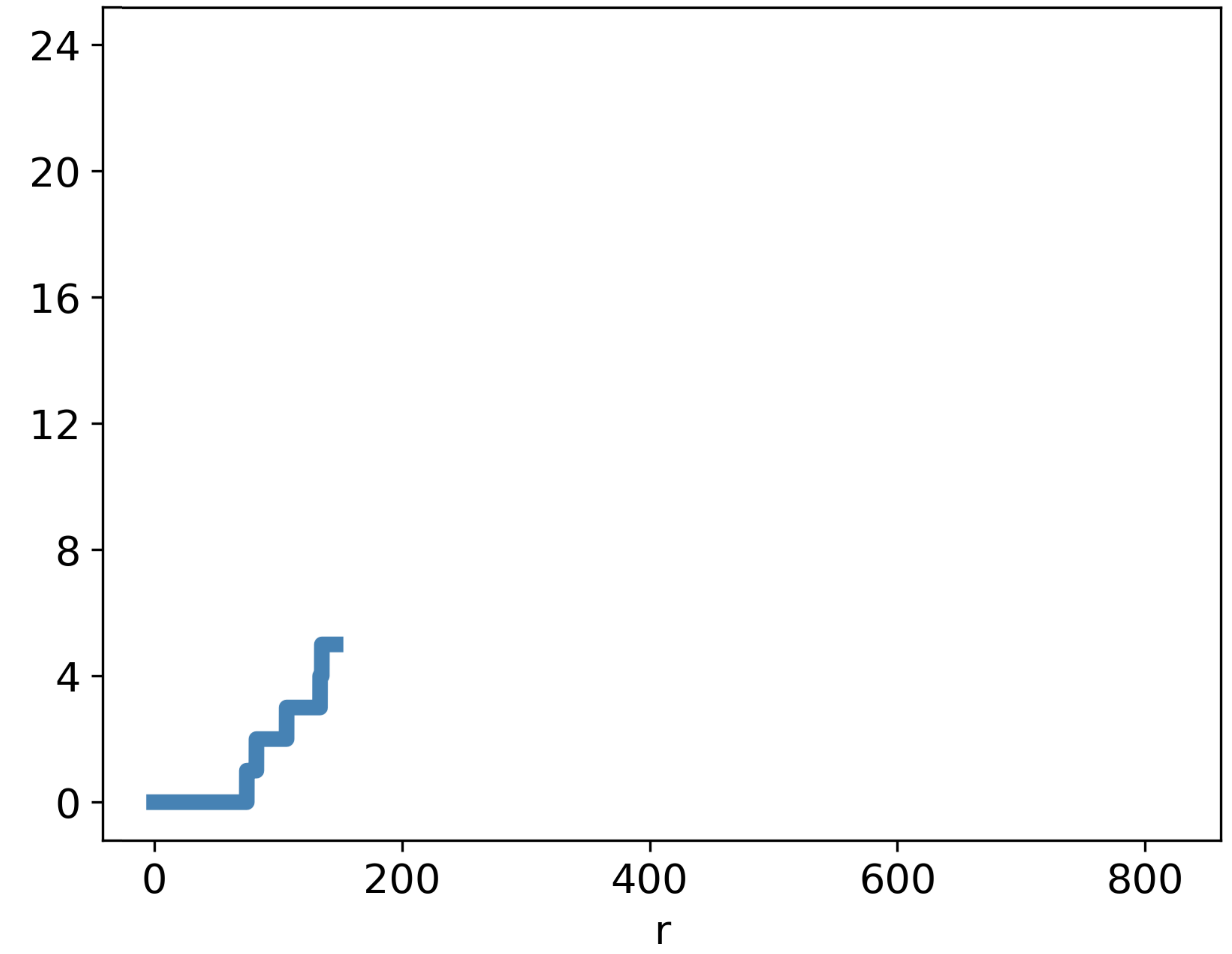
$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) b(i, j, r)$$



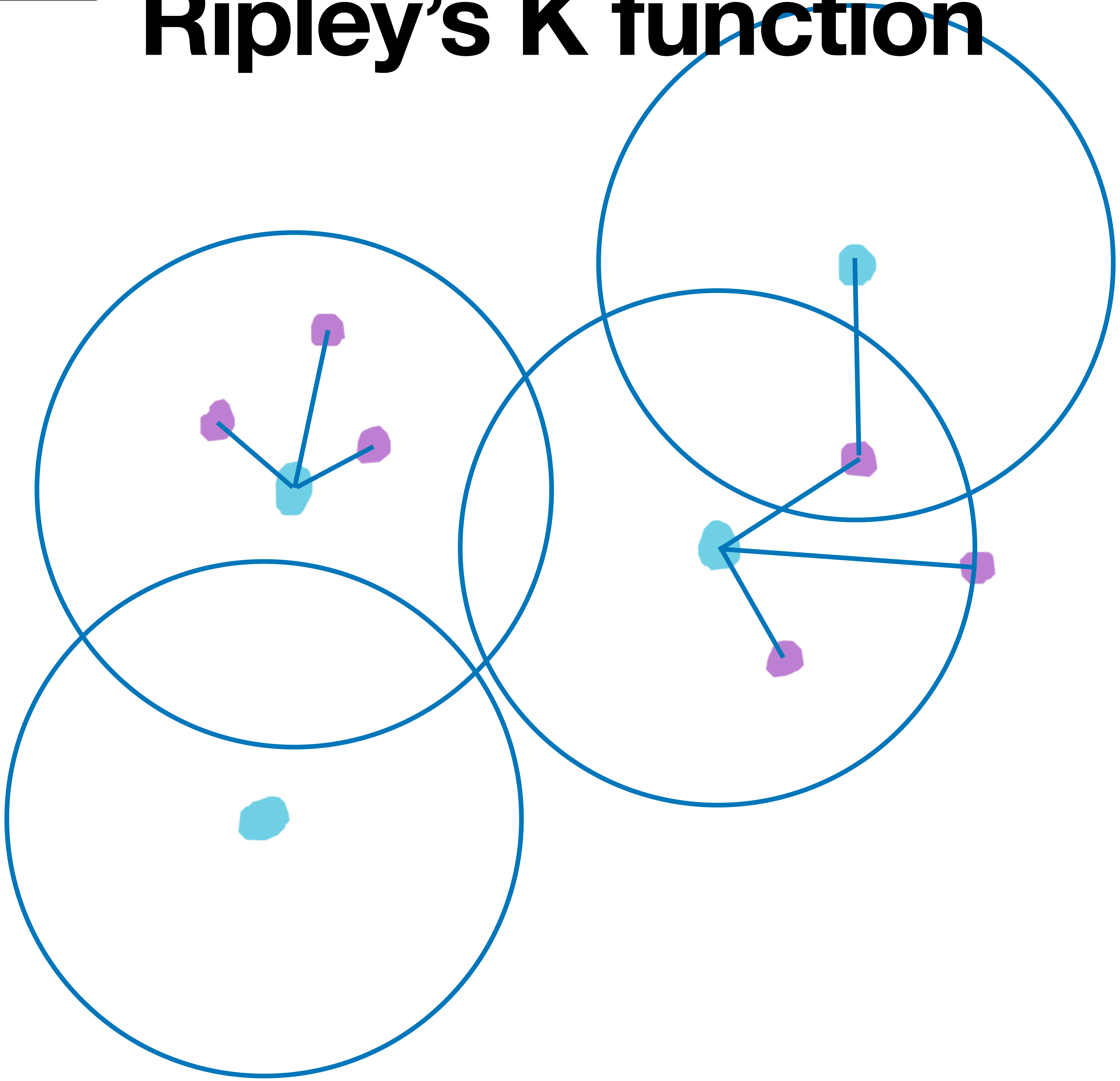
Ripley's K function



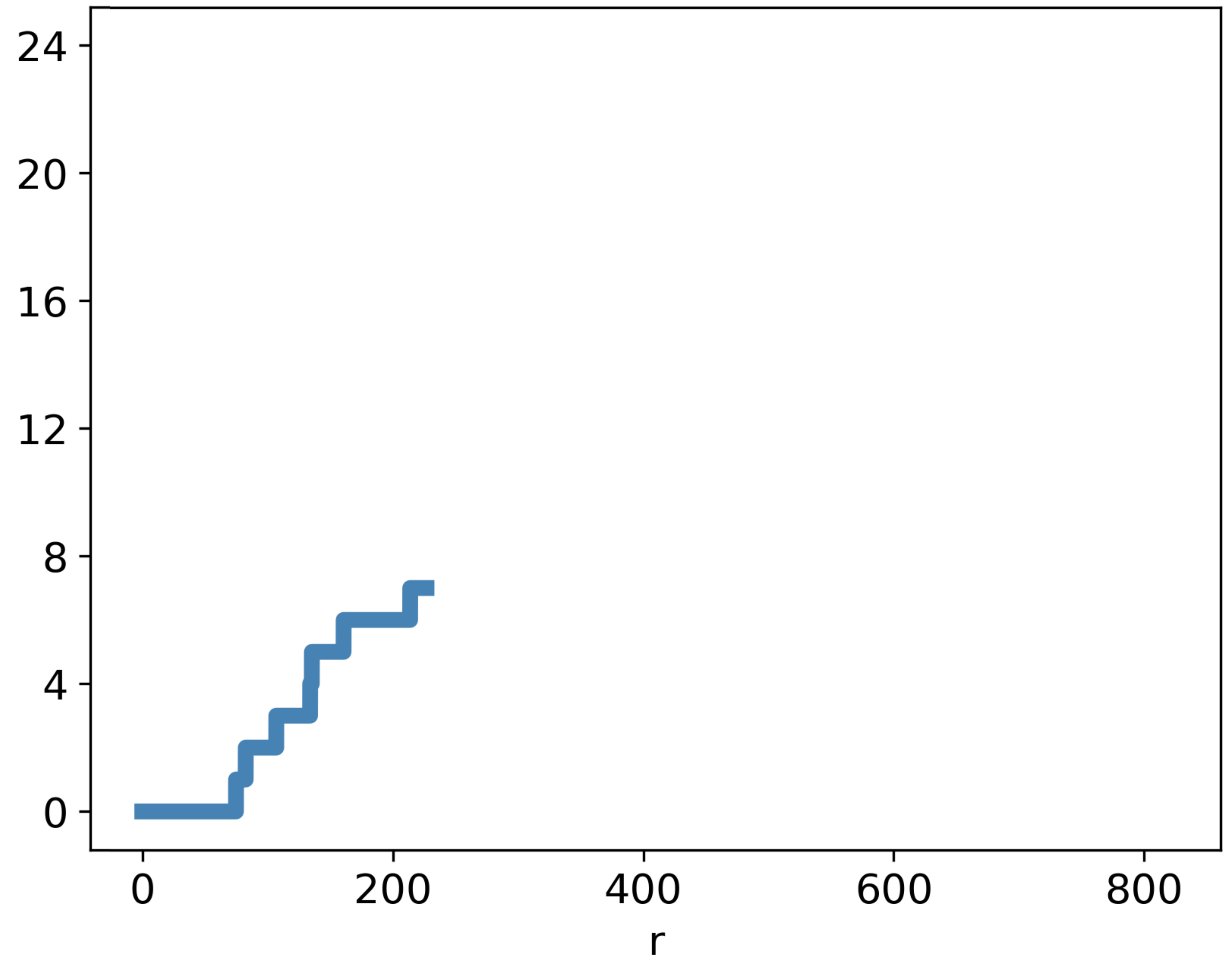
$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) b(i, j, r)$$



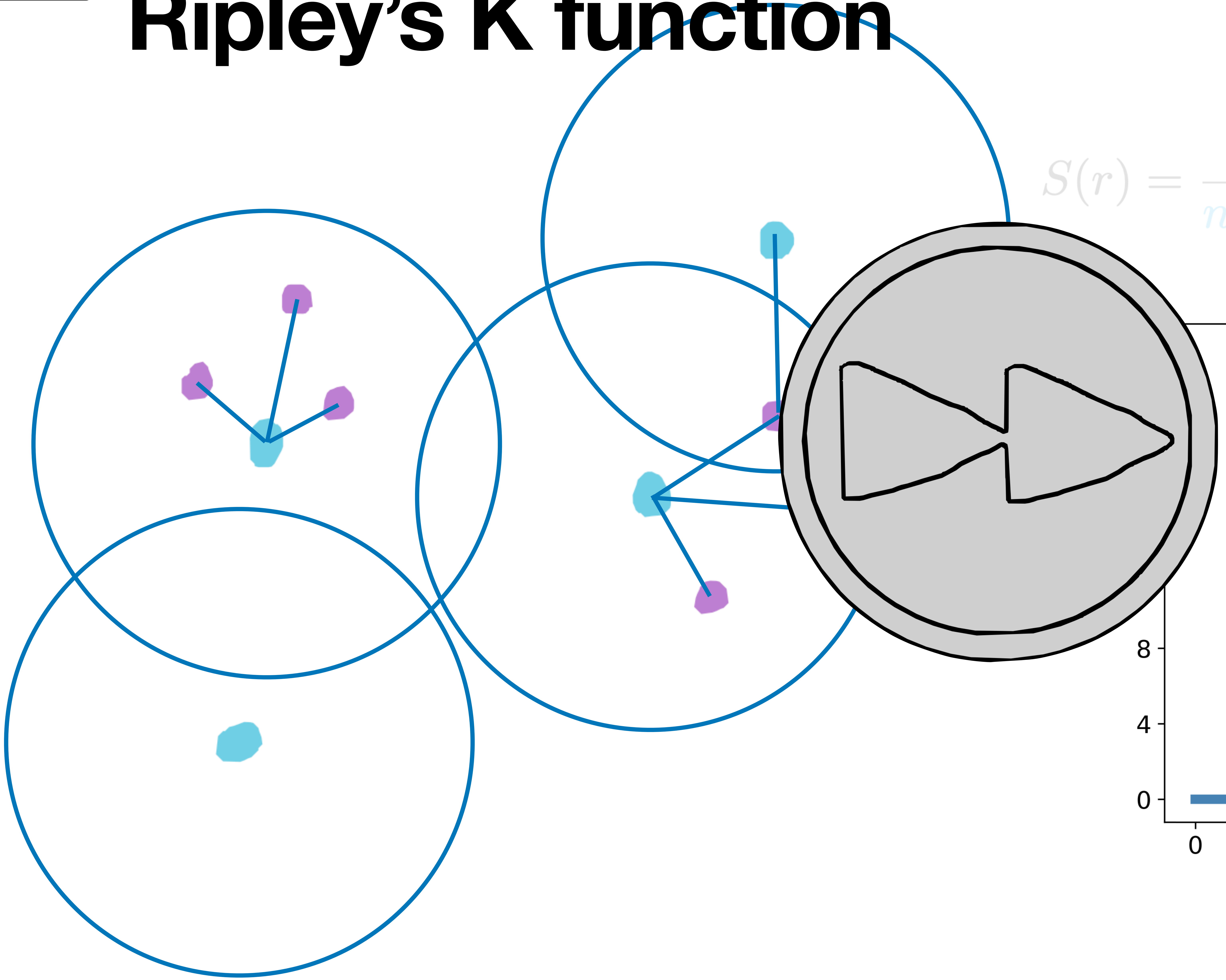
Ripley's K function



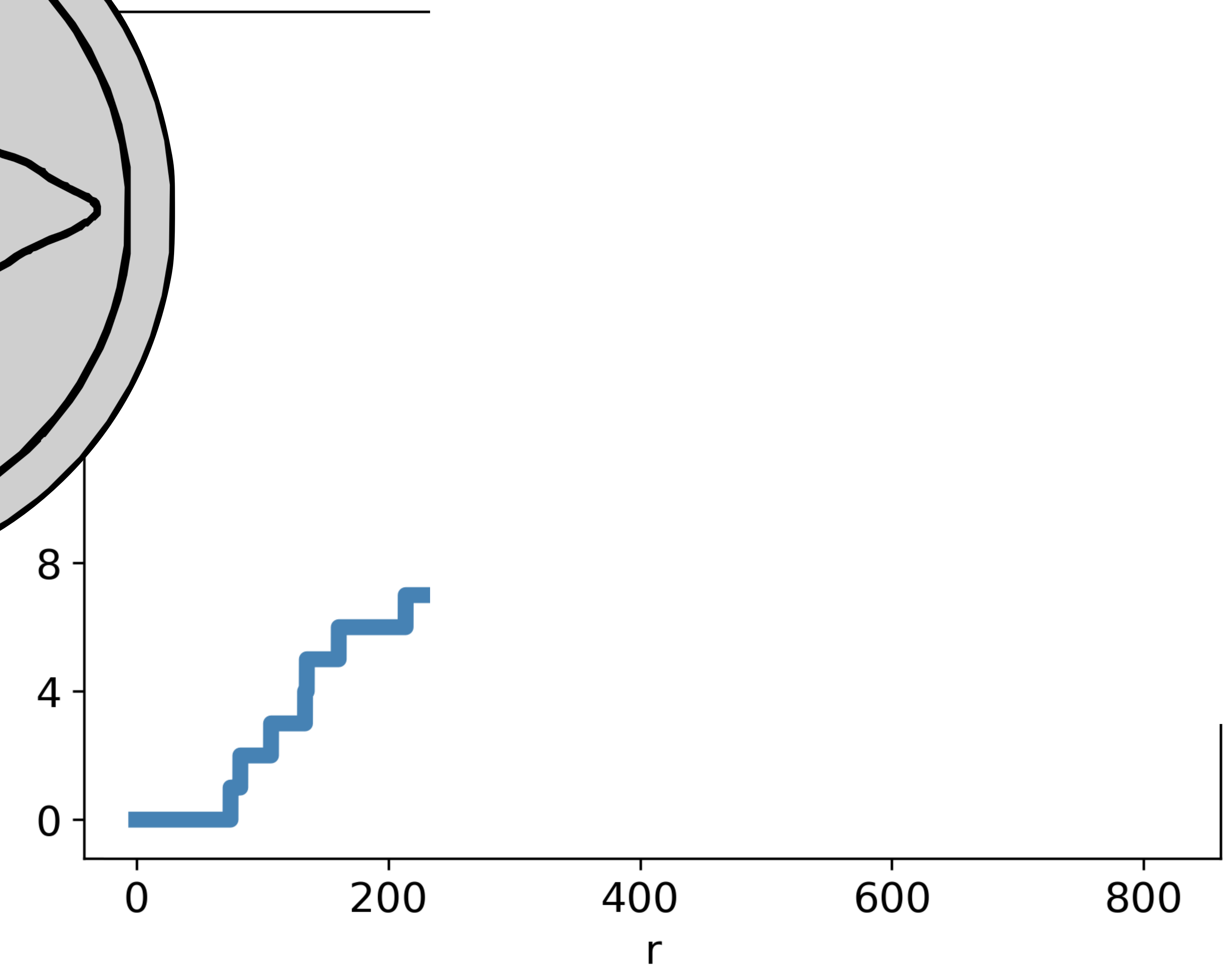
$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) b(i, j, r)$$



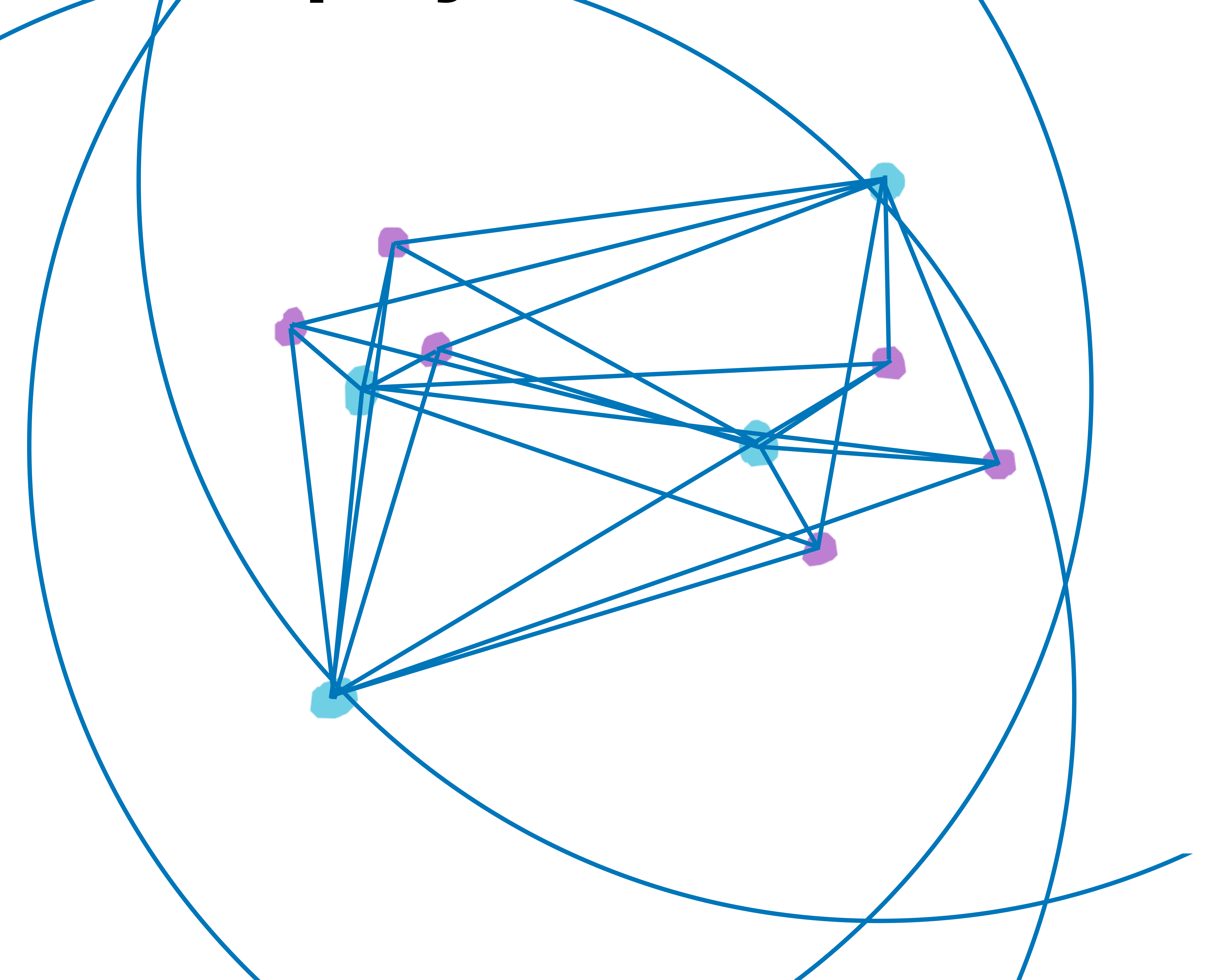
Ripley's K function

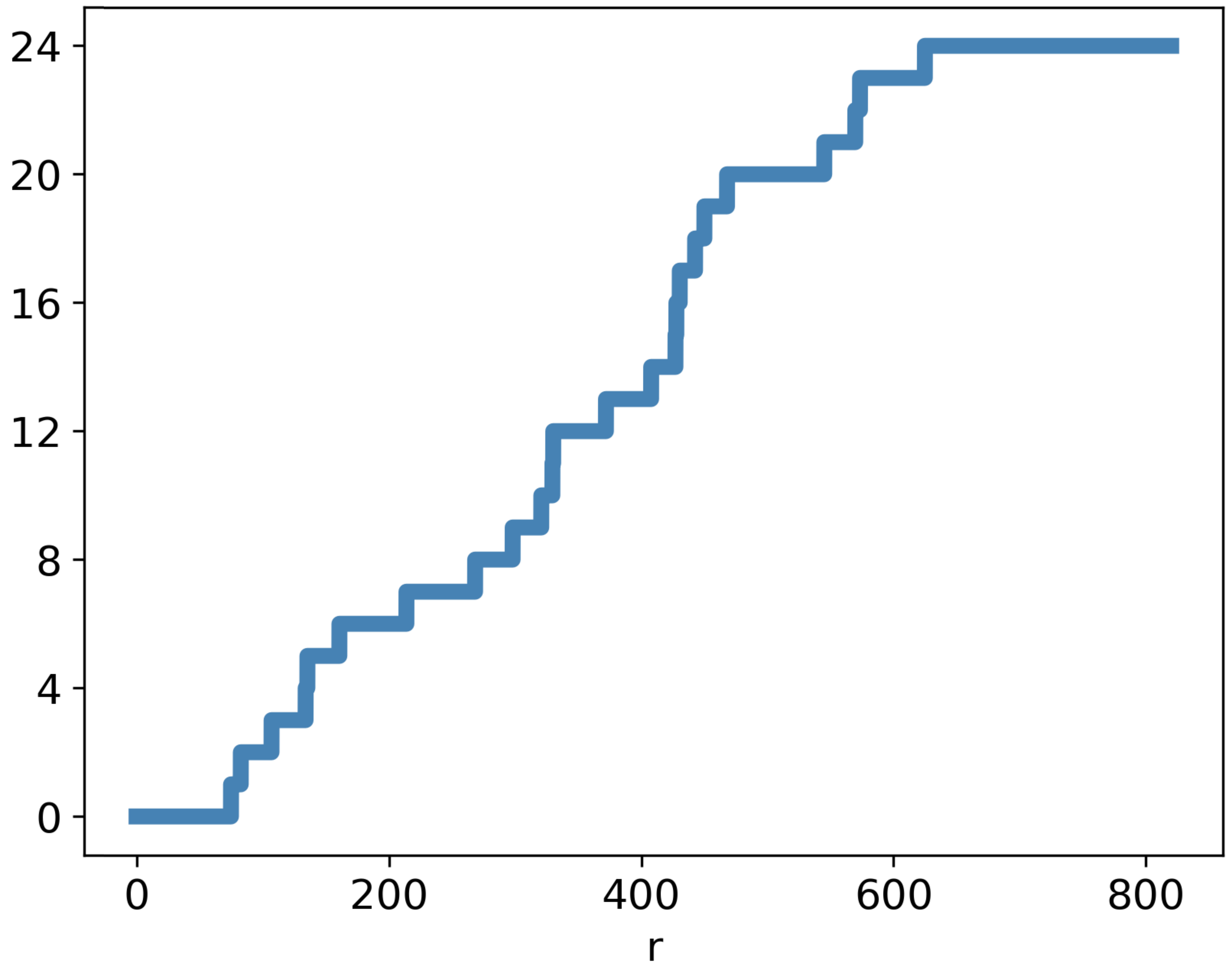


$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) b(i, j, r)$$



Ripley’s K function

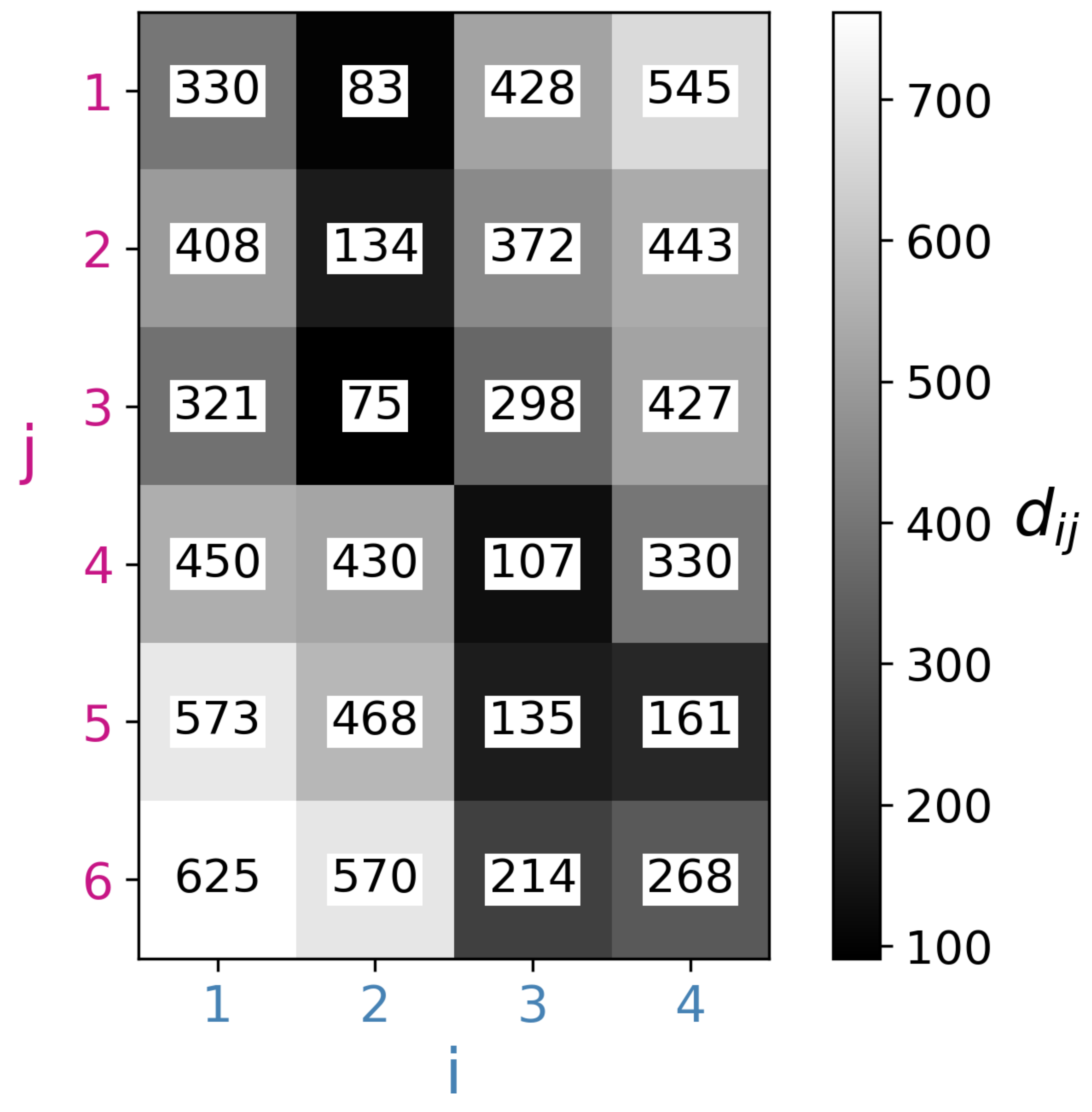


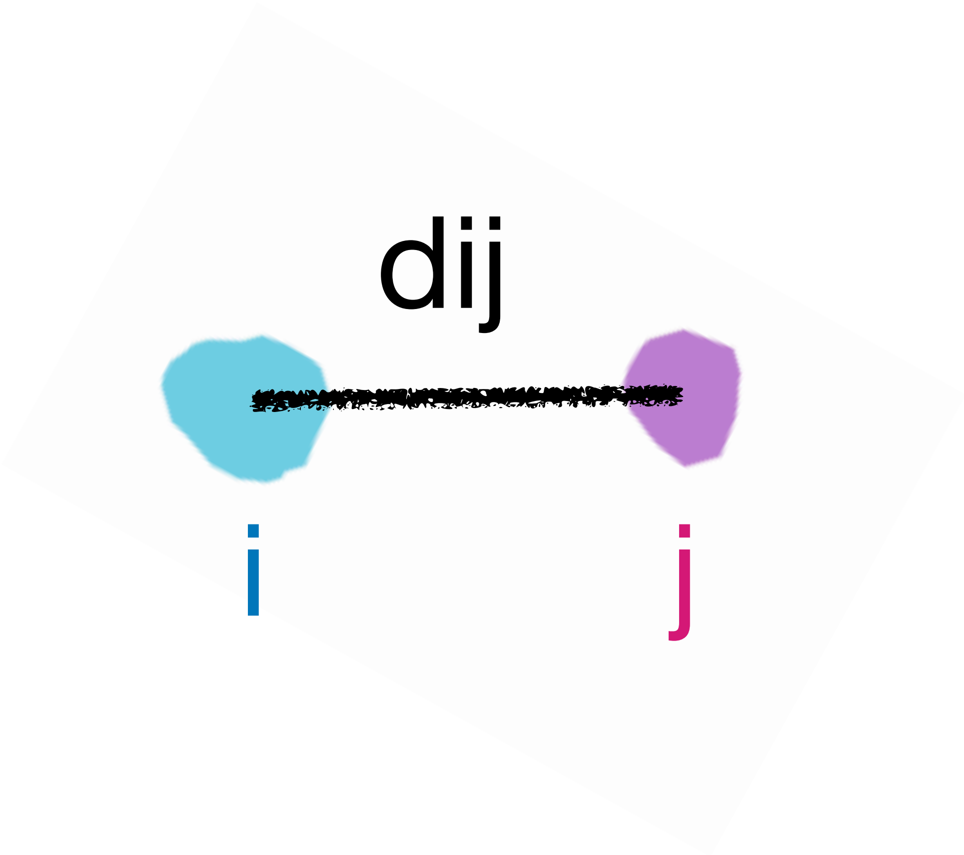


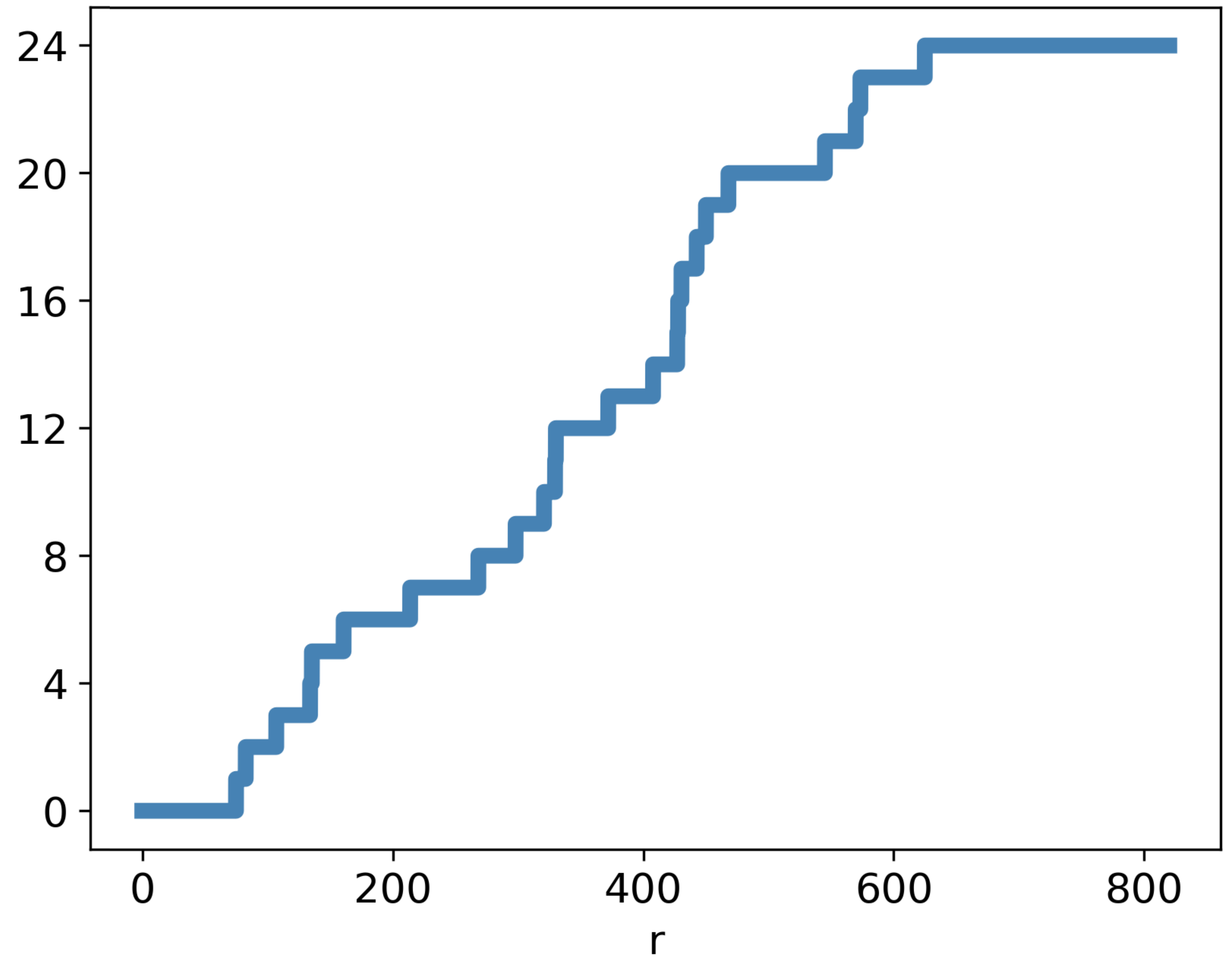
Ripley's K function

Ripley’s K function

$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) b(i, j, r)$$



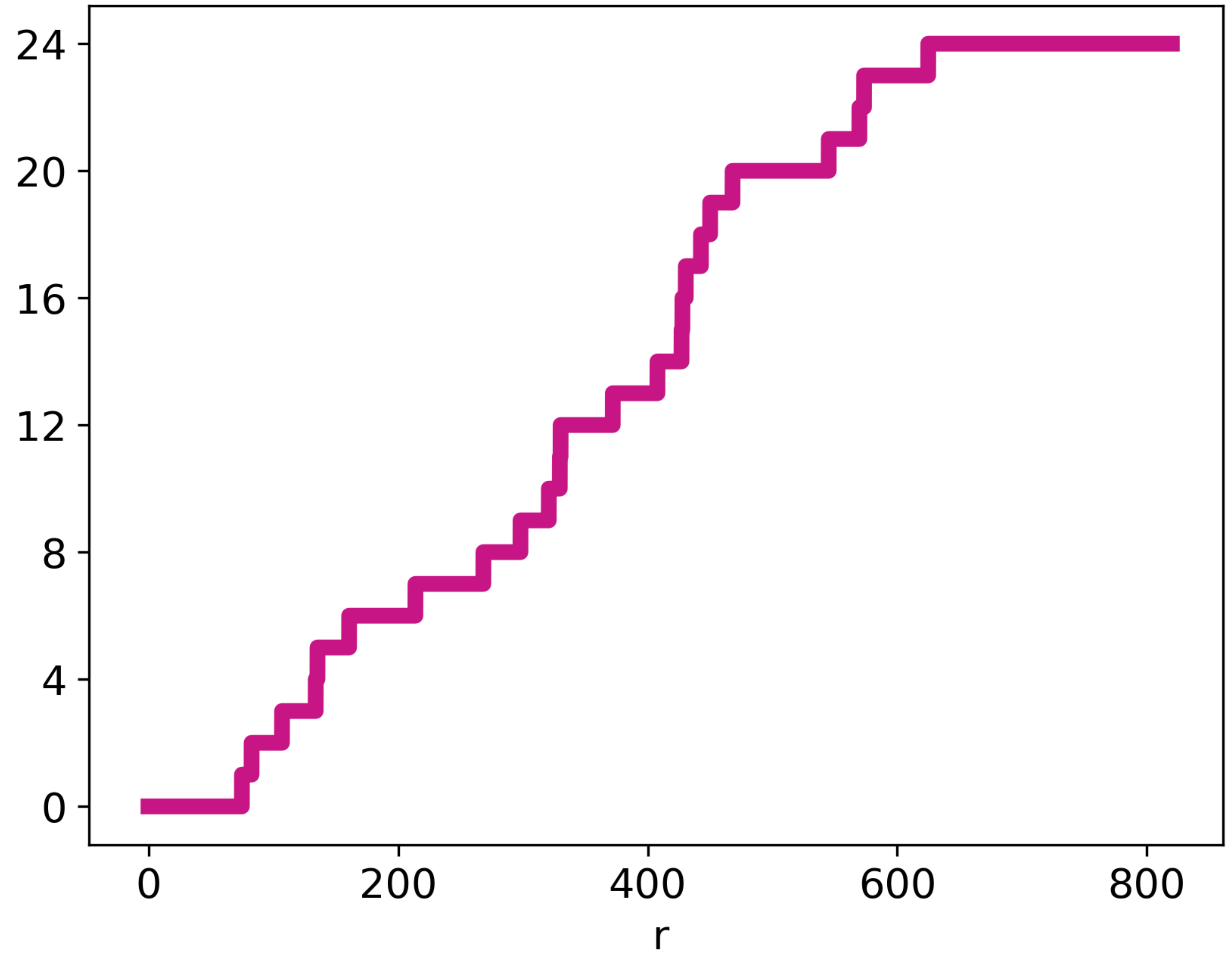




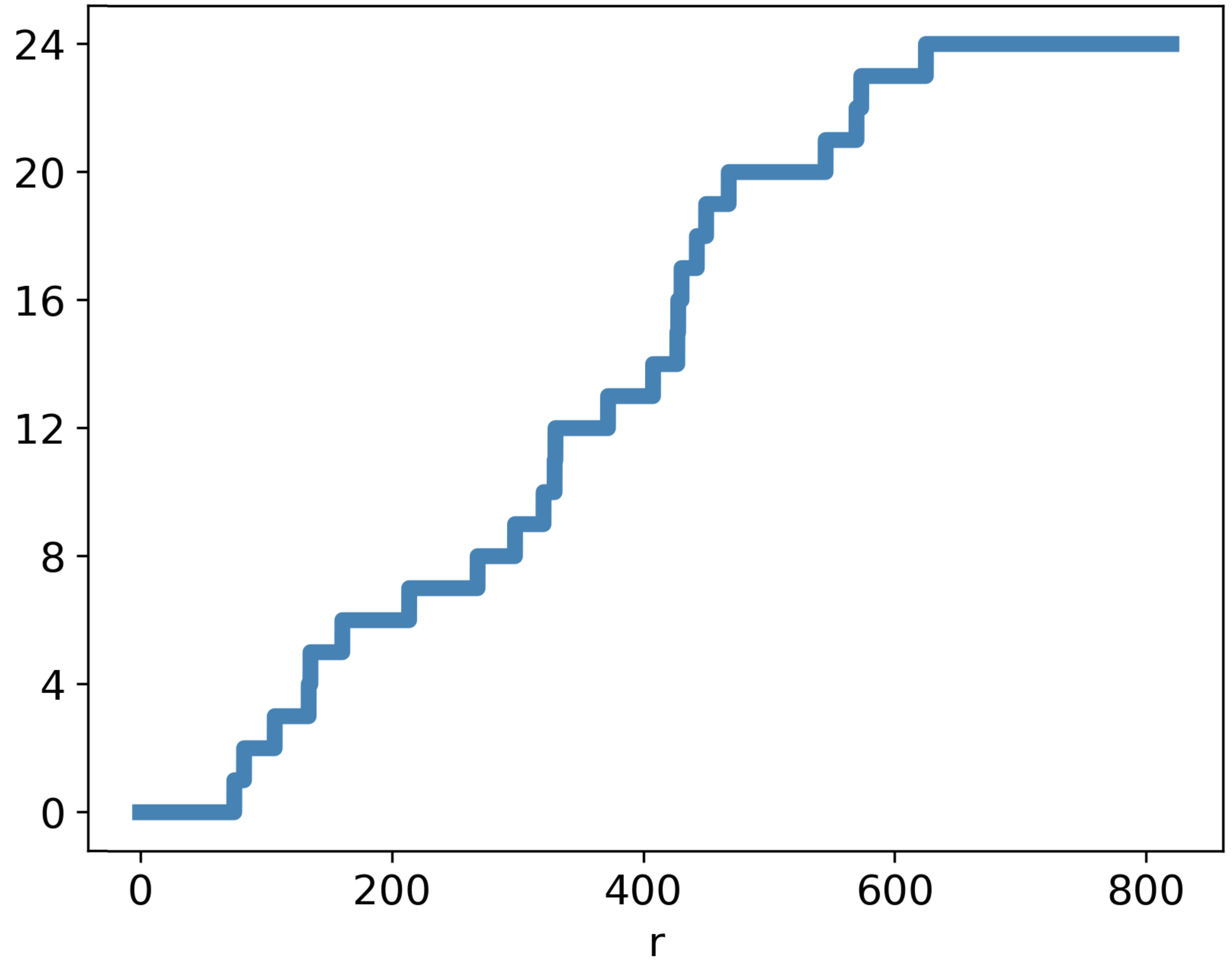
Ripley’s K function

$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) b(i, j, r)$$

$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) b(i, j, r)$$

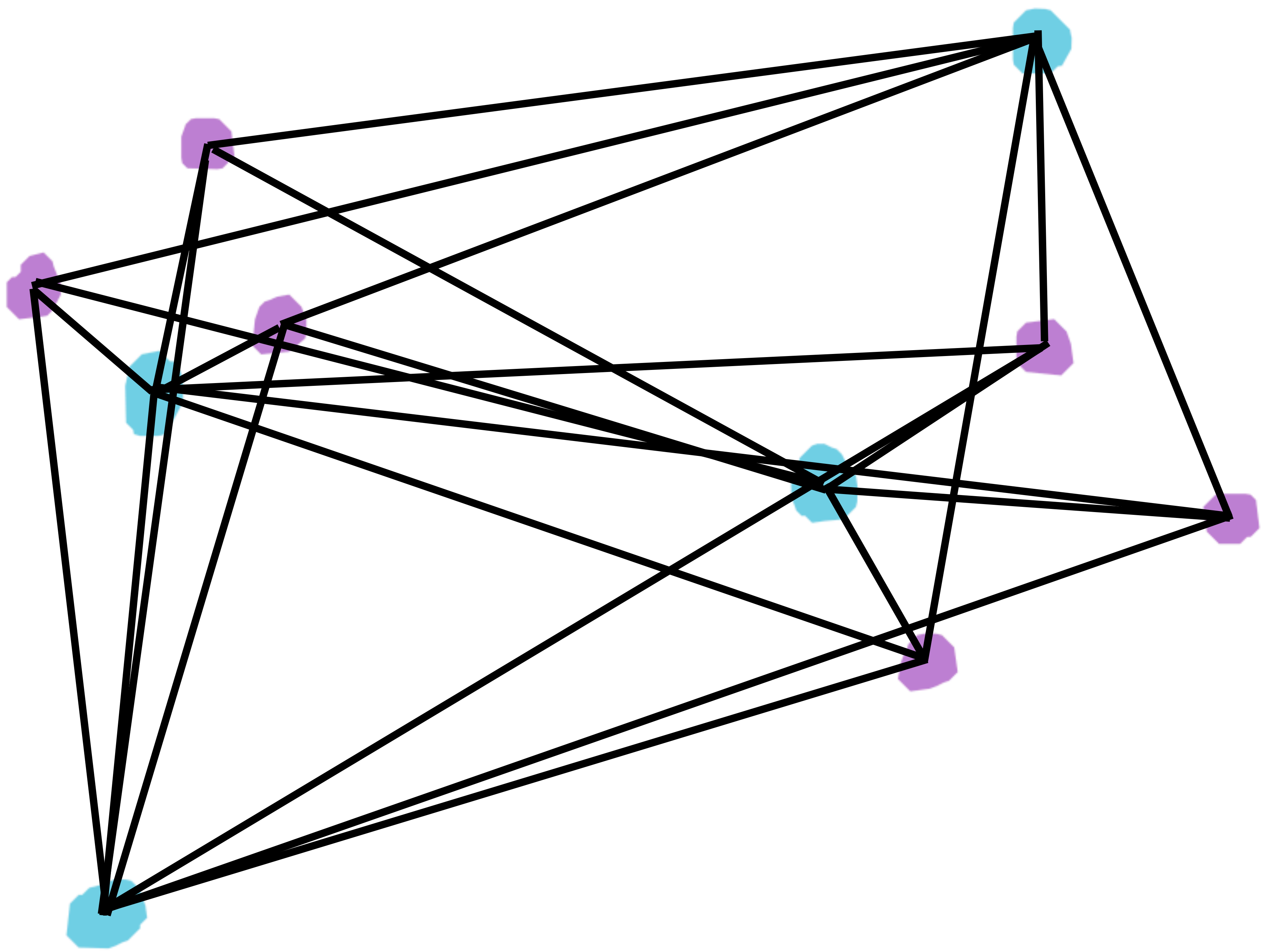


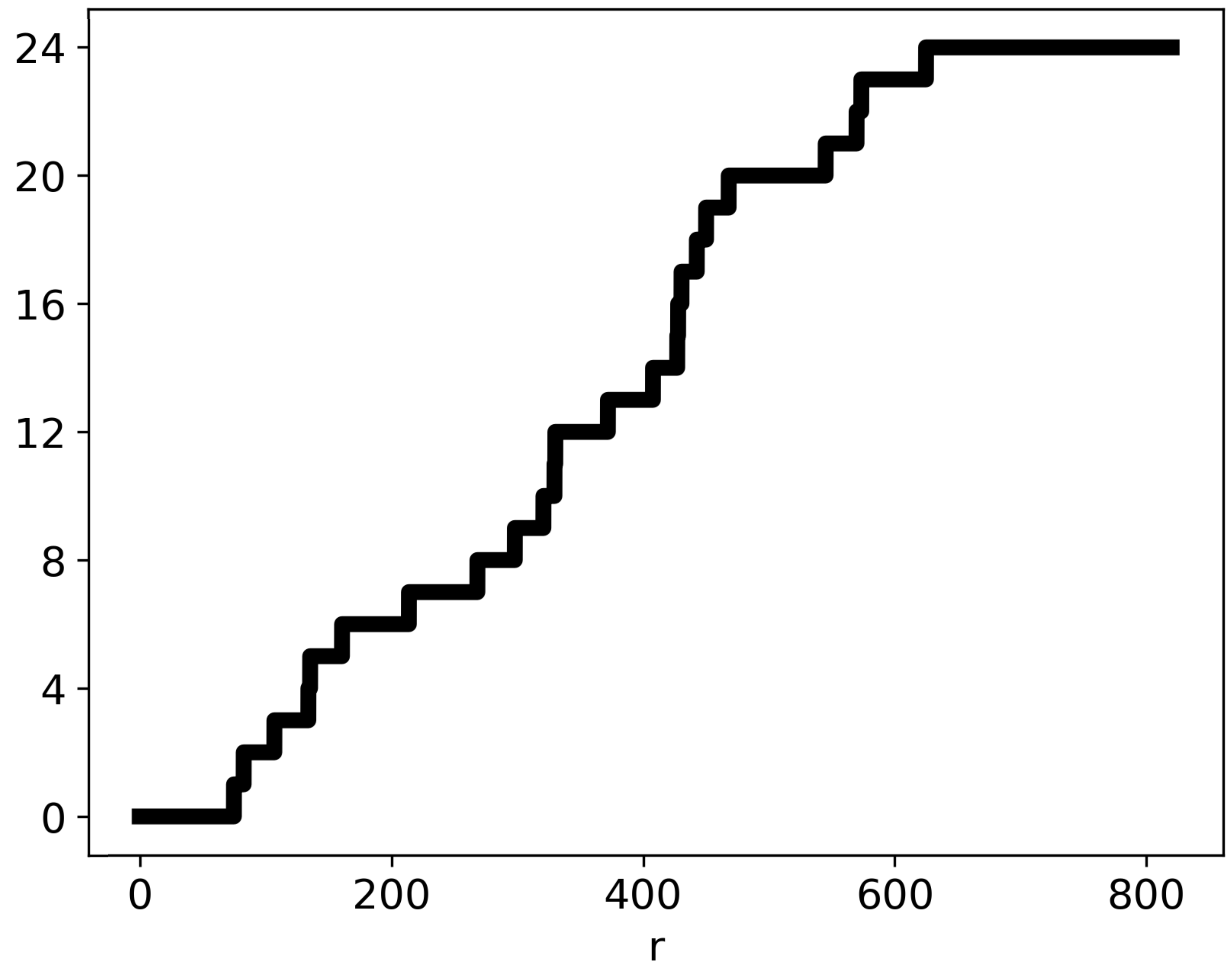
=



Ripley's K function

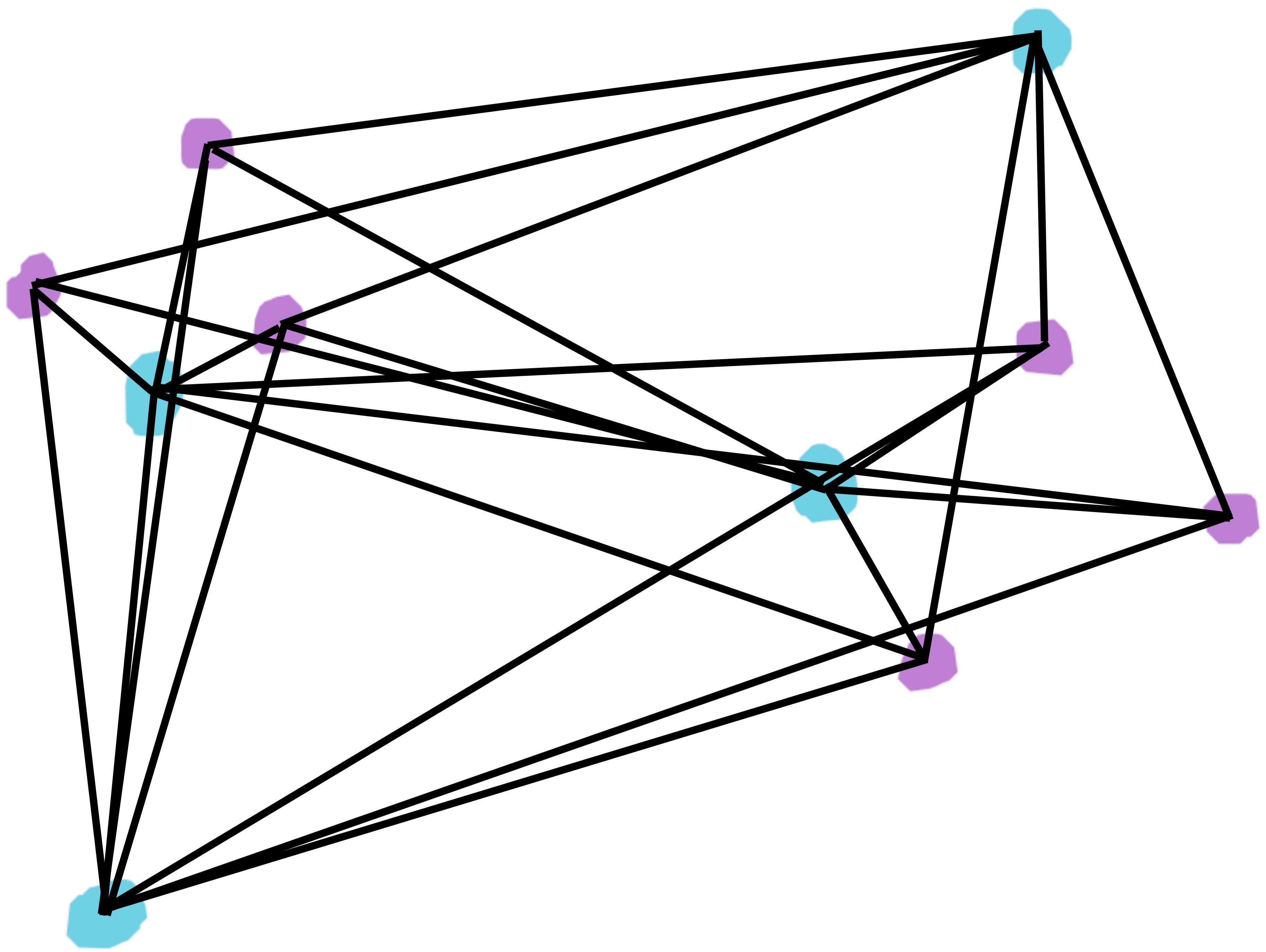
$$S(r) = \boxed{\frac{|\Omega|}{n_1 n_2}} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) b(i, j, r)$$



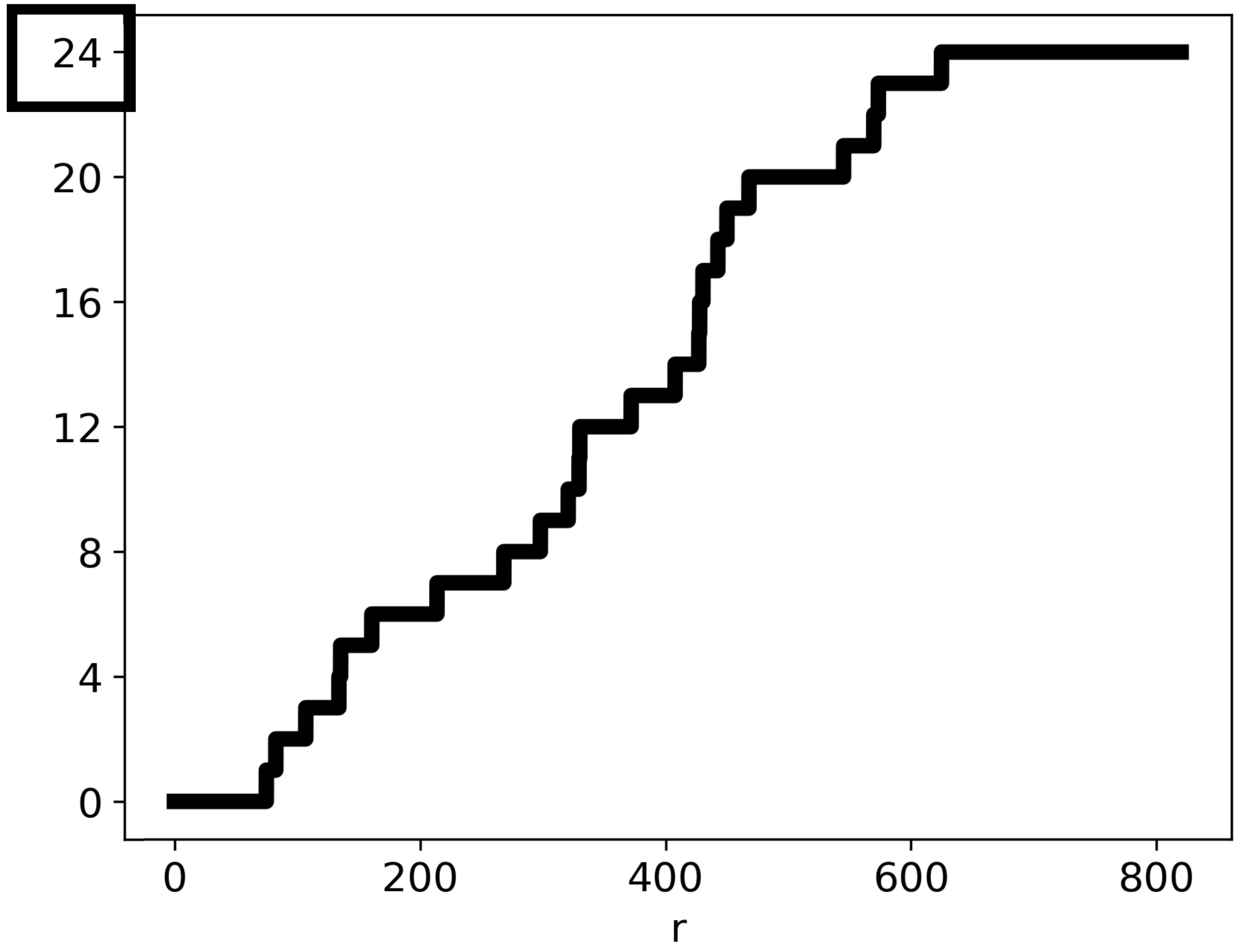


Ripley's K function

$$S(r) = \boxed{\frac{|\Omega|}{n_1 n_2}} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) b(i,j,r)$$

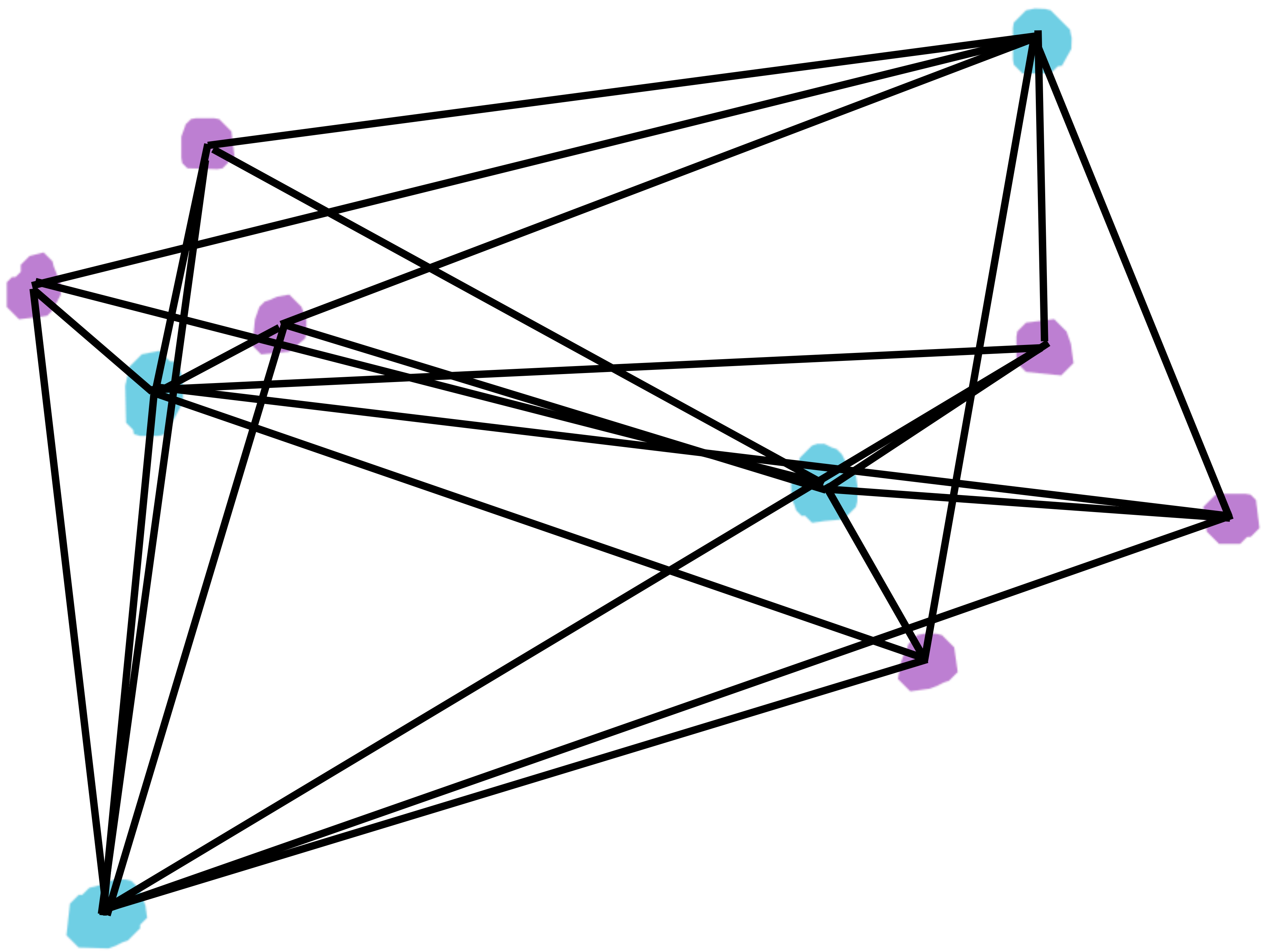


$$n_1 n_2 = \text{n connections} = 24$$

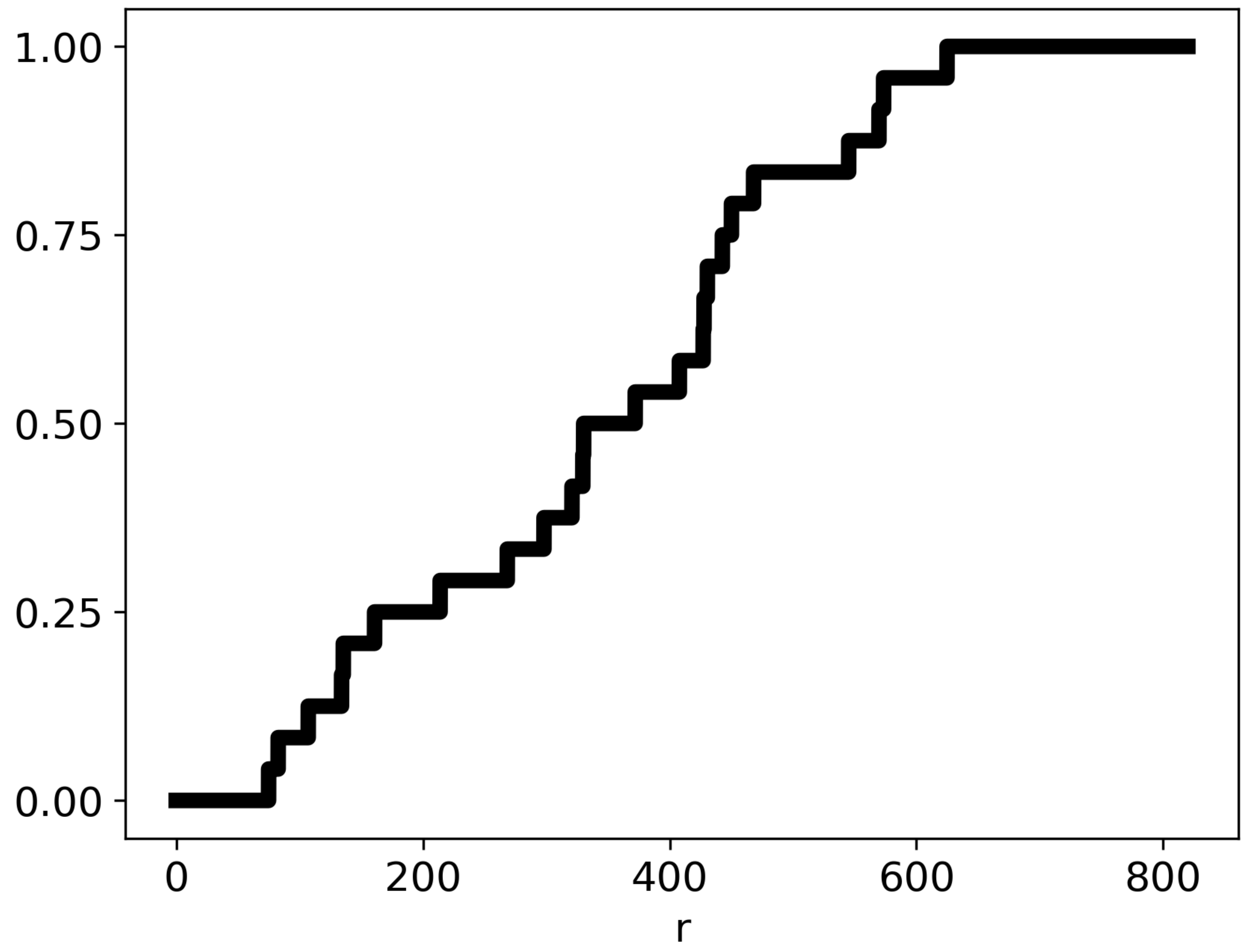


Ripley’s K function

$$S(r) = \boxed{\frac{|\Omega|}{n_1 n_2}} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) b(i, j, r)$$

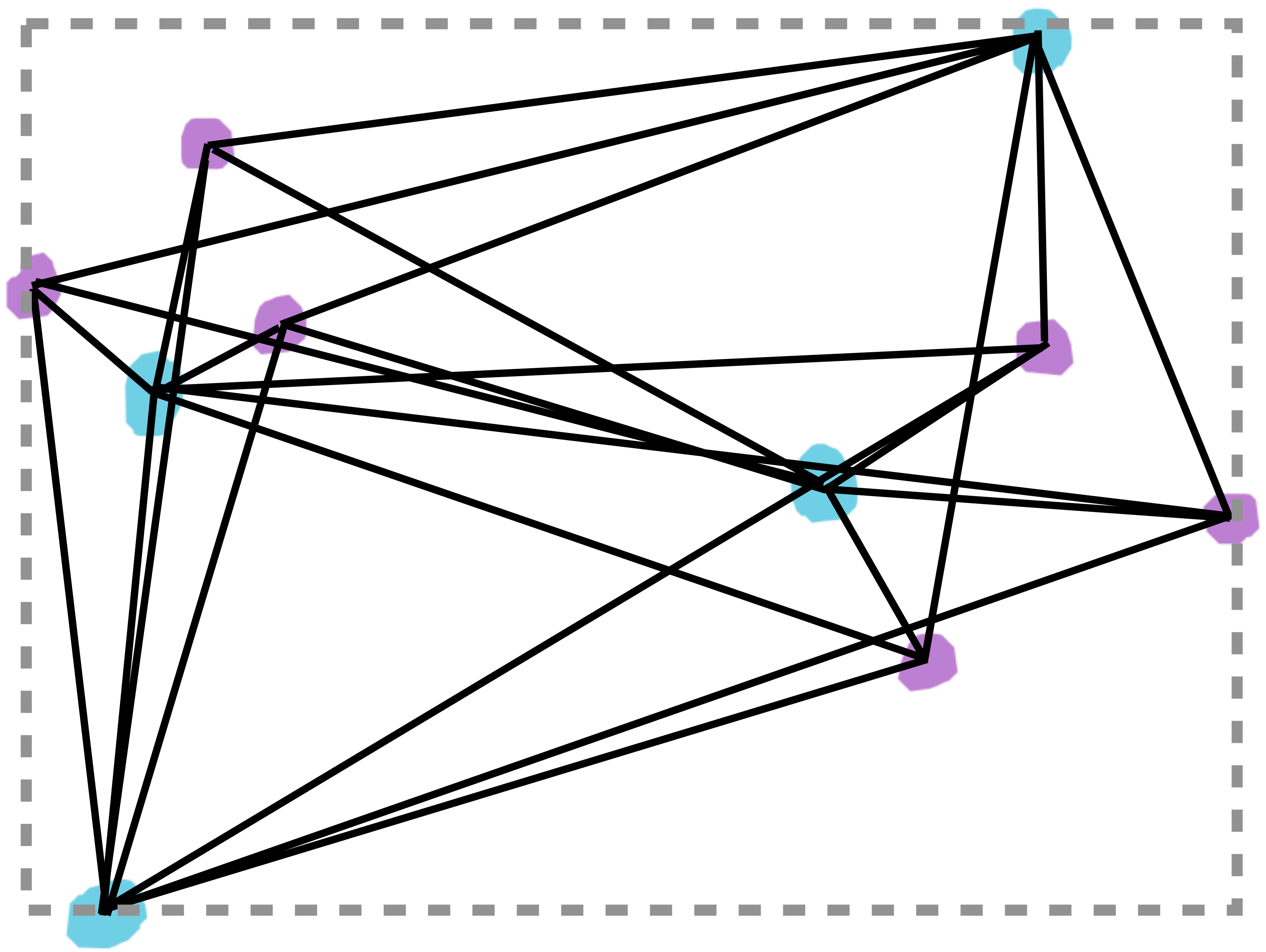


$$n_1 n_2 = \text{n connections} = 24$$

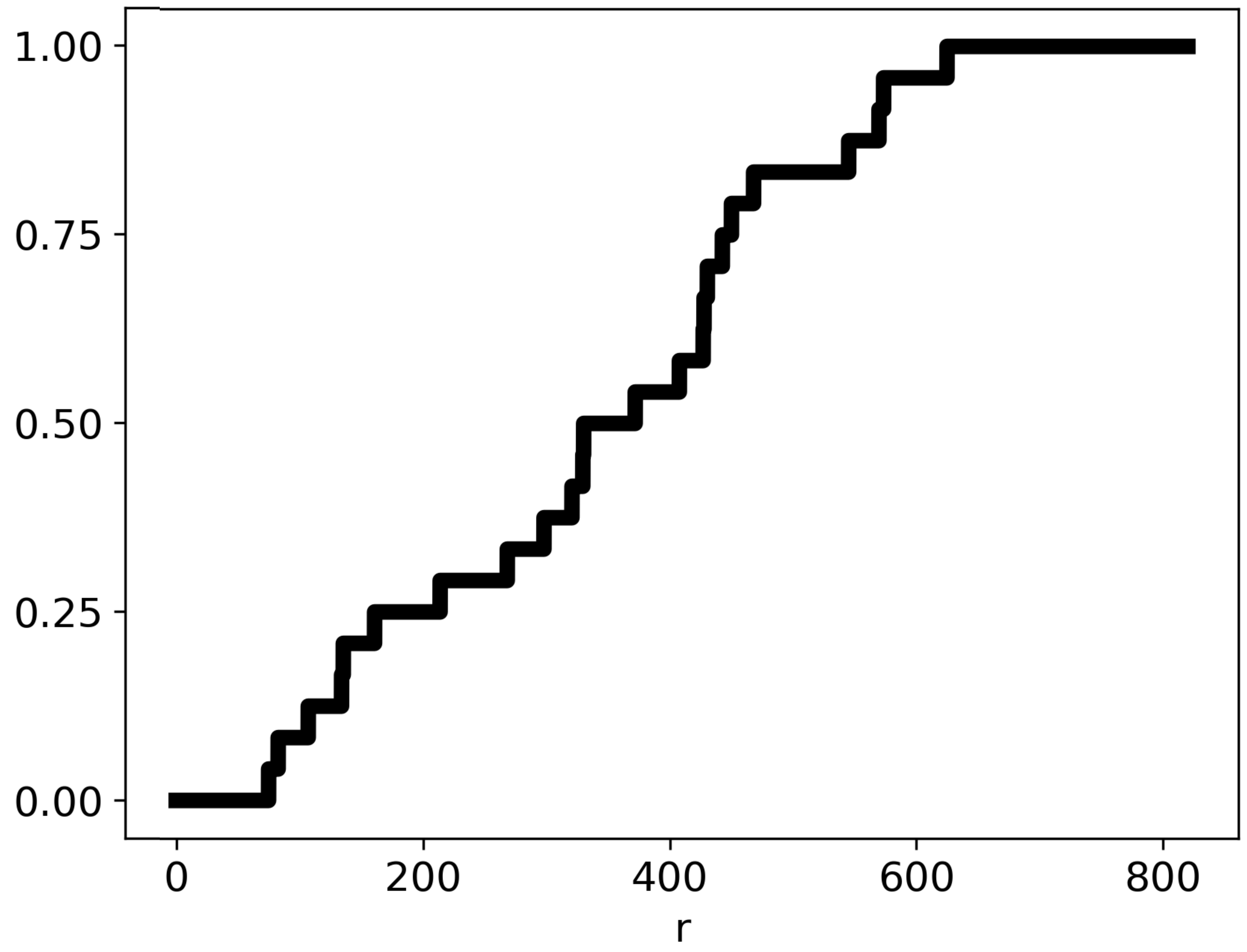


Ripley's K function

$$S(r) = \boxed{\frac{|\Omega|}{n_1 n_2}} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) b(i,j,r)$$

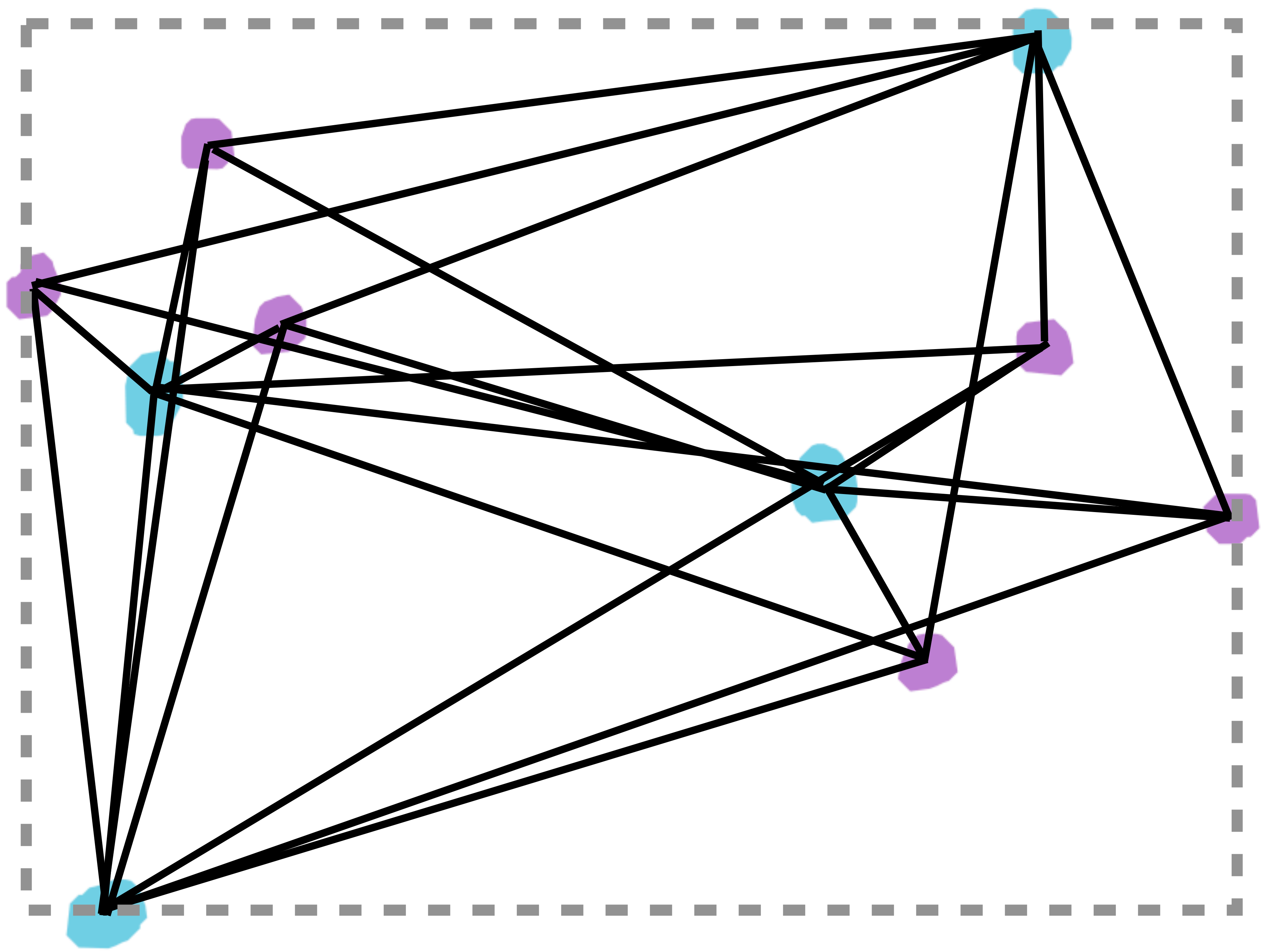


$$|\Omega| = \text{Area of FOV}$$

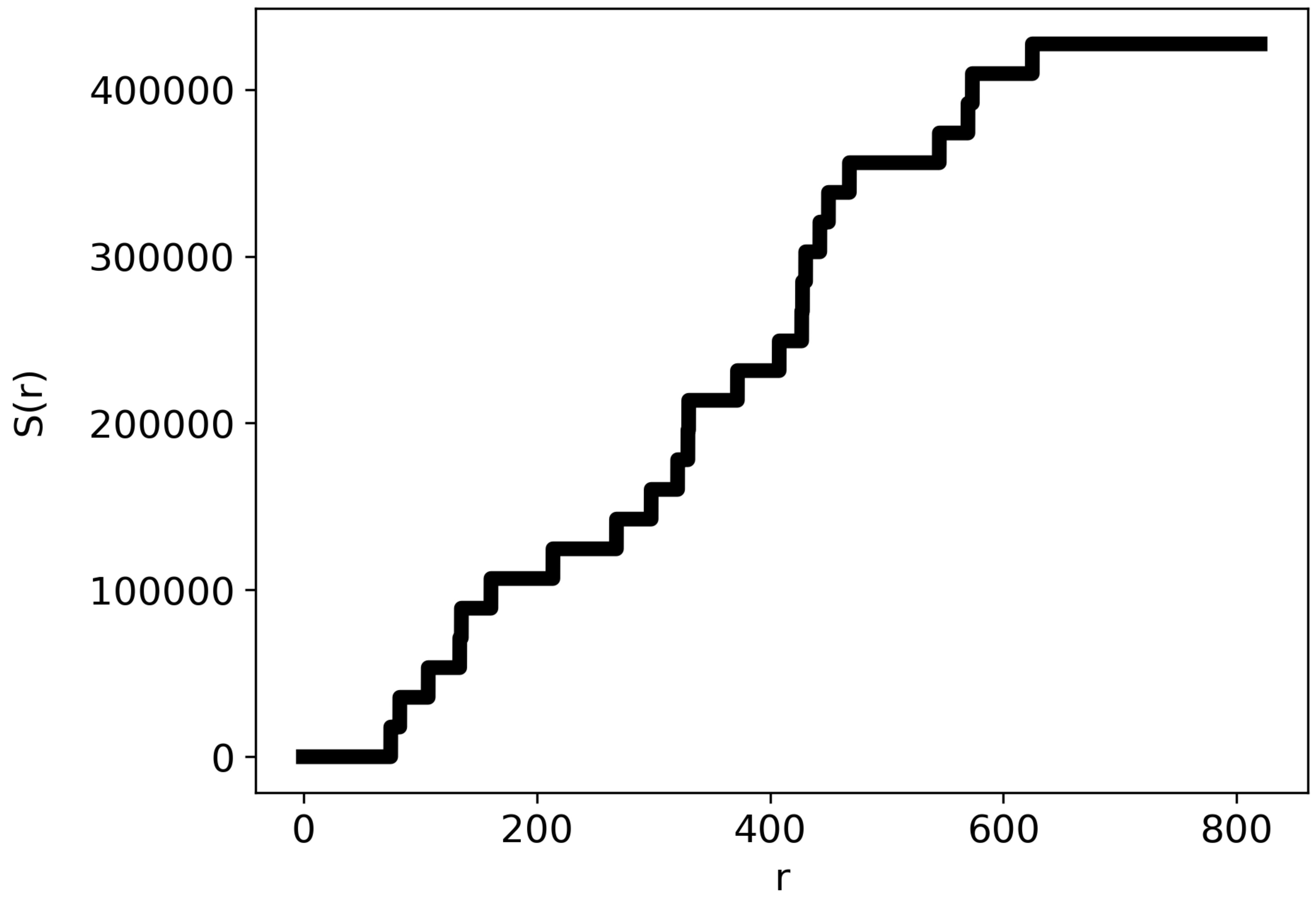


Ripley's K function

$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) b(i,j,r)$$



$$|\Omega| = \text{Area of FOV}$$

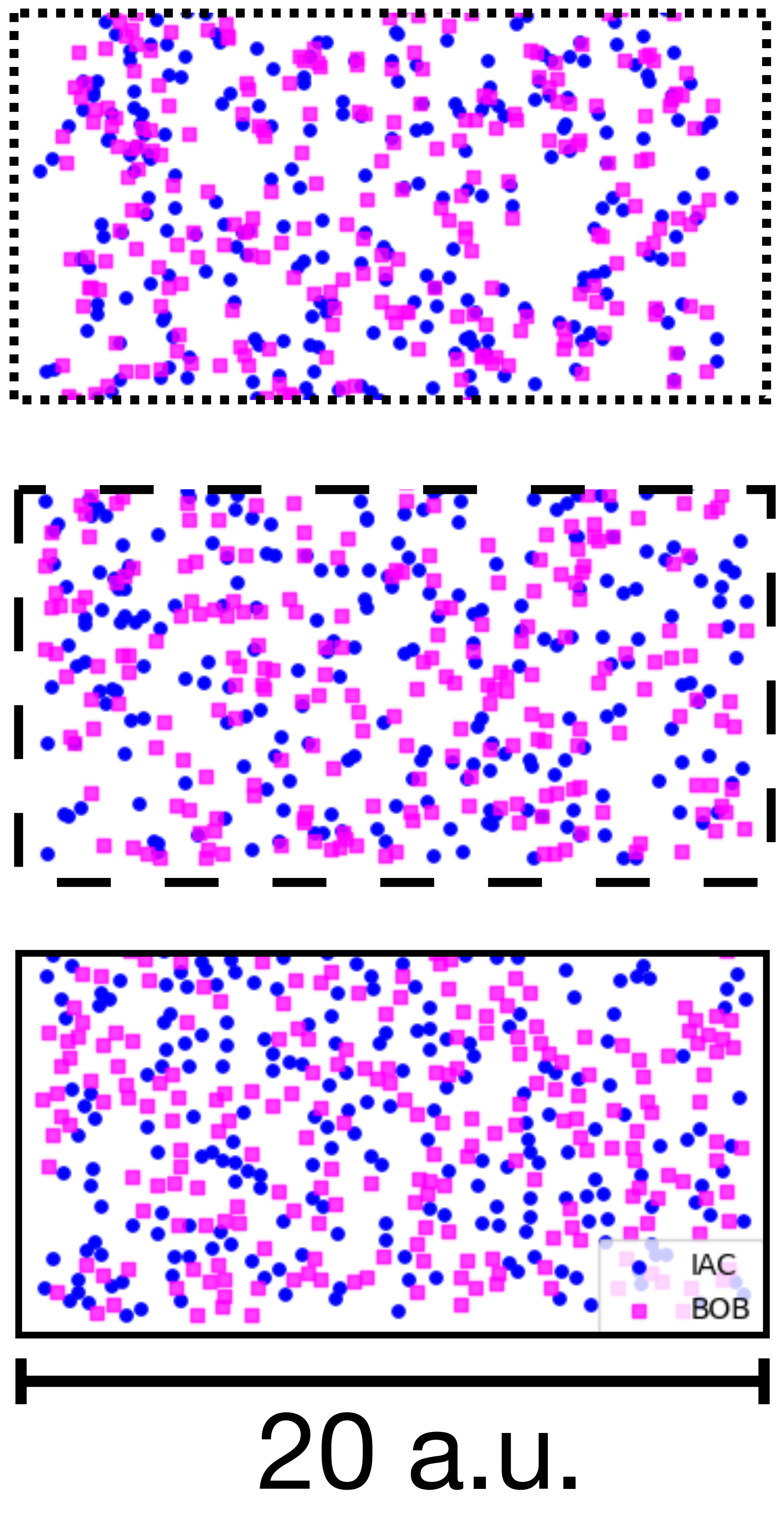


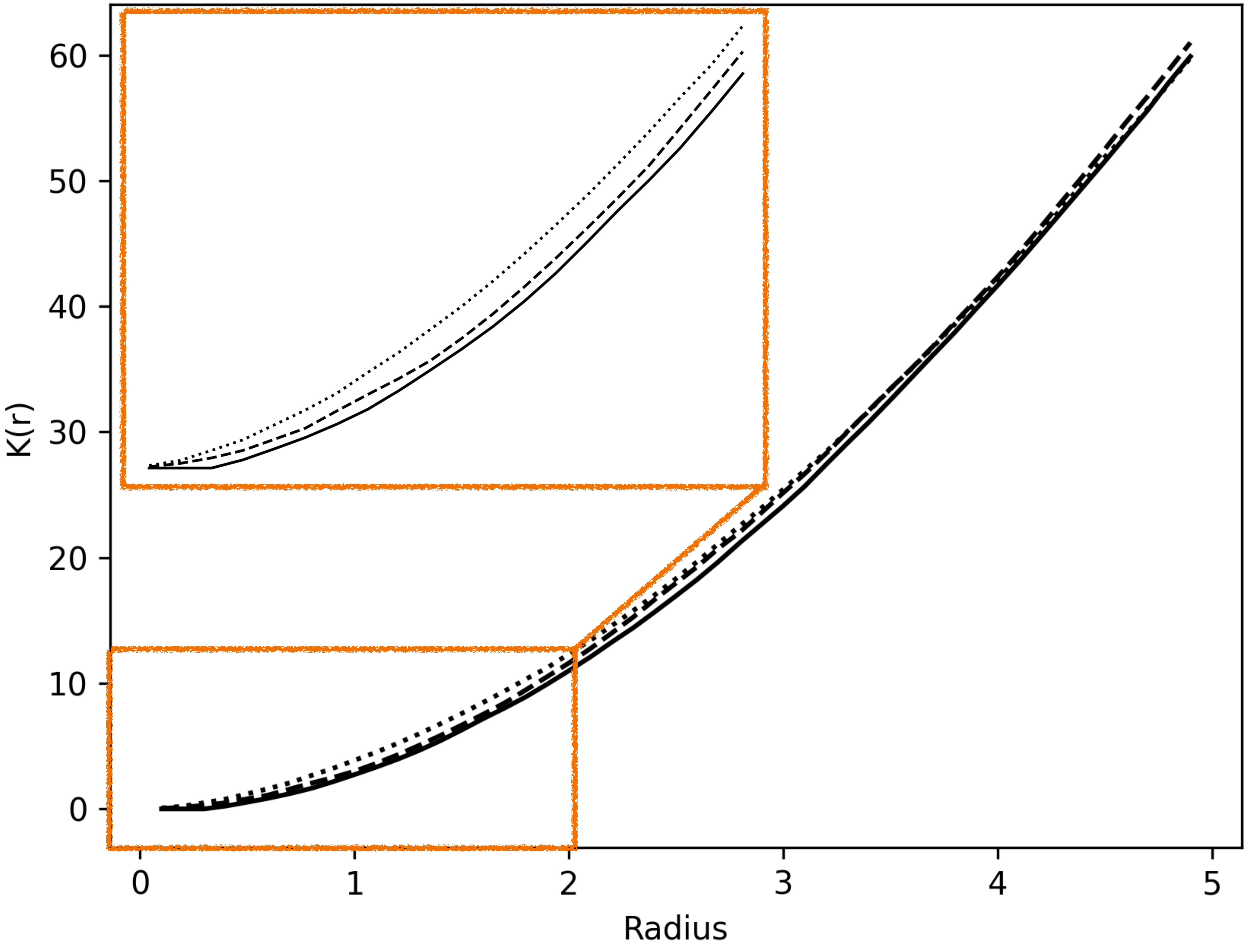
Results: Ripley's K function

Fall — cold

Fall — medium

Fall — warm



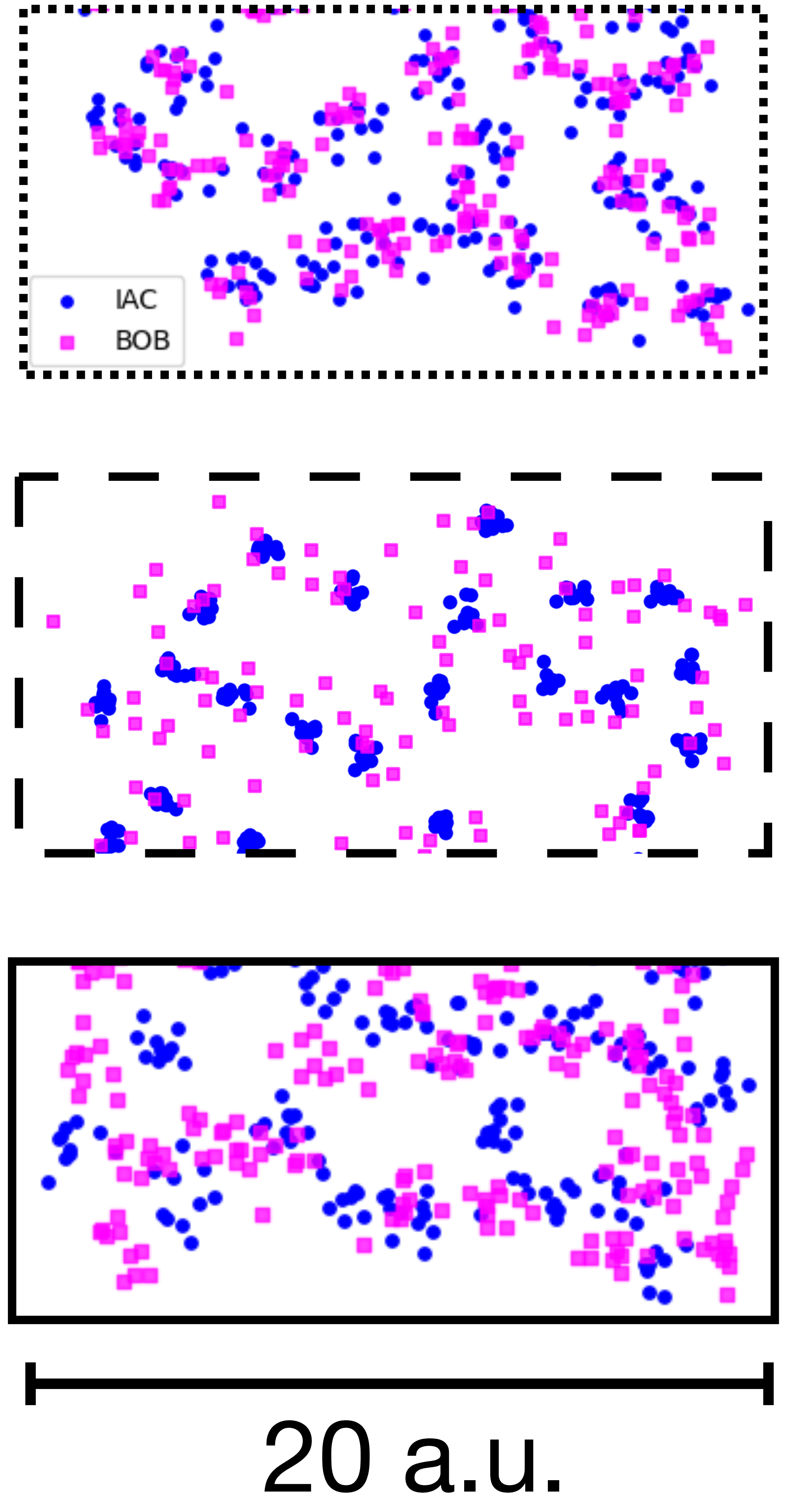


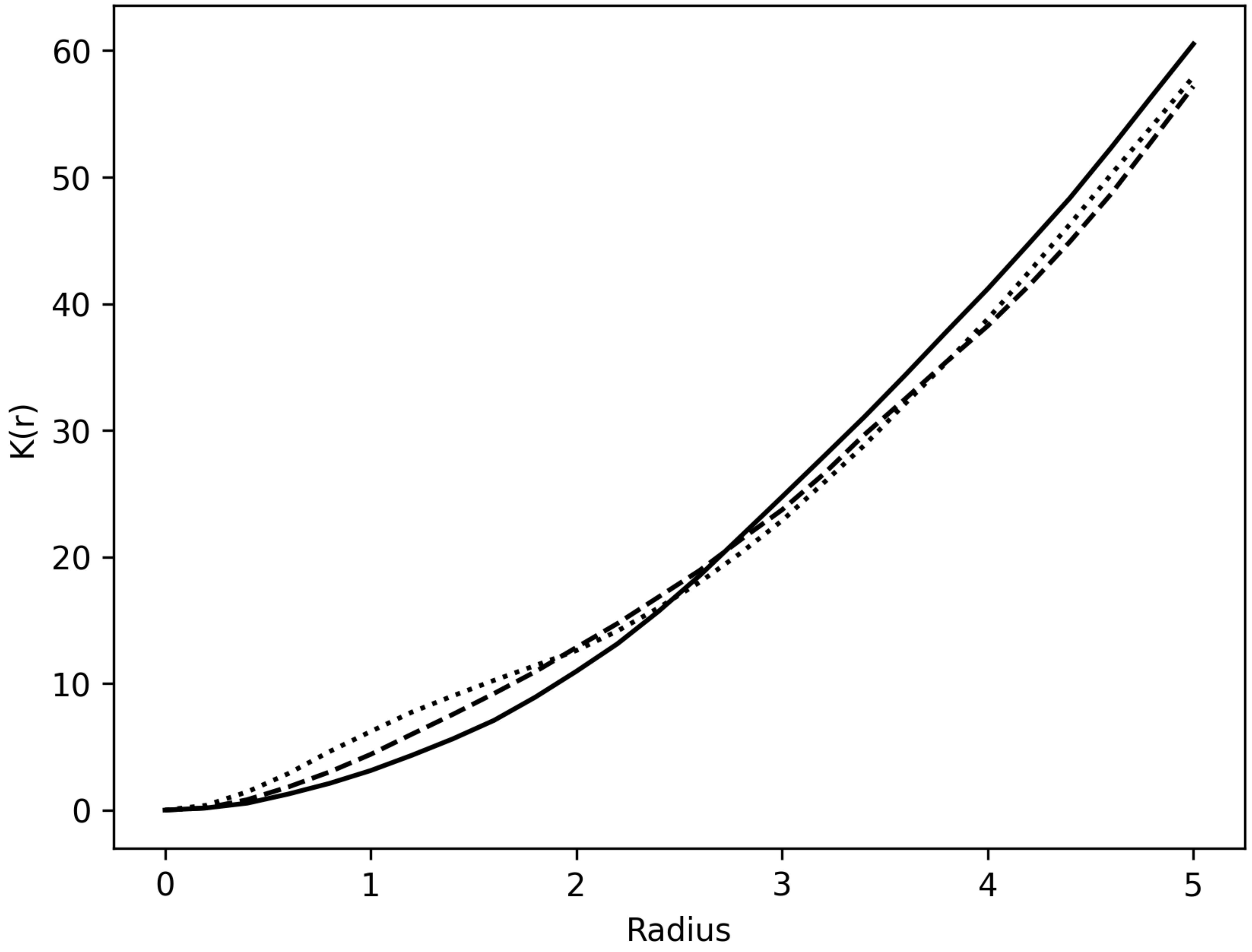
Results: Ripley's K function

Winter — cold

Winter — medium

Winter — warm



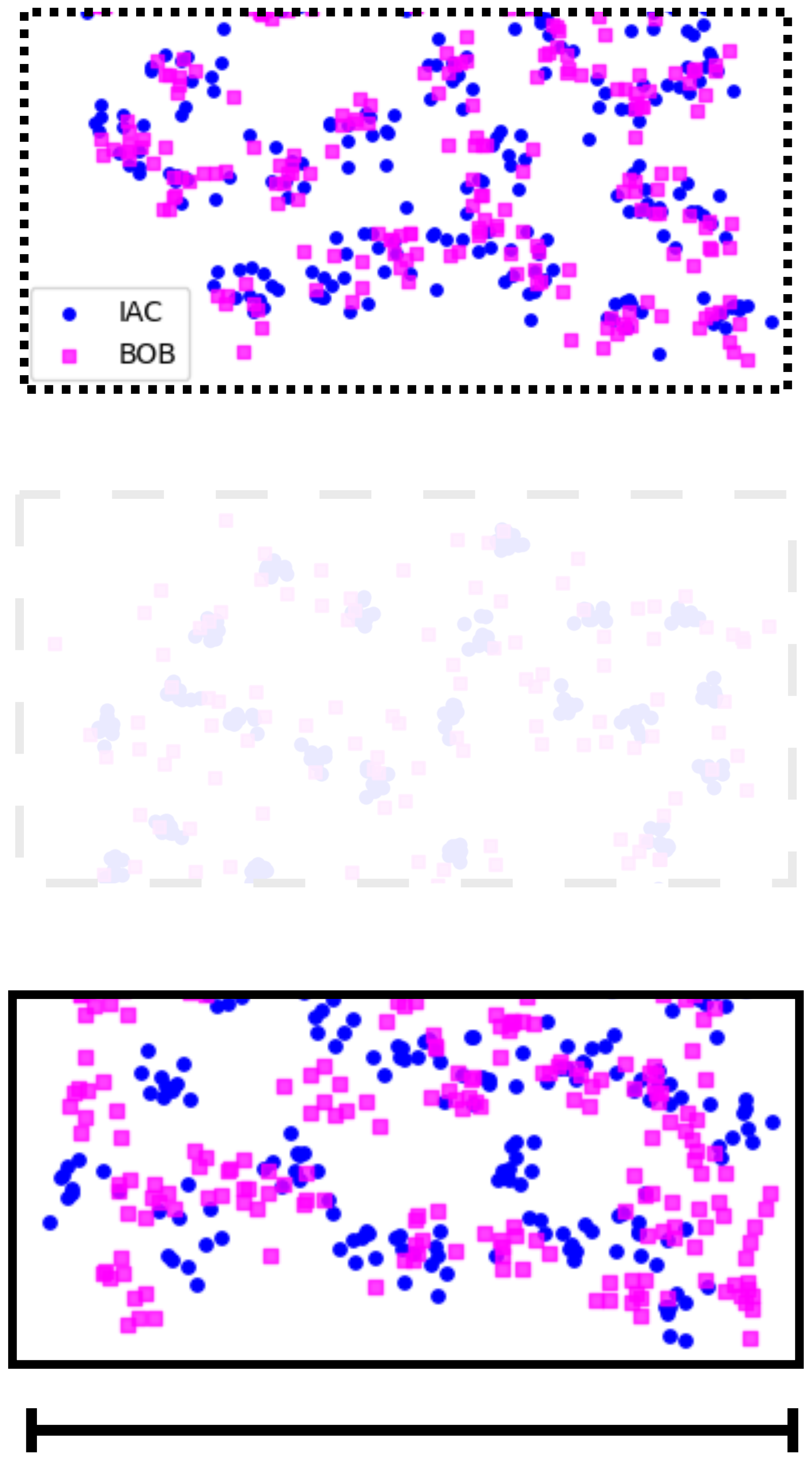


Results: Ripley's K function

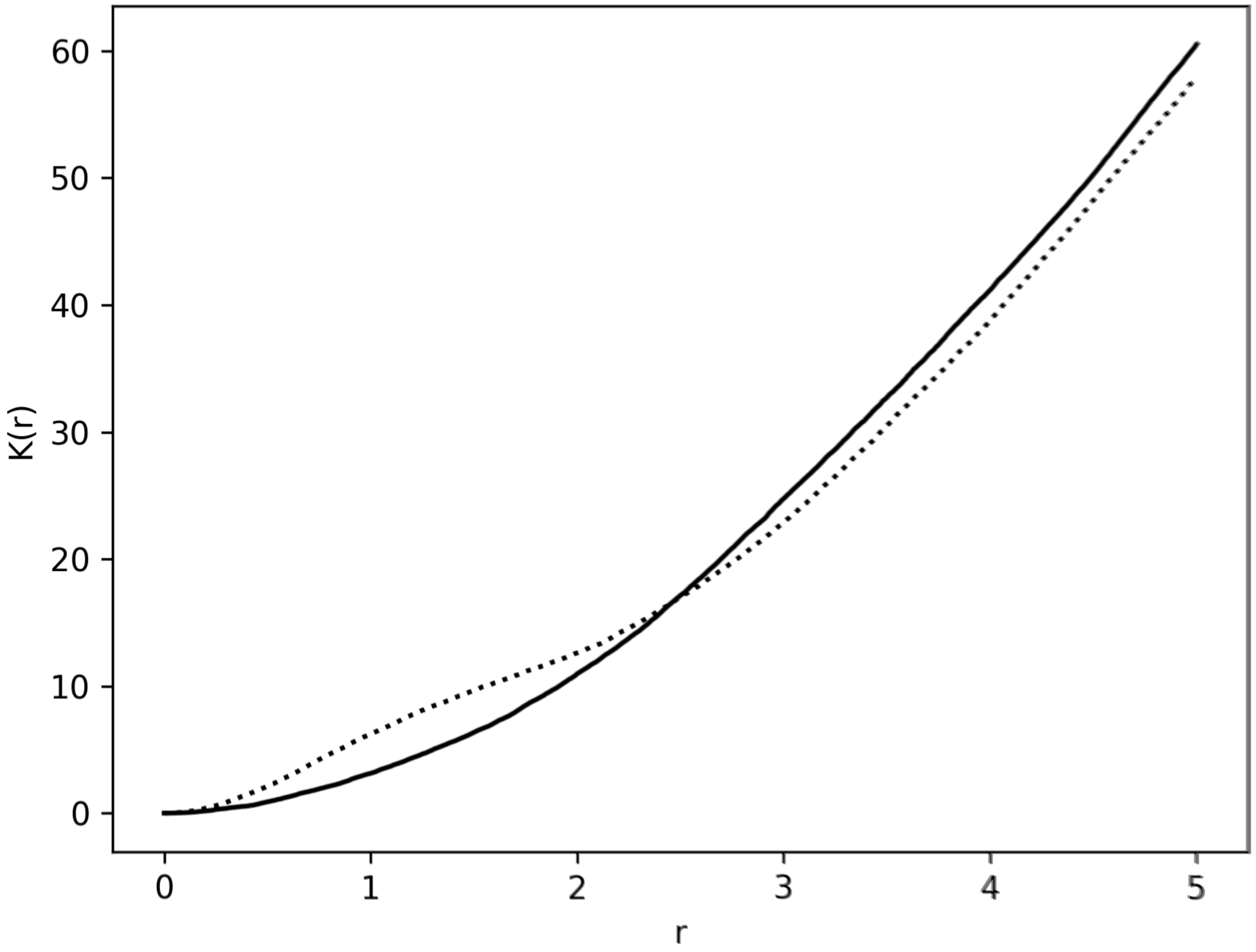
Winter — cold

Winter — medium

Winter — warm



20 a.u.

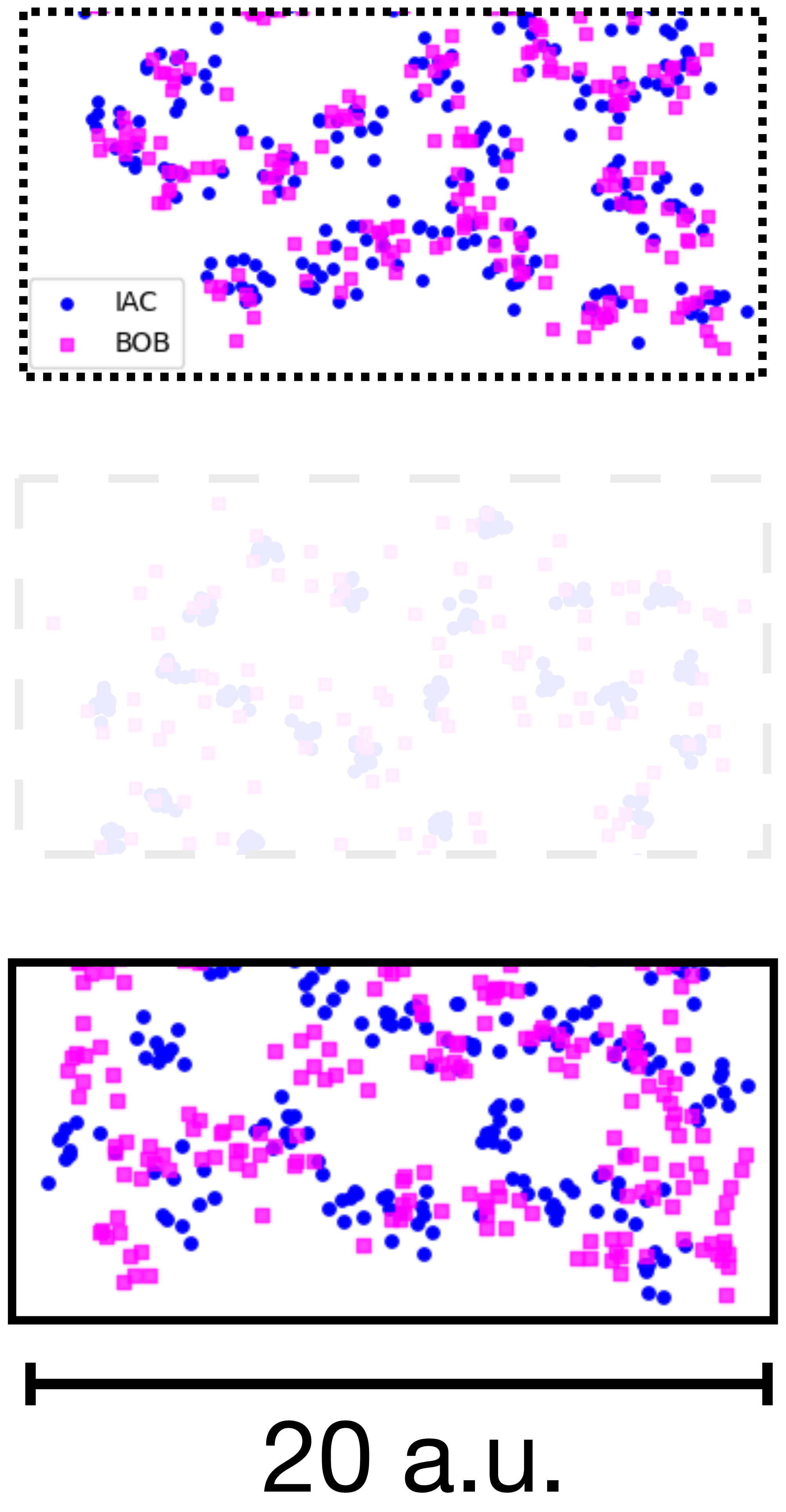


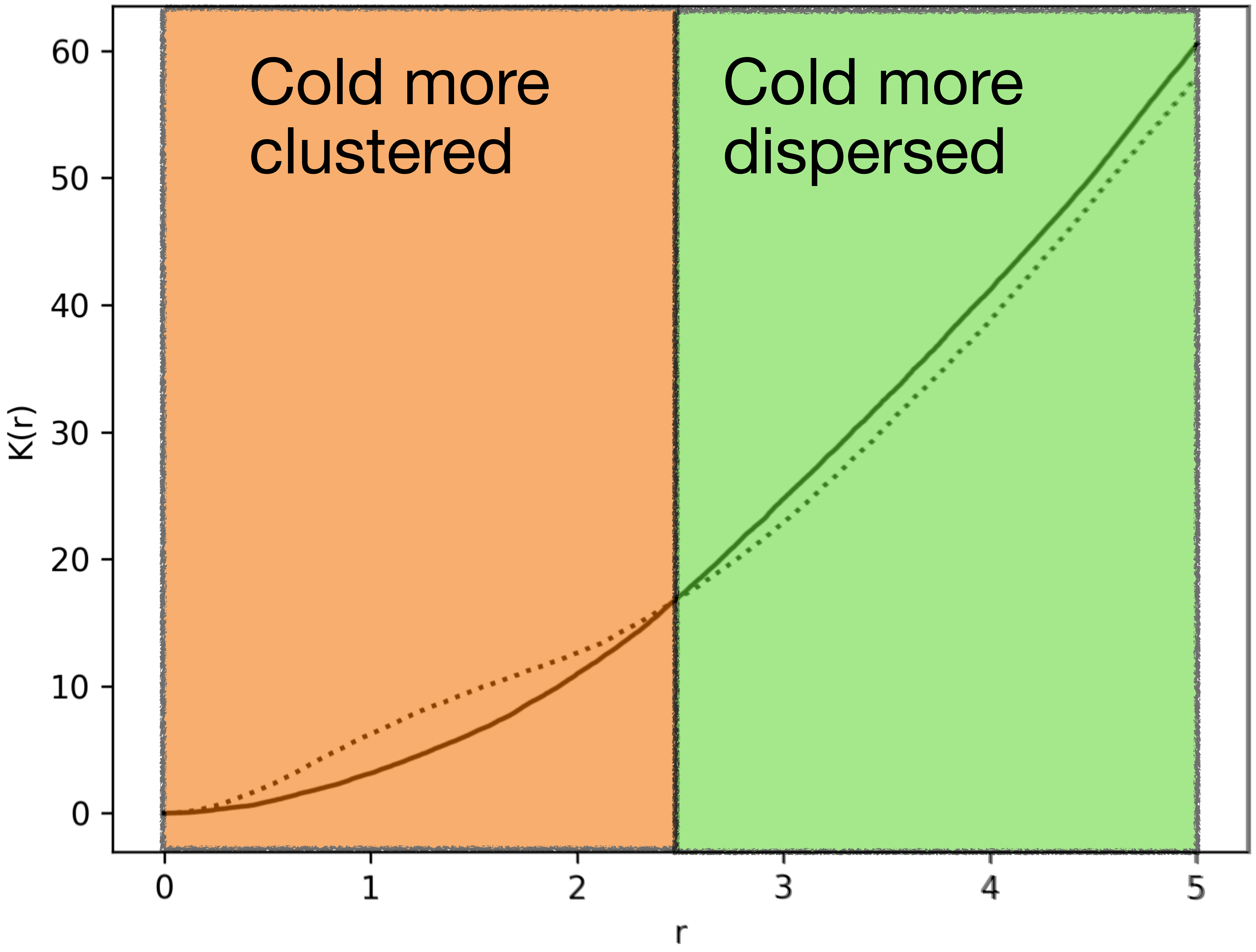
Results: Ripley's K function

Winter — cold

Winter — medium

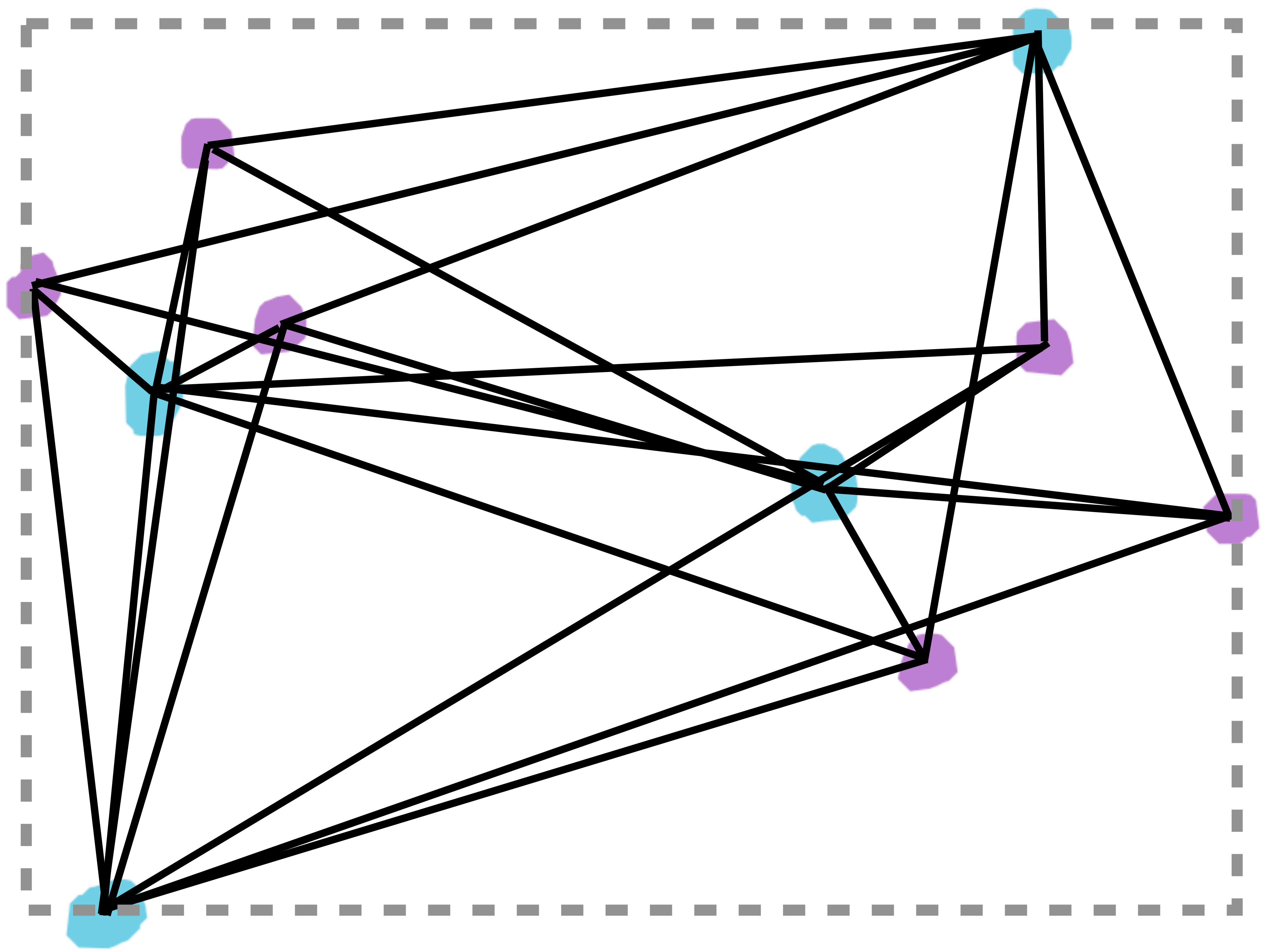
Winter — warm



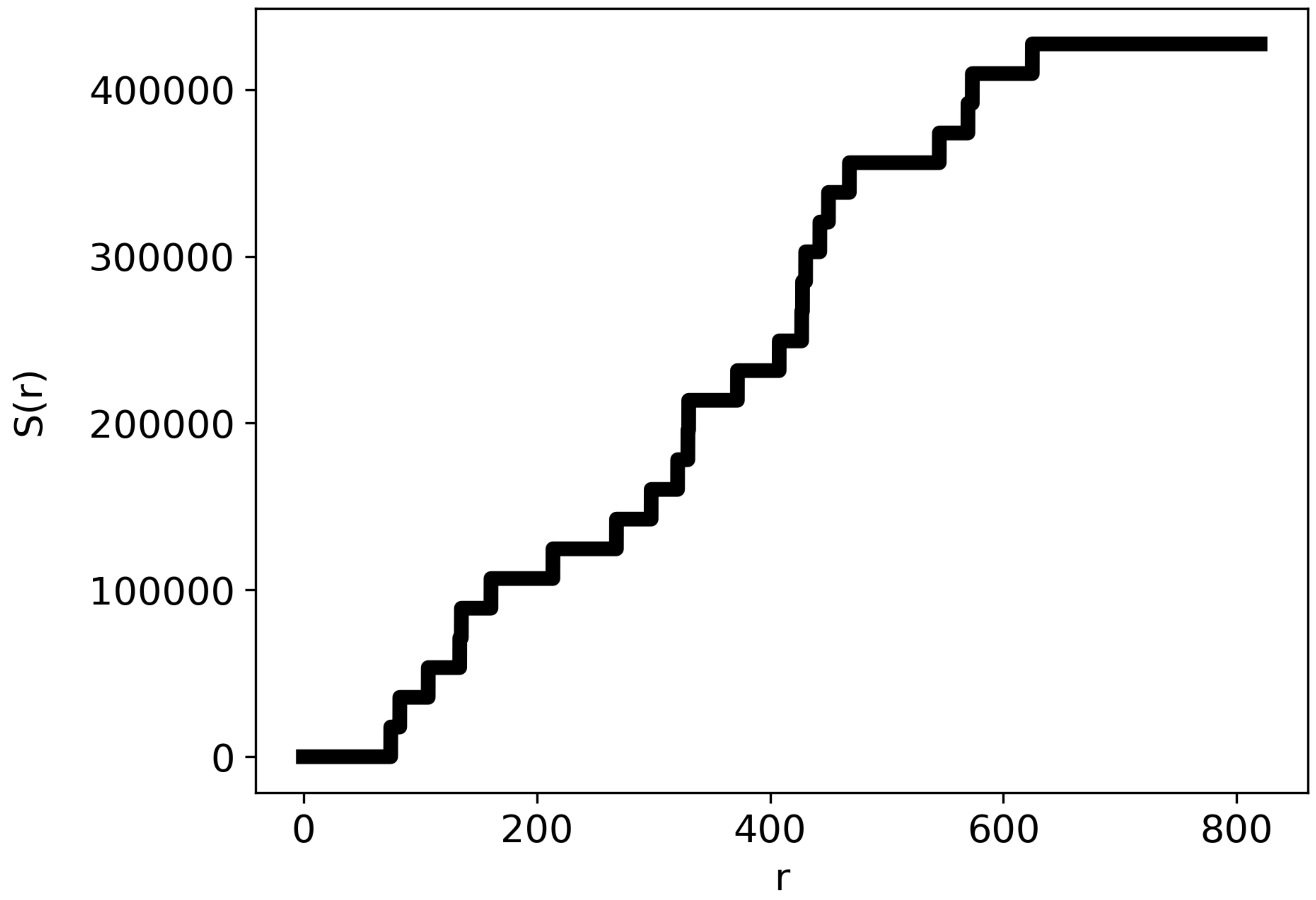


Ripley's K function

$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) \boxed{b(i,j,r)}$$

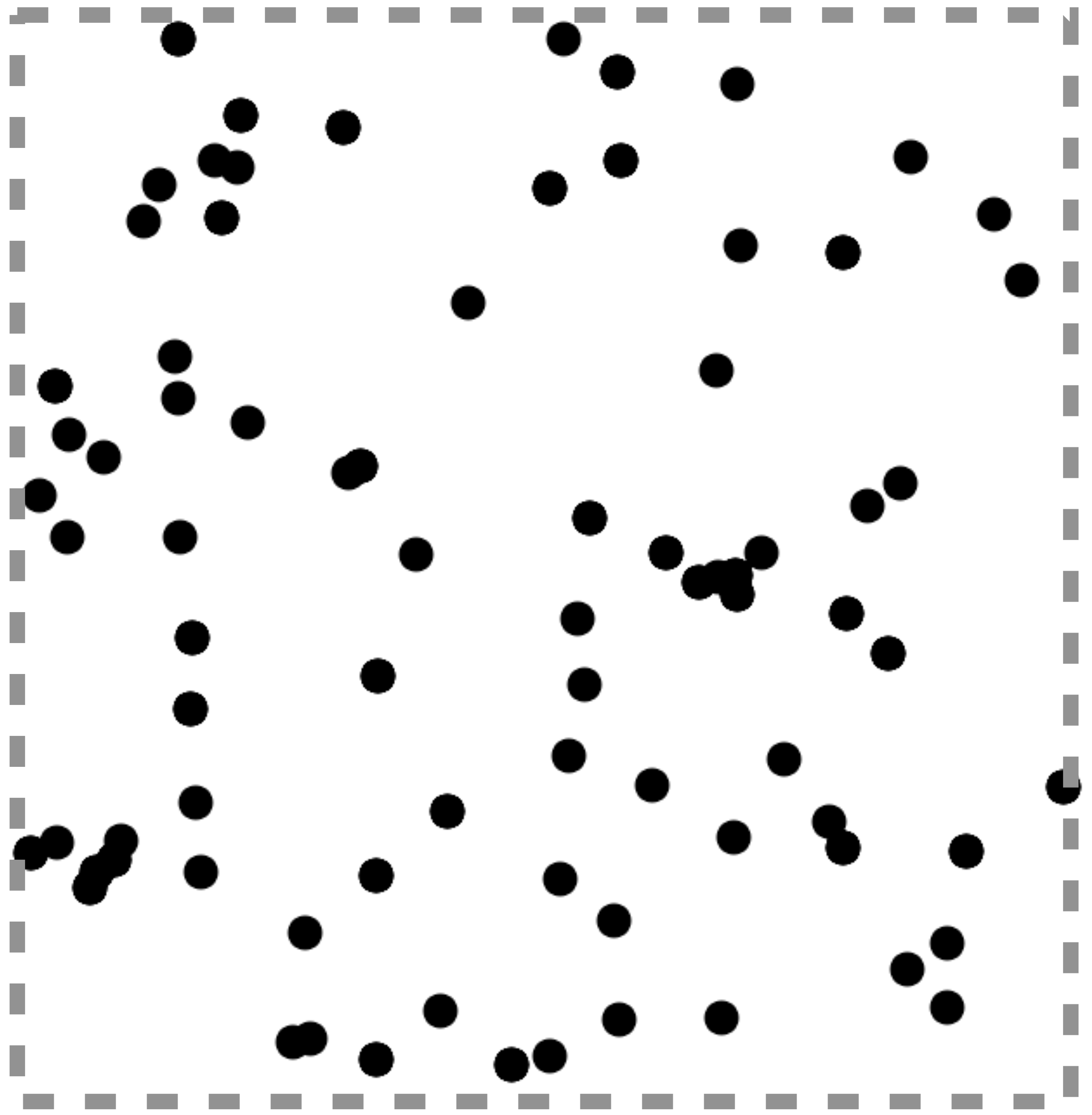


$|\Omega| = \text{Area of FOV}$



Ripley's K function

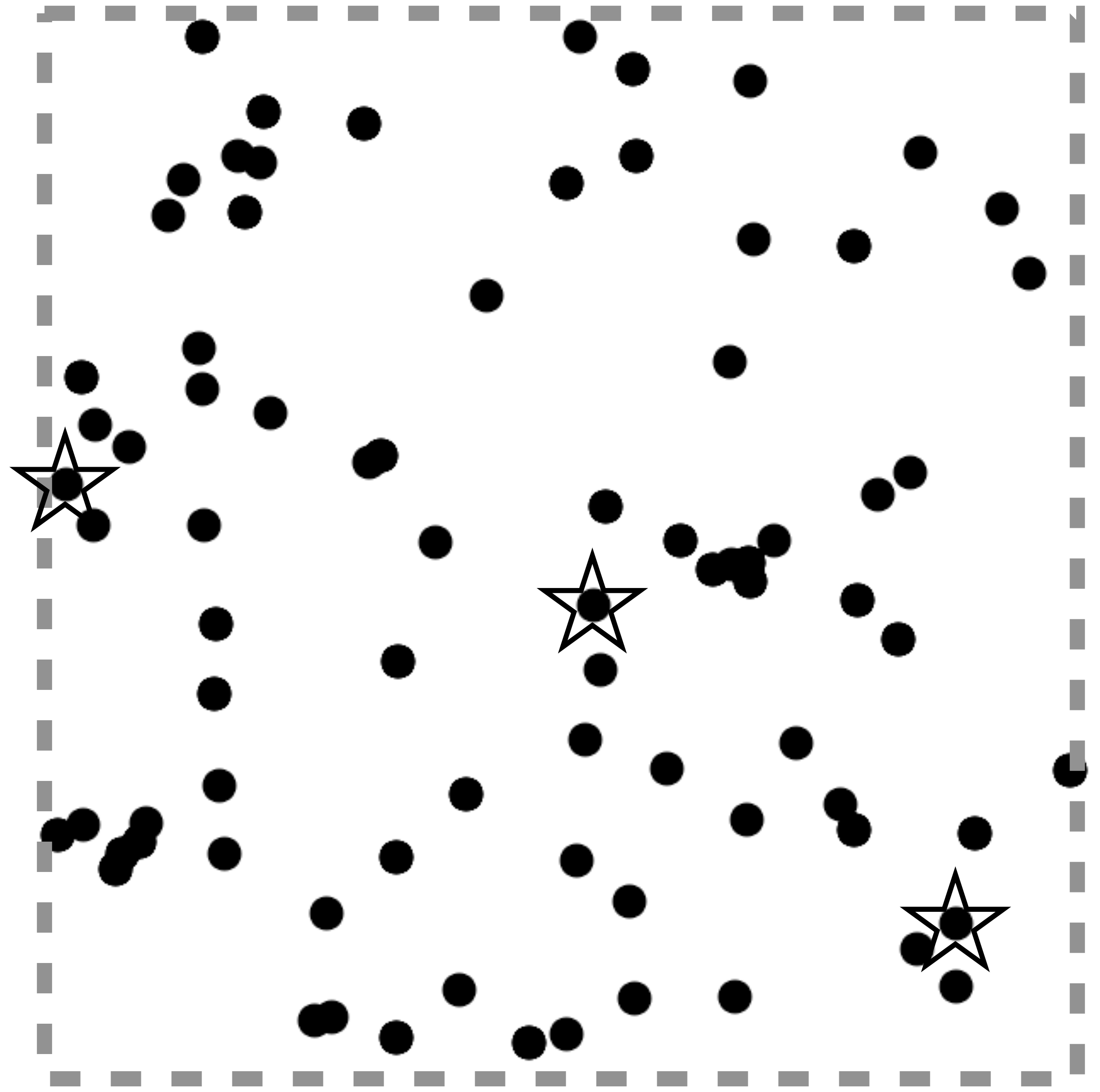
$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) \boxed{b(i,j,r)}$$



$$b(i,j,r) = \frac{|c(i, d_{ij})|}{|c(i, d_{ij}) \cap \Omega|}$$

Ripley's K function

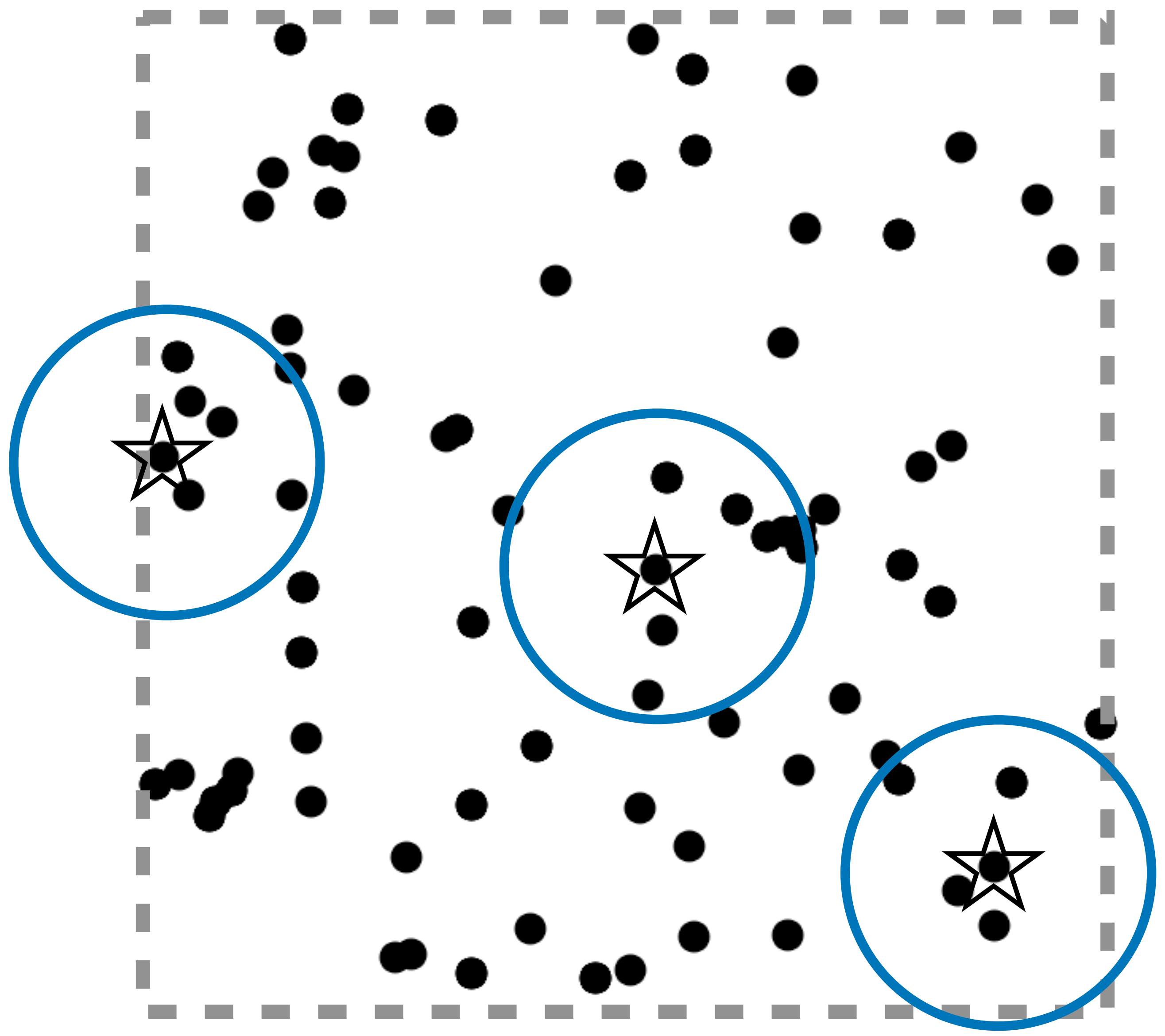
$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r)\boxed{b(i, j, r)}$$



$$b(i, j, r) = \frac{|c(i, d_{ij})|}{|c(i, d_{ij}) \cap \Omega|}$$

Ripley's K function

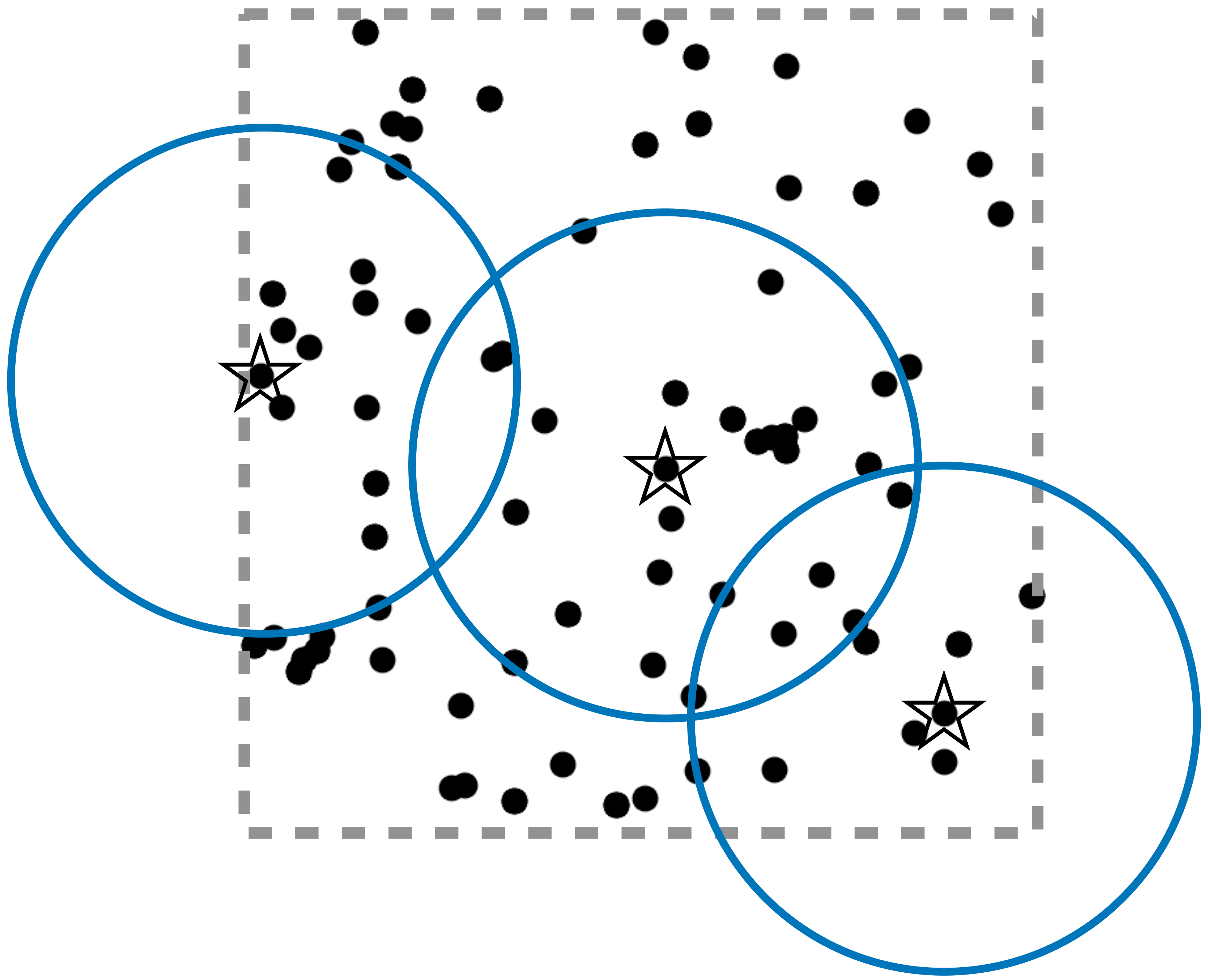
$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) \boxed{b(i, j, r)}$$



$$b(i, j, r) = \frac{|c(i, d_{ij})|}{|c(i, d_{ij}) \cap \Omega|}$$

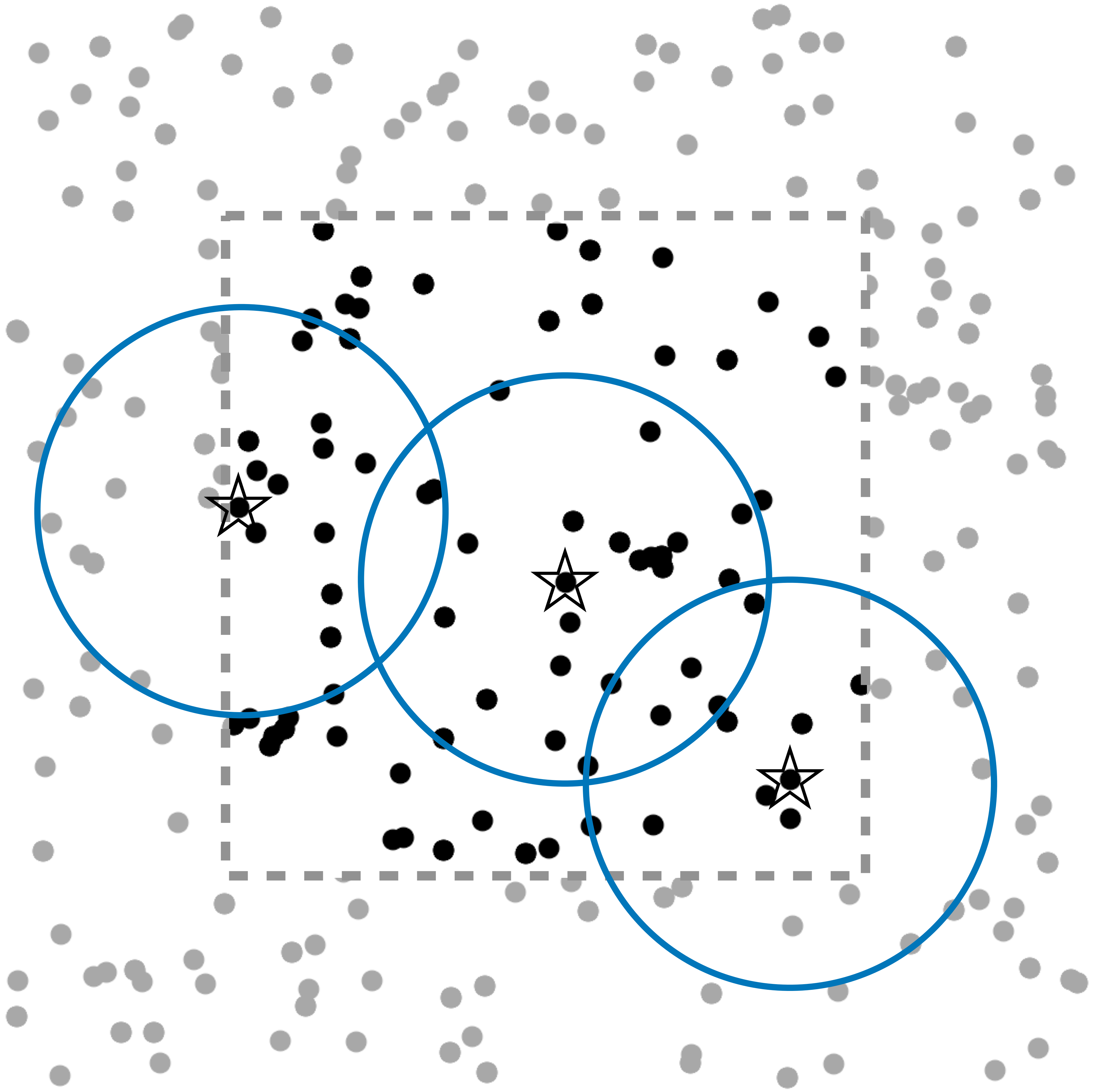
Ripley's K function

$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) \boxed{b(i,j,r)}$$



$$b(i,j,r) = \frac{|c(i, d_{ij})|}{|c(i, d_{ij}) \cap \Omega|}$$

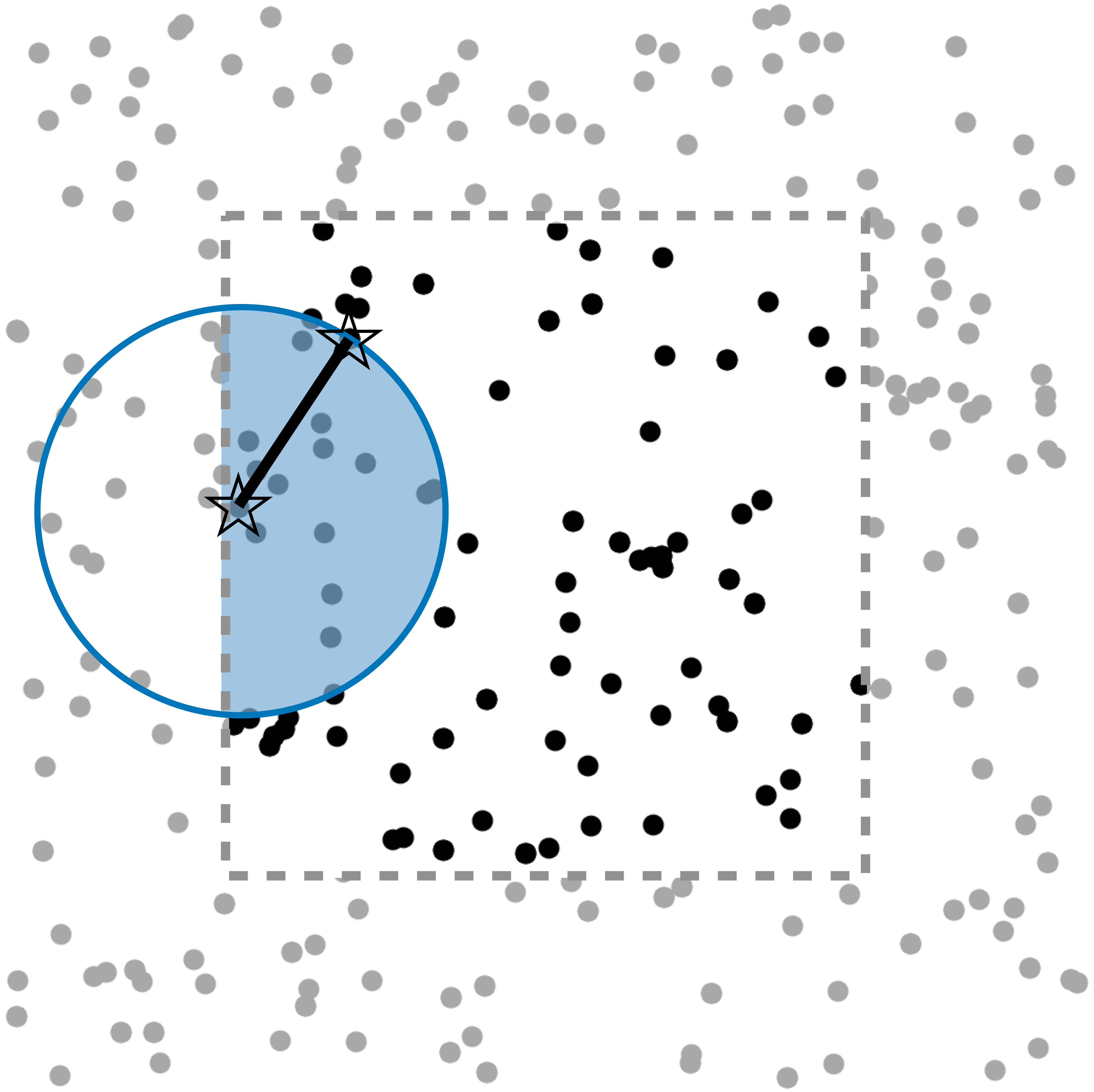
Ripley's K function



$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) \boxed{b(i,j,r)}$$

$$b(i,j,r) = \frac{|c(i, d_{ij})|}{|c(i, d_{ij}) \cap \Omega|}$$

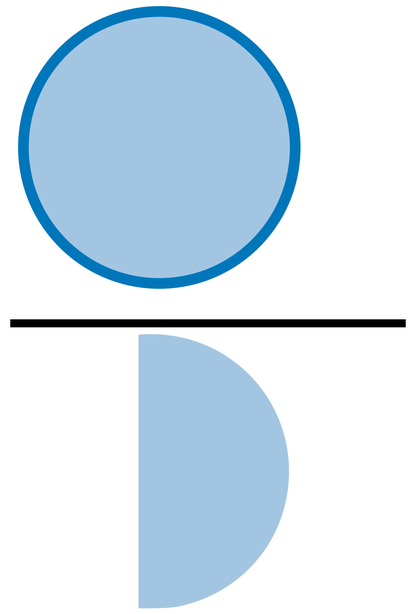
Ripley's K function



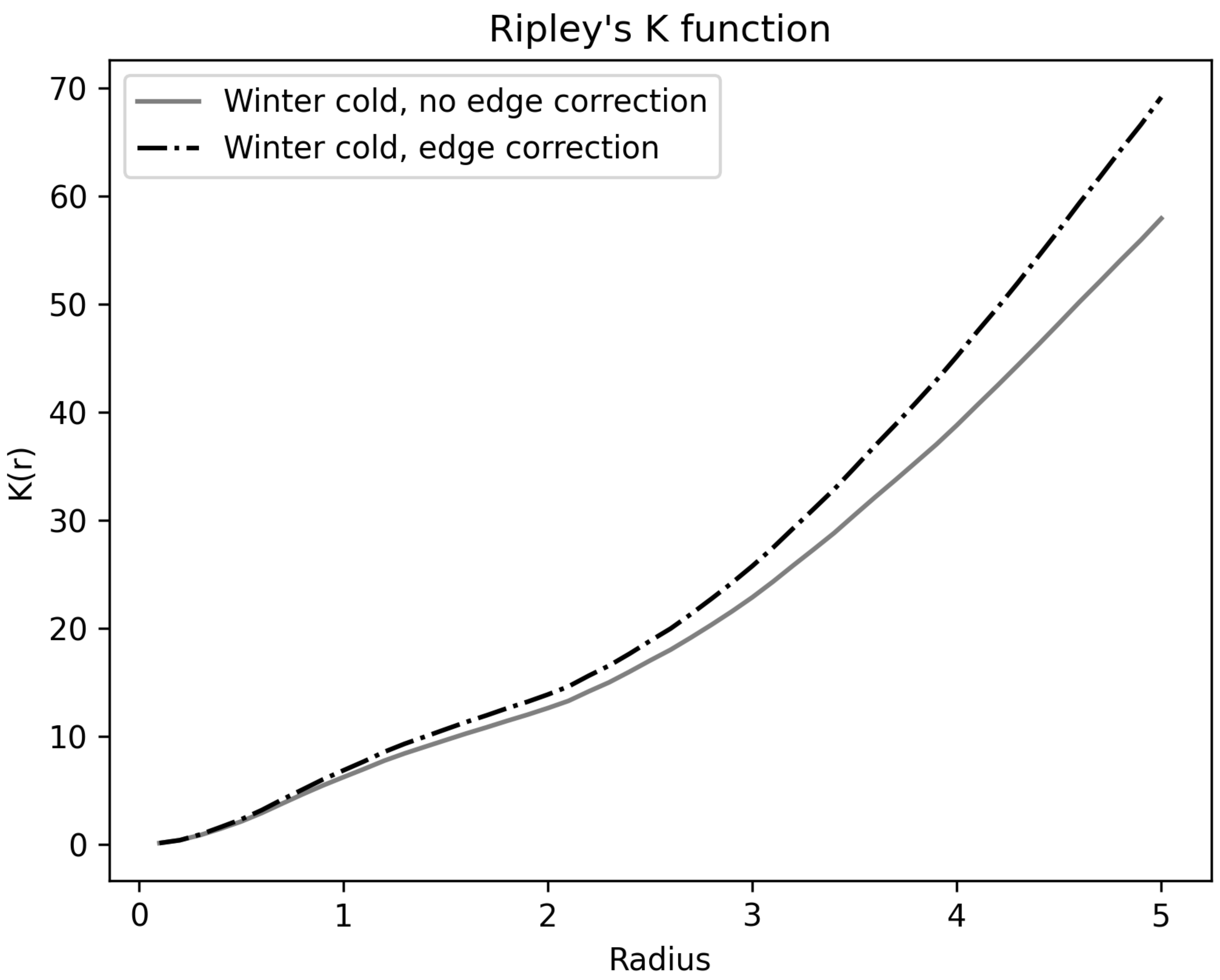
$$S(r) = \frac{|\Omega|}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbf{1}(d_{ij} < r) \boxed{b(i,j,r)}$$

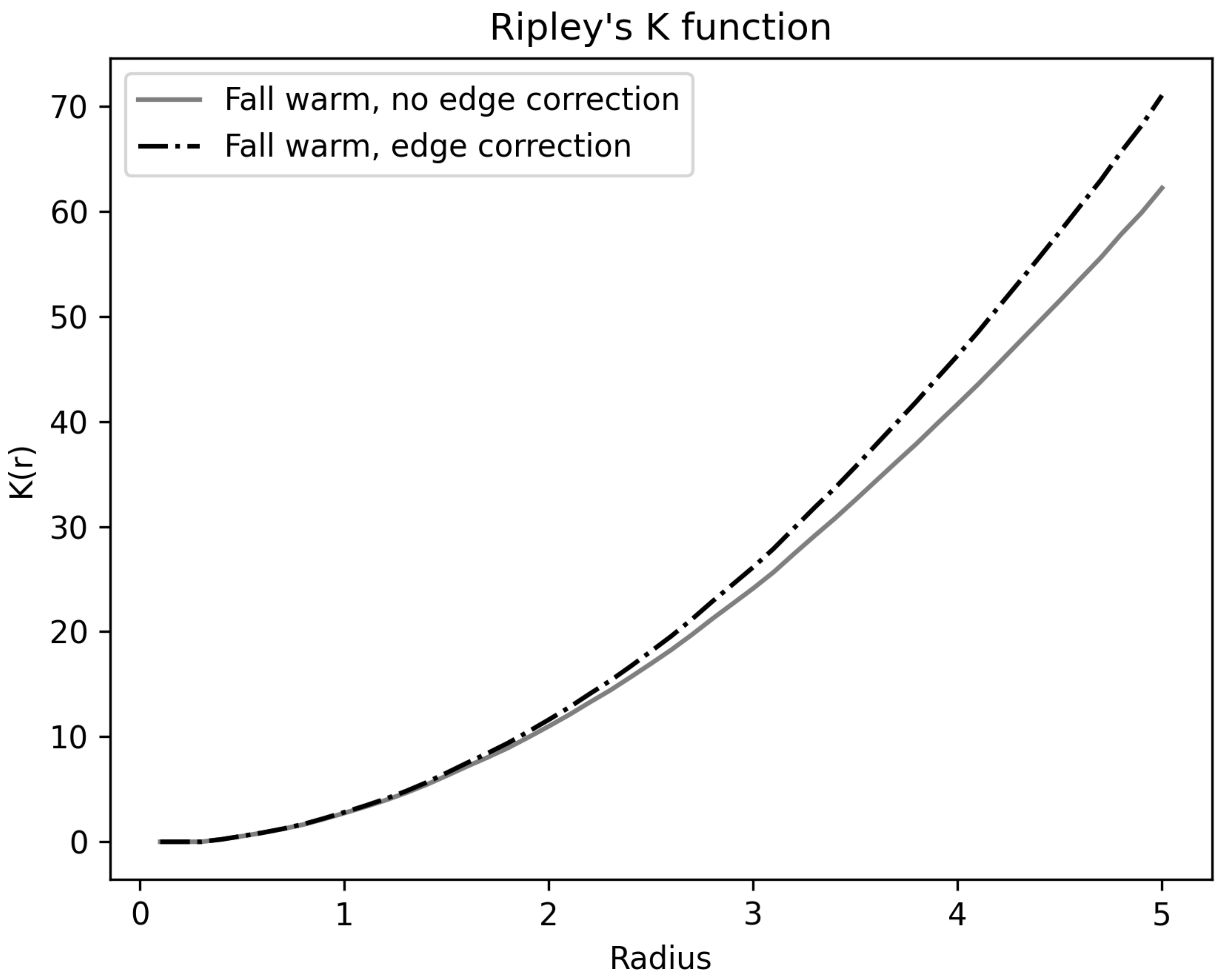
$$b(i,j,r) = \frac{|c(i,d_{ij})|}{|c(i,d_{ij}) \cap \Omega|}$$

$=$



Ripley’s K function



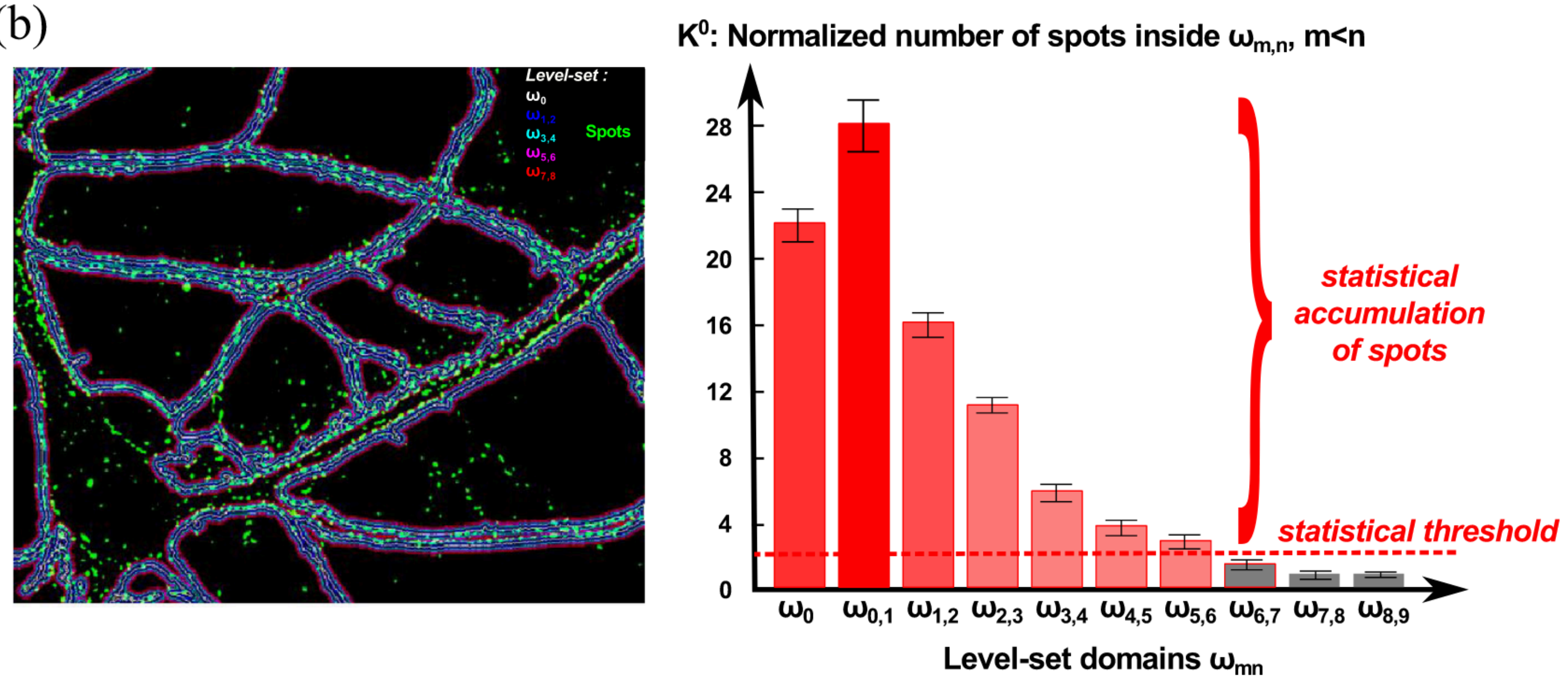


Ripley's K function

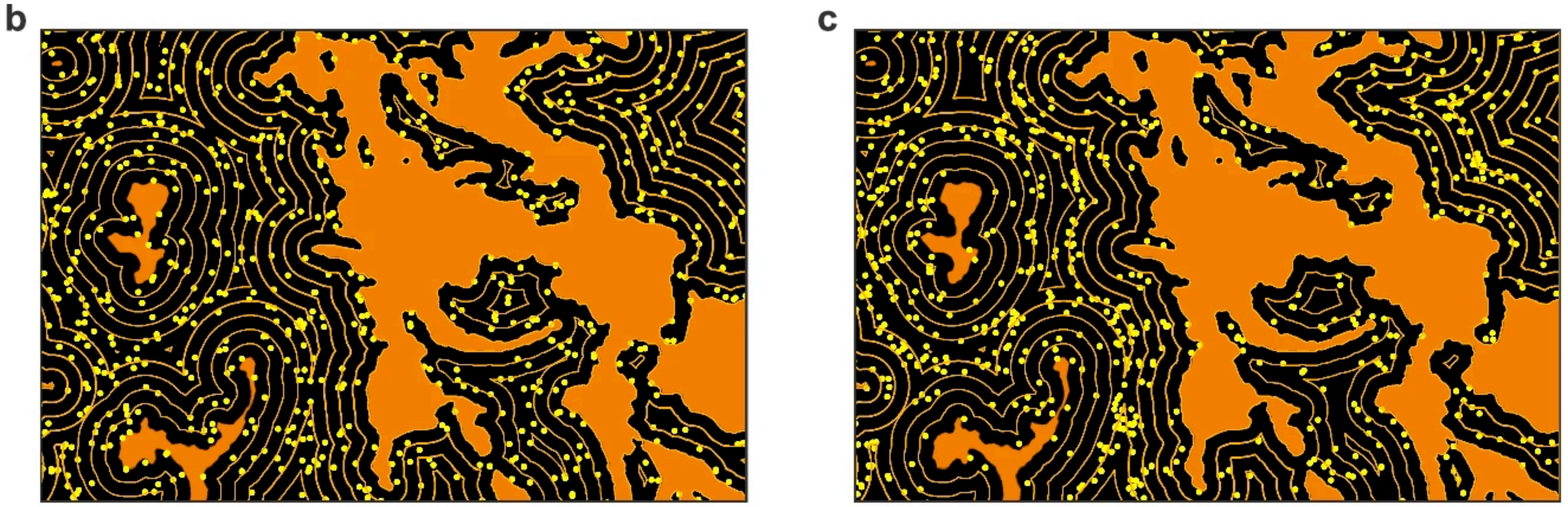
- Symmetric: BOB → IAC = IAC → BOB
- Returns: A number for each radius
- Range: Long

Beyond Ripley’s K function

(b)



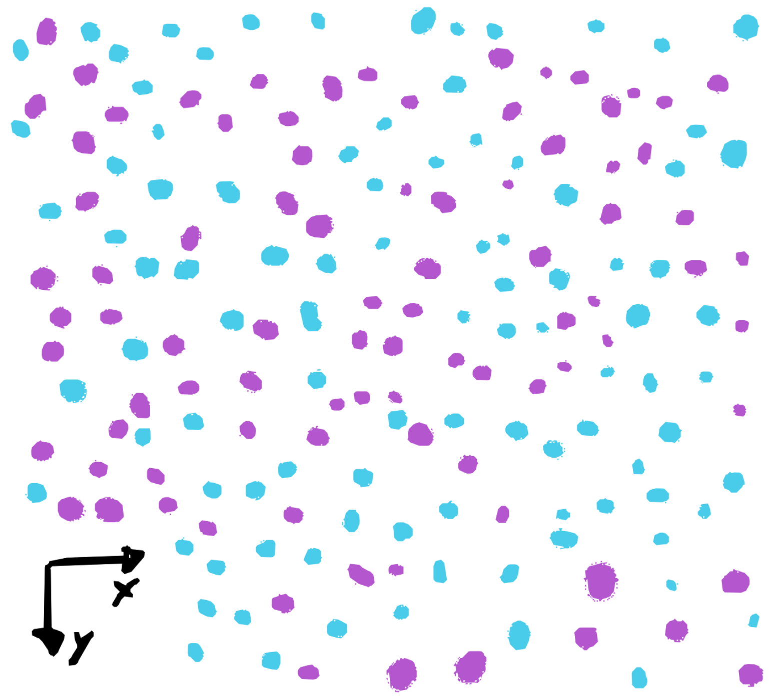
S. Mukherjee, C. Gonzalez-Gomez, L. Danglot, T. Lagache and J. -C. Olivo-Marin, "Generalizing the Statistical Analysis of Objects’ Spatial Coupling in Bioimaging," in *IEEE Signal Processing Letters*, vol. 27, pp. 1085-1089, 2020, doi: 10.1109/LSP.2020.3003821.



Benimam, M.M., Meas-Yedid, V., Mukherjee, S. *et al.* Statistical analysis of spatial patterns in tumor microenvironment images. *Nat Commun* **16**, 3090 (2025). https://doi.org/10.1038/s41467-025-57943-y

Summary

Short-range



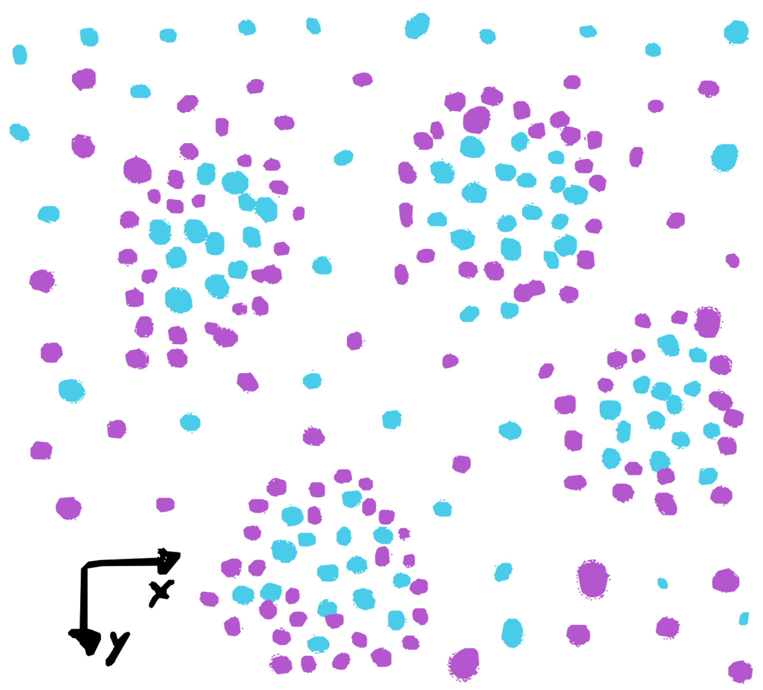
One number

Mean dist to nearest neighbor

One number
per radius

Nearest neighbor function

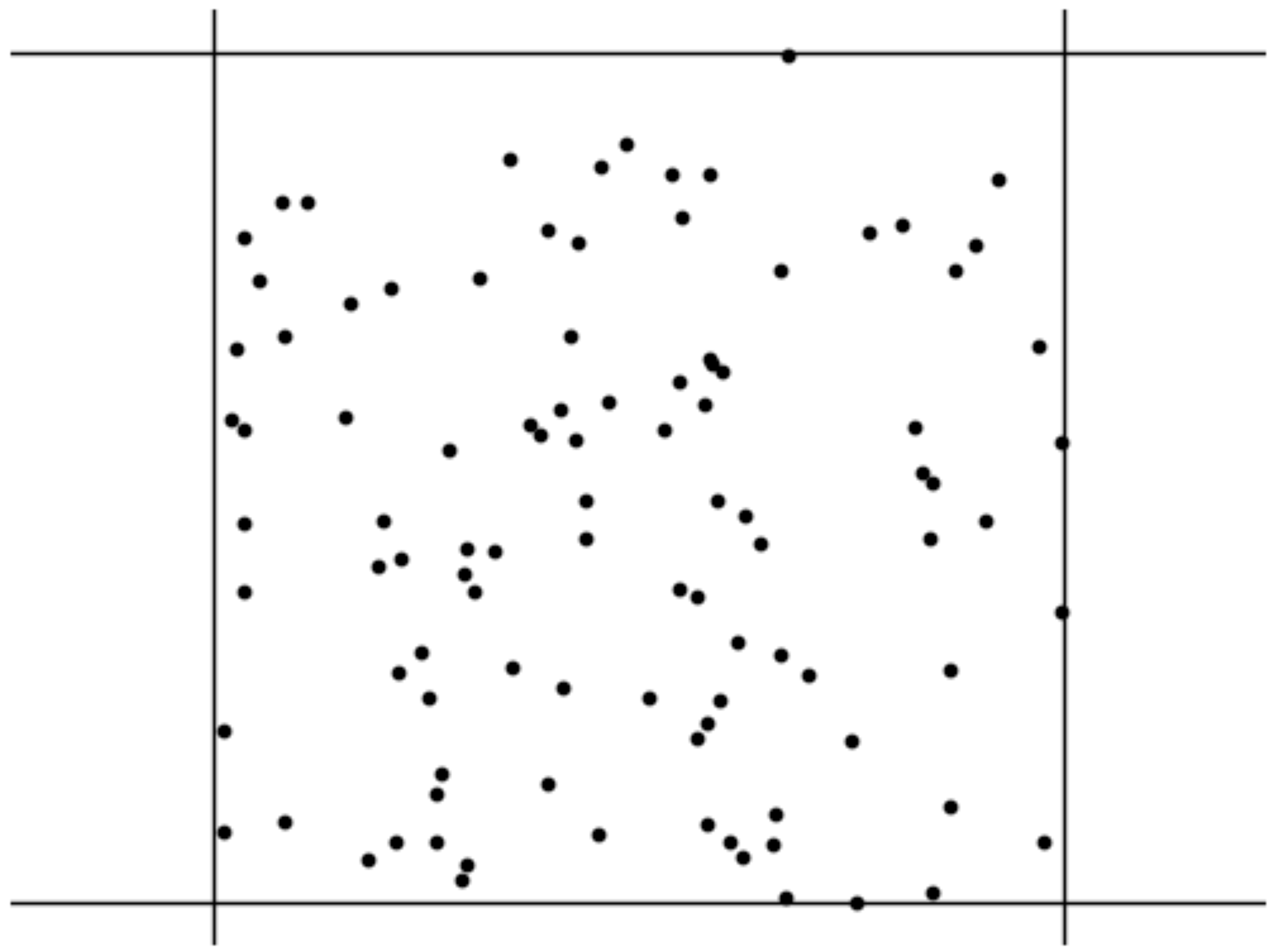
Long-range



Ripley’s K

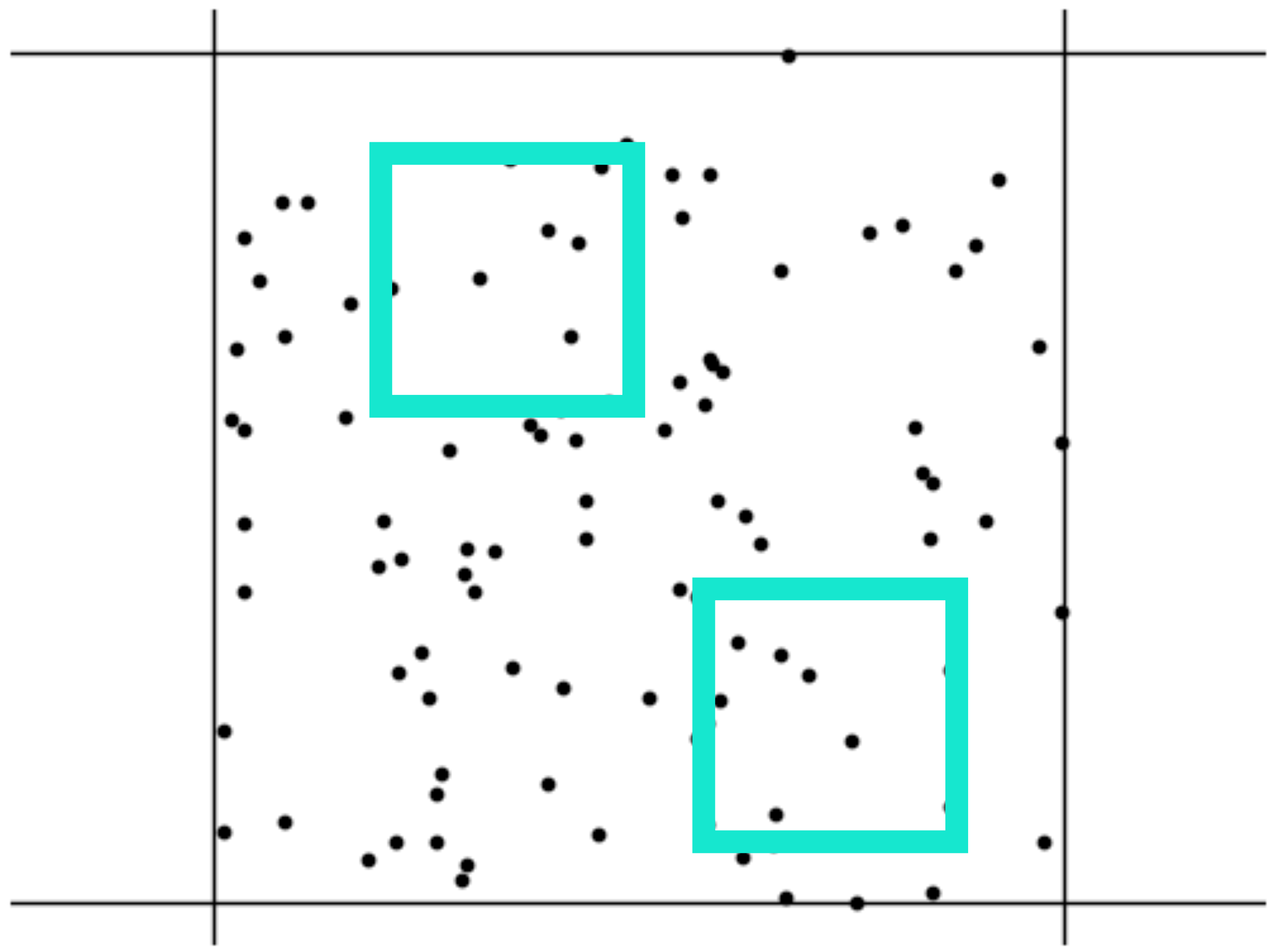
Validation — the null distribution

- How are proteins distributed that
  - Don't interact with each other
  - Or their surrounding
- To find out, you blindly throw darts at a board



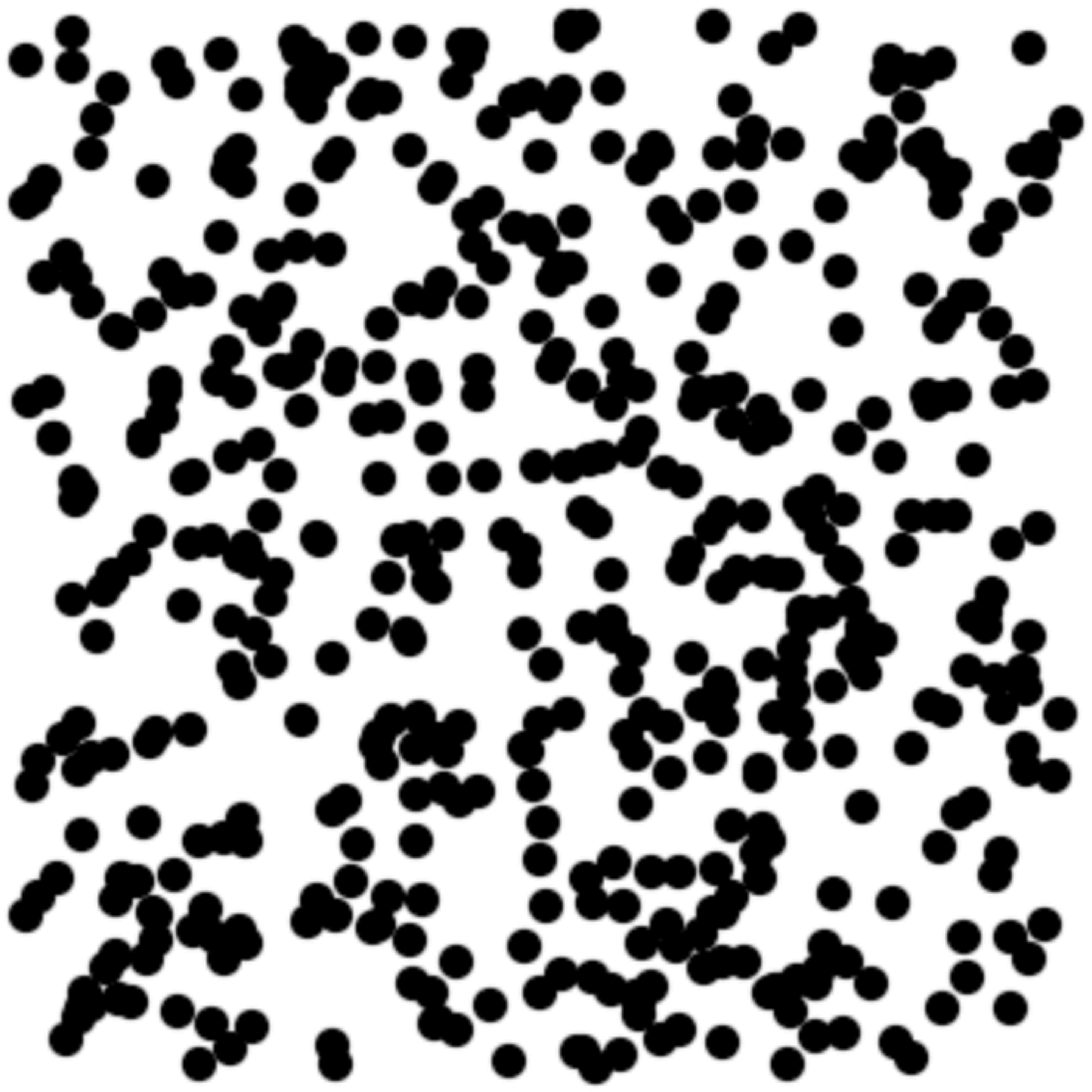
Validation — the null distribution

- How are proteins distributed that
  - Don’t interact with each other
  - Or their surrounding
- To find out, you blindly throw darts at a board
- The chance of a dart landing is the same, no matter where on the board



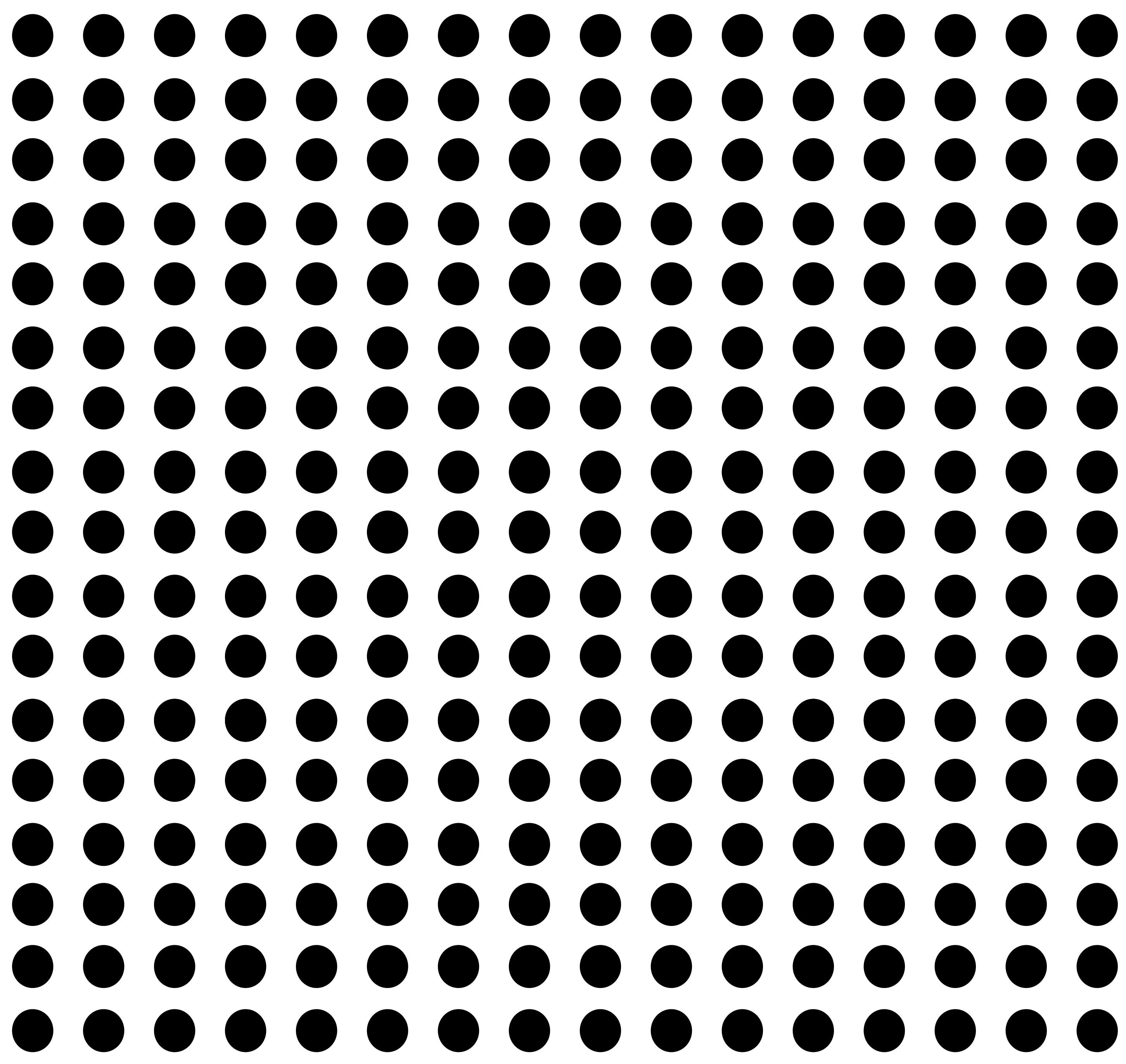
Validation — the null distribution

"Uniformly distributed"



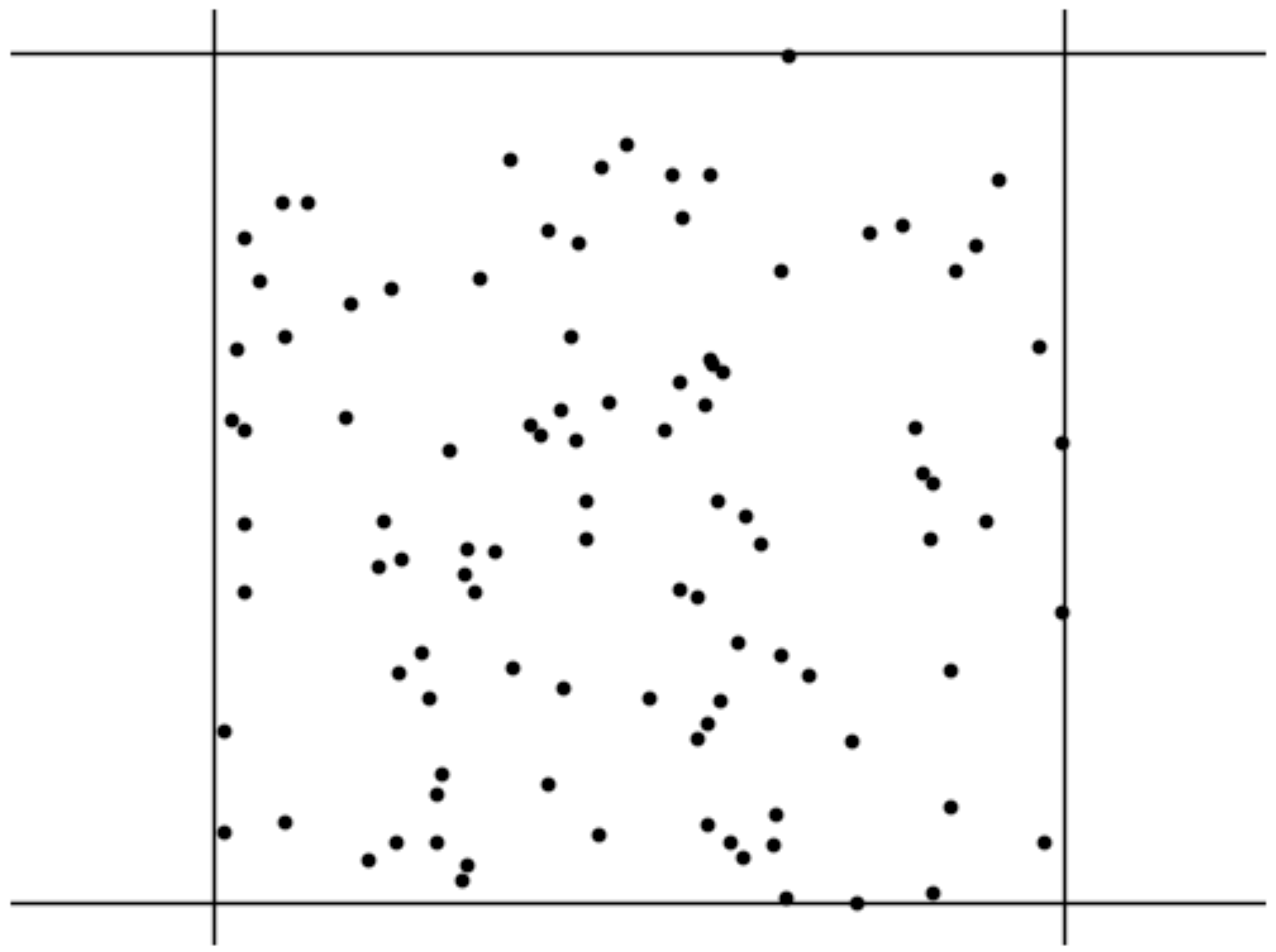
≠

"Uniformly spaced"



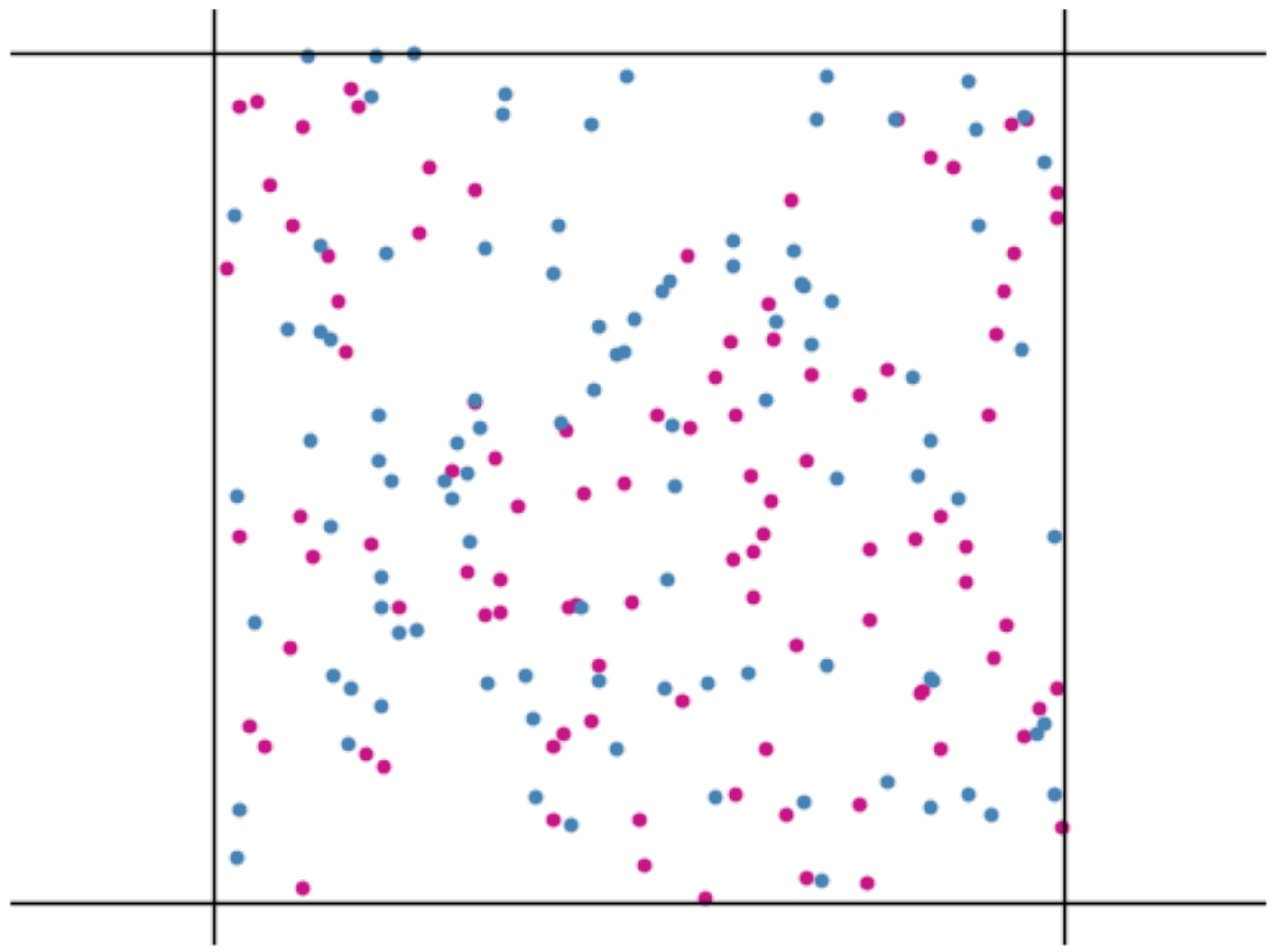
Validation — the null distribution

- How are proteins distributed that
  - Don't interact with each other
  - Or their surrounding
- To find out, you blindly throw darts at a board
- The chance of a dart landing is the same, no matter where on the board



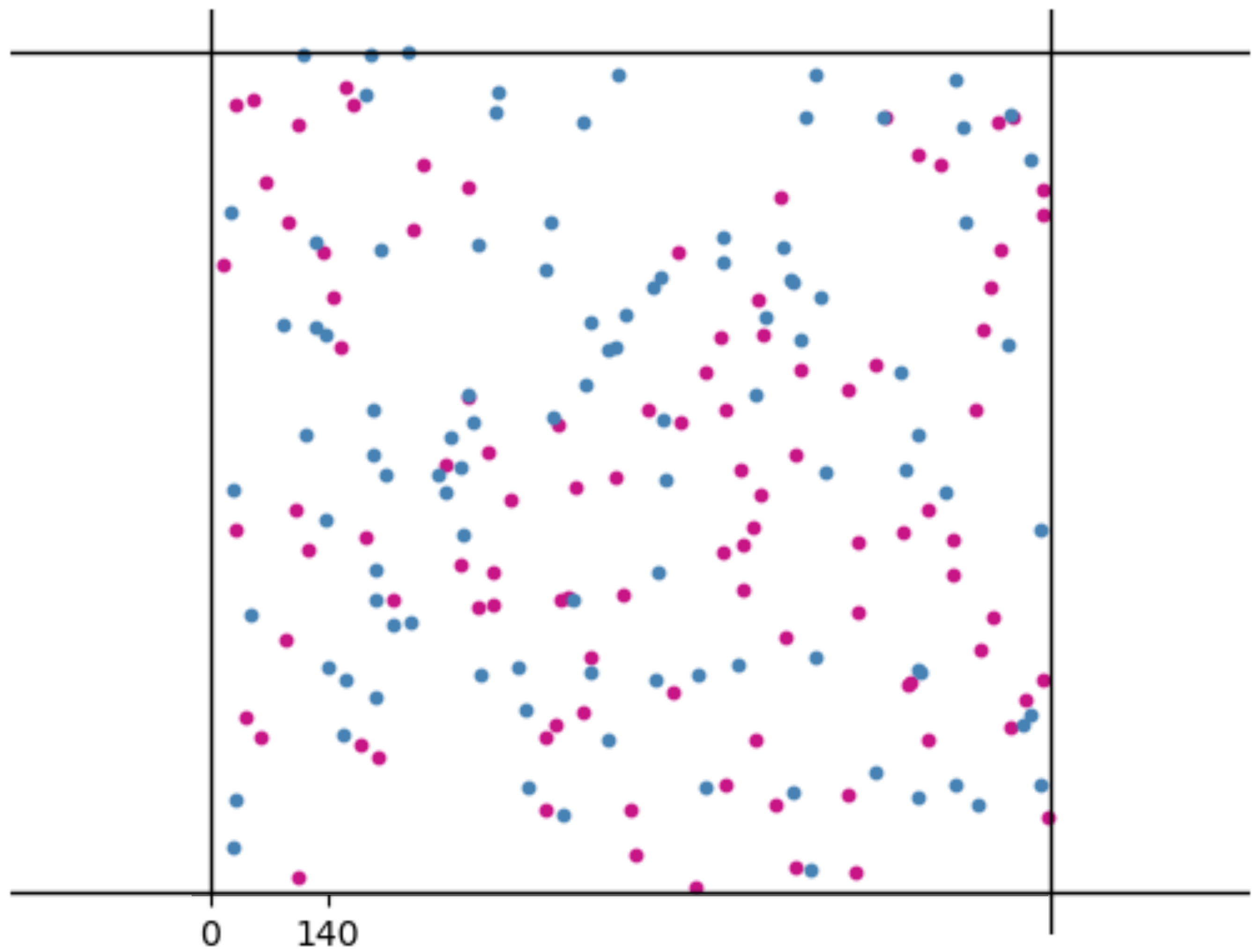
Validation — the null distribution

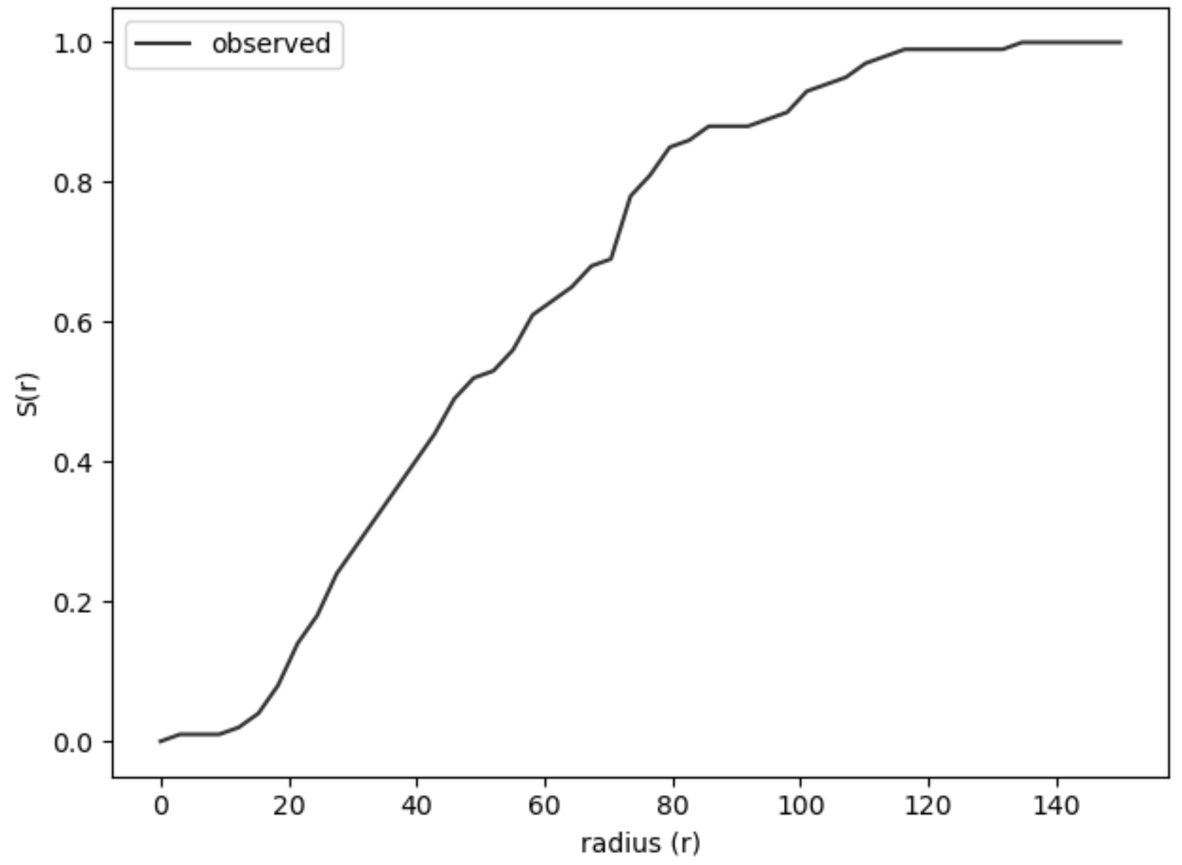
- How are proteins distributed that
  - Don't interact with each other
  - Or their surrounding
- To find out, you blindly throw darts at a board
- The chance of a dart landing is the same, no matter where on the board
- The darts can have multiple colors



Validation — the null distribution

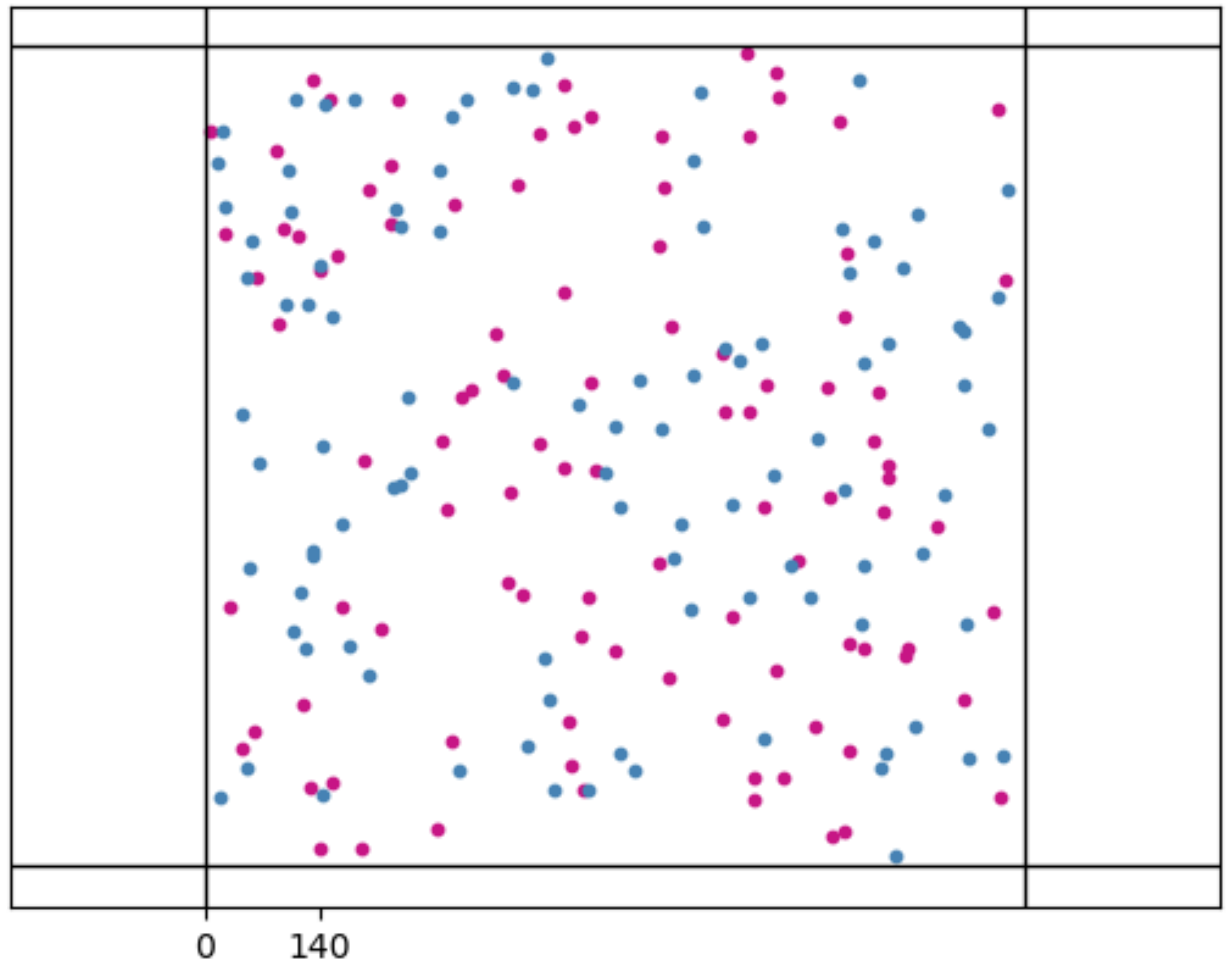
- How are proteins distributed that
  - Don’t interact with each other
  - Or their surrounding
- To find out, you blindly throw darts at a board
- The chance of a dart landing is the same, no matter where on the board
- The darts can have multiple colors

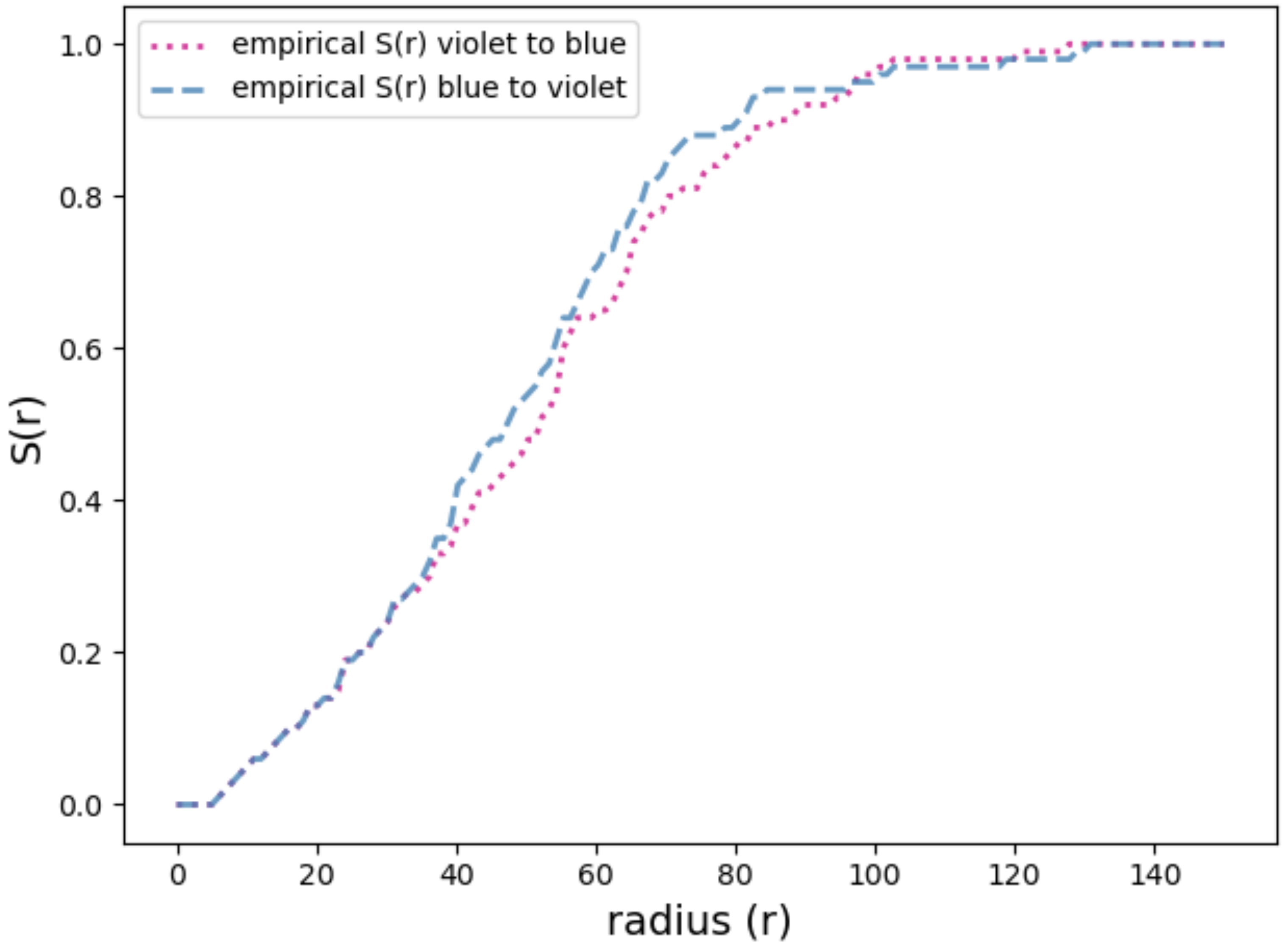




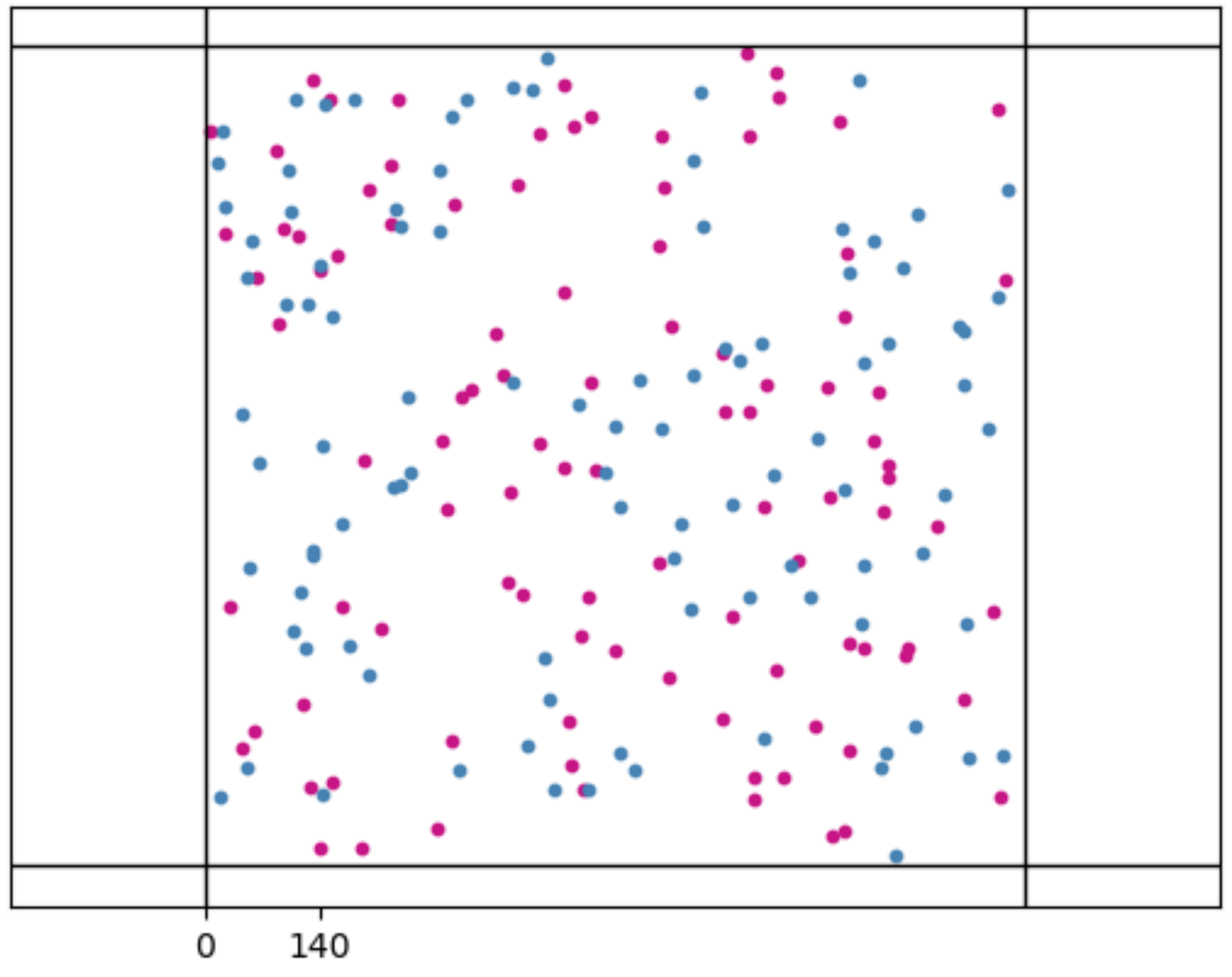
Validation — the null distribution

Empirical null distributions

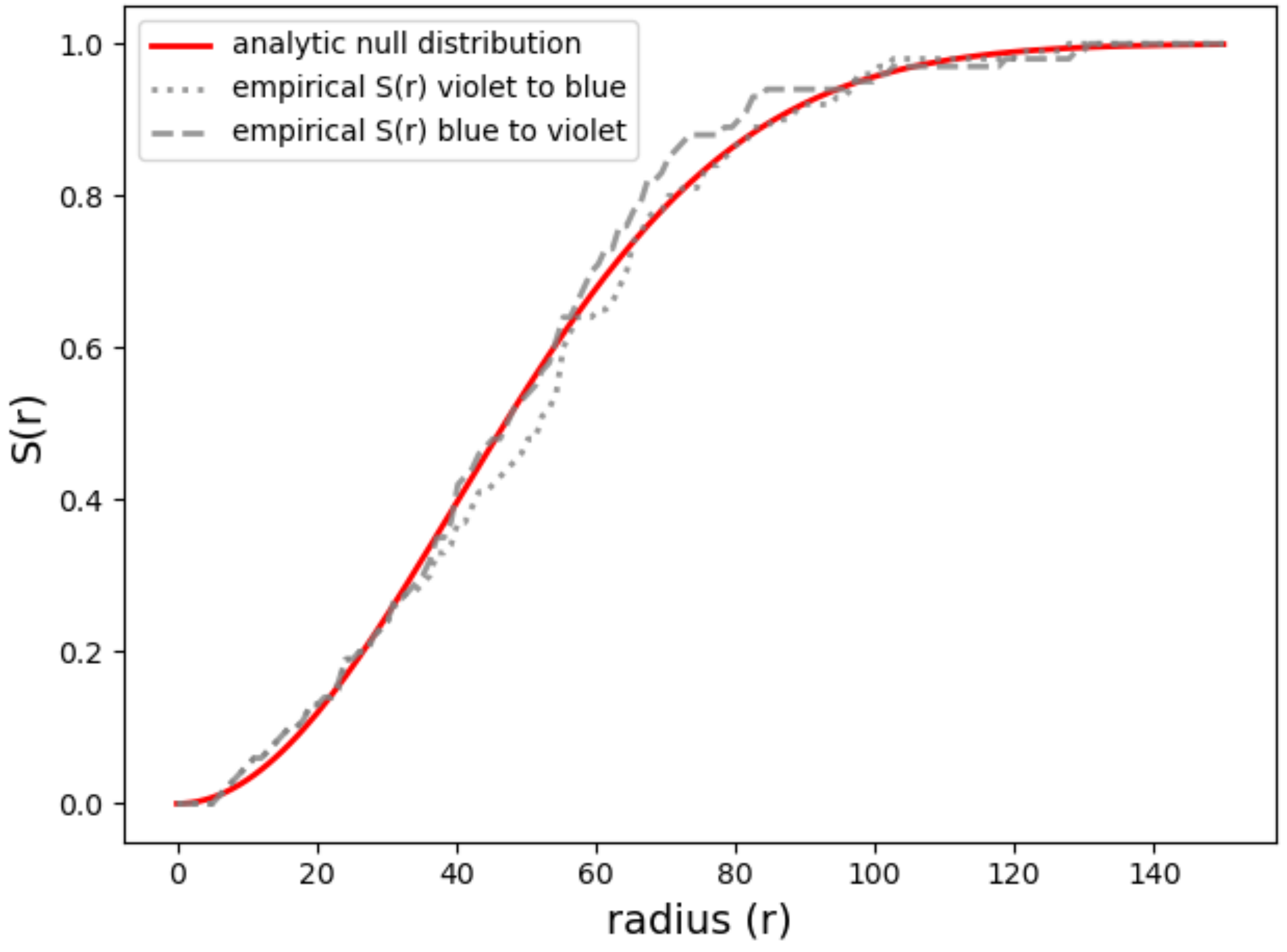




Validation — the null distribution



Analytic null distribution

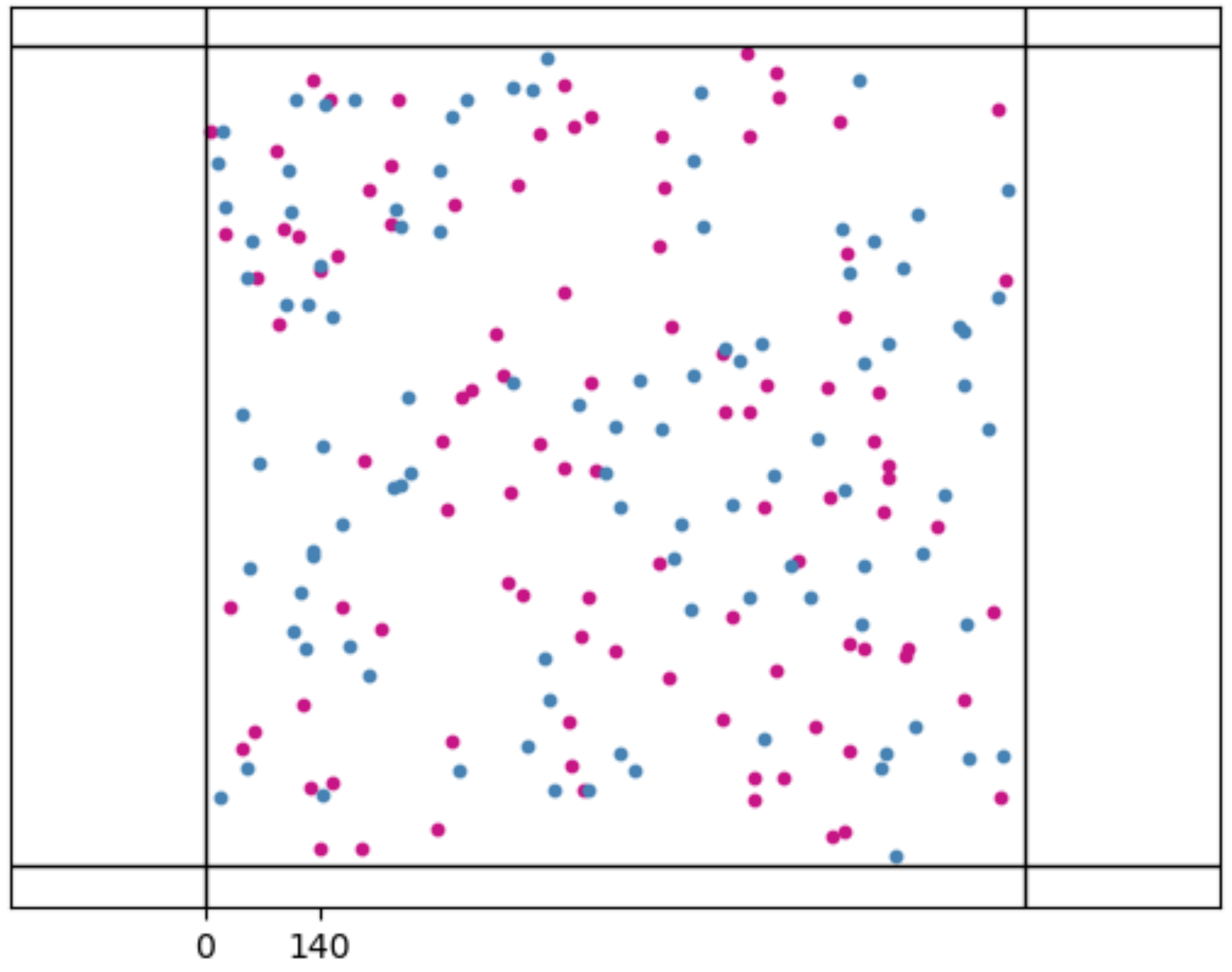


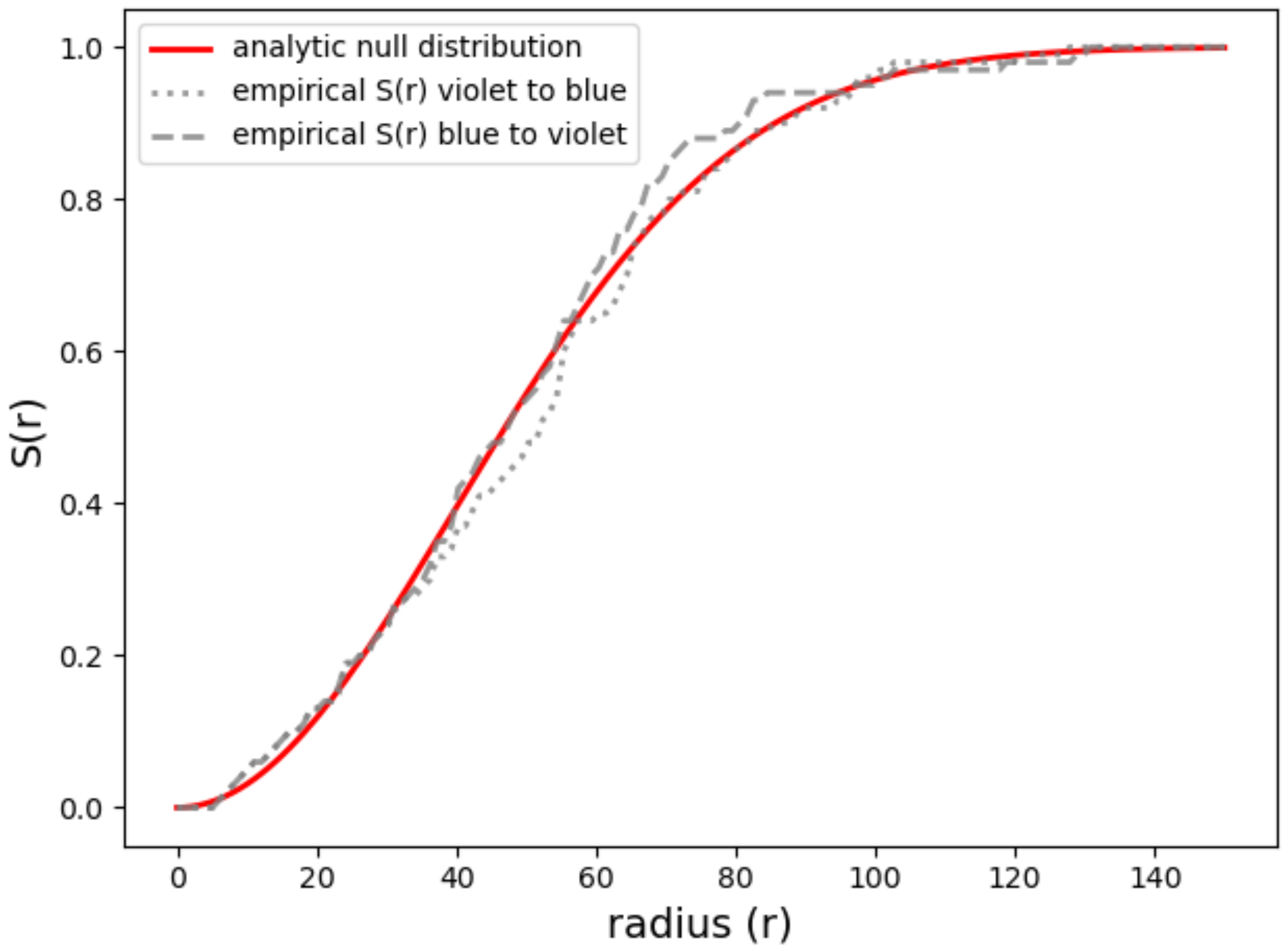
$$S(r) = 1 - e^{-\frac{n_2}{|\Omega|}\pi r^2}$$

Density of points n_2

Validation — the null distribution

Analytic null distribution

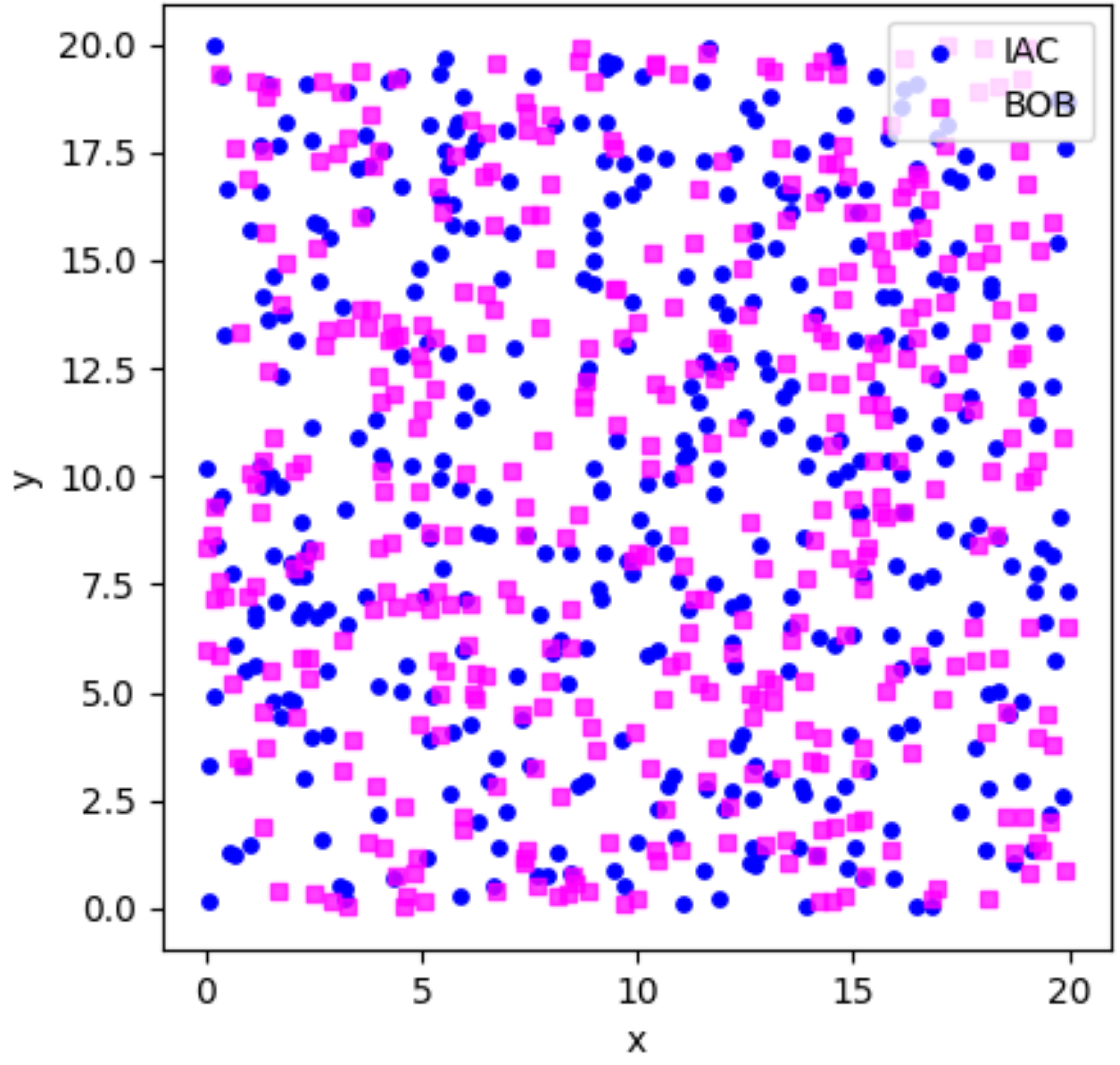




$$S(r) = 1 - e^{-\frac{n_2}{|\Omega|}\pi r^2}$$

Area of a circle with radius r

Validation — the null distribution

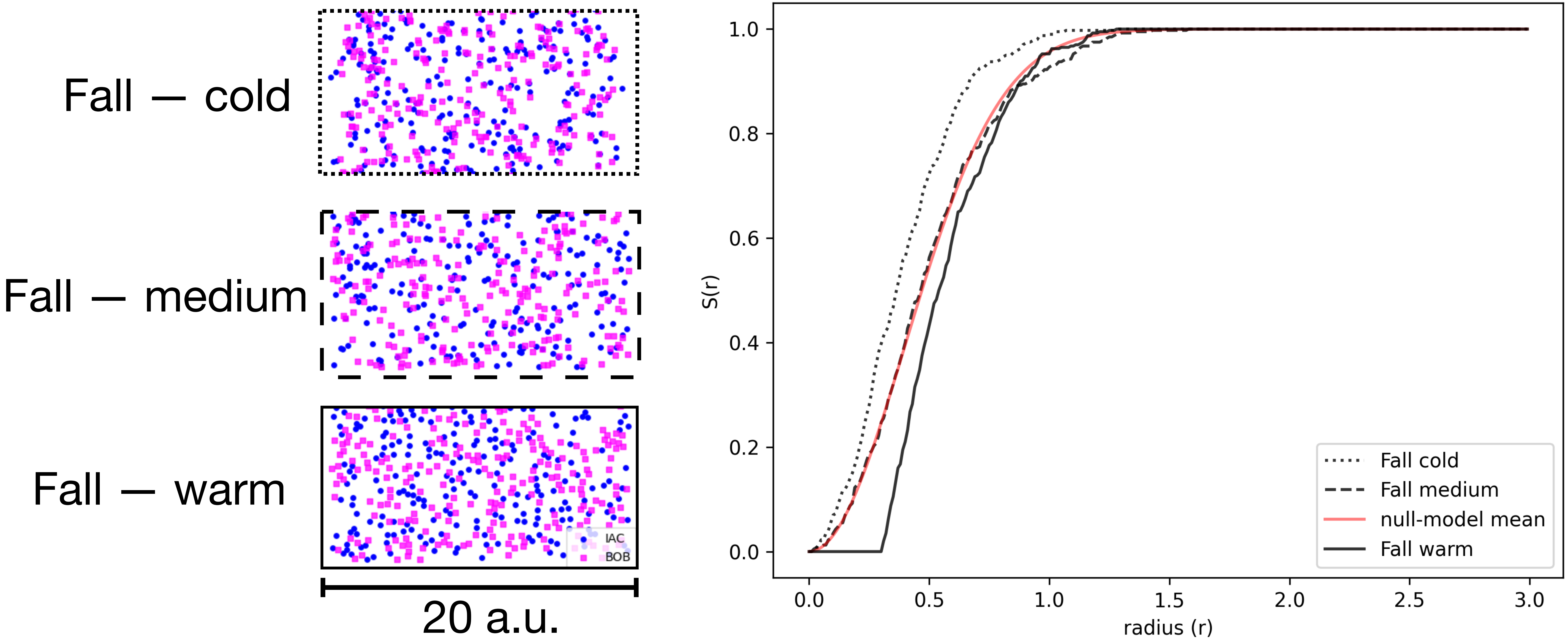


$$S(r) = 1 - e^{-\frac{n_2}{|\Omega|}\pi r^2}$$

Exercise: Find good values for n2 and |Ω|

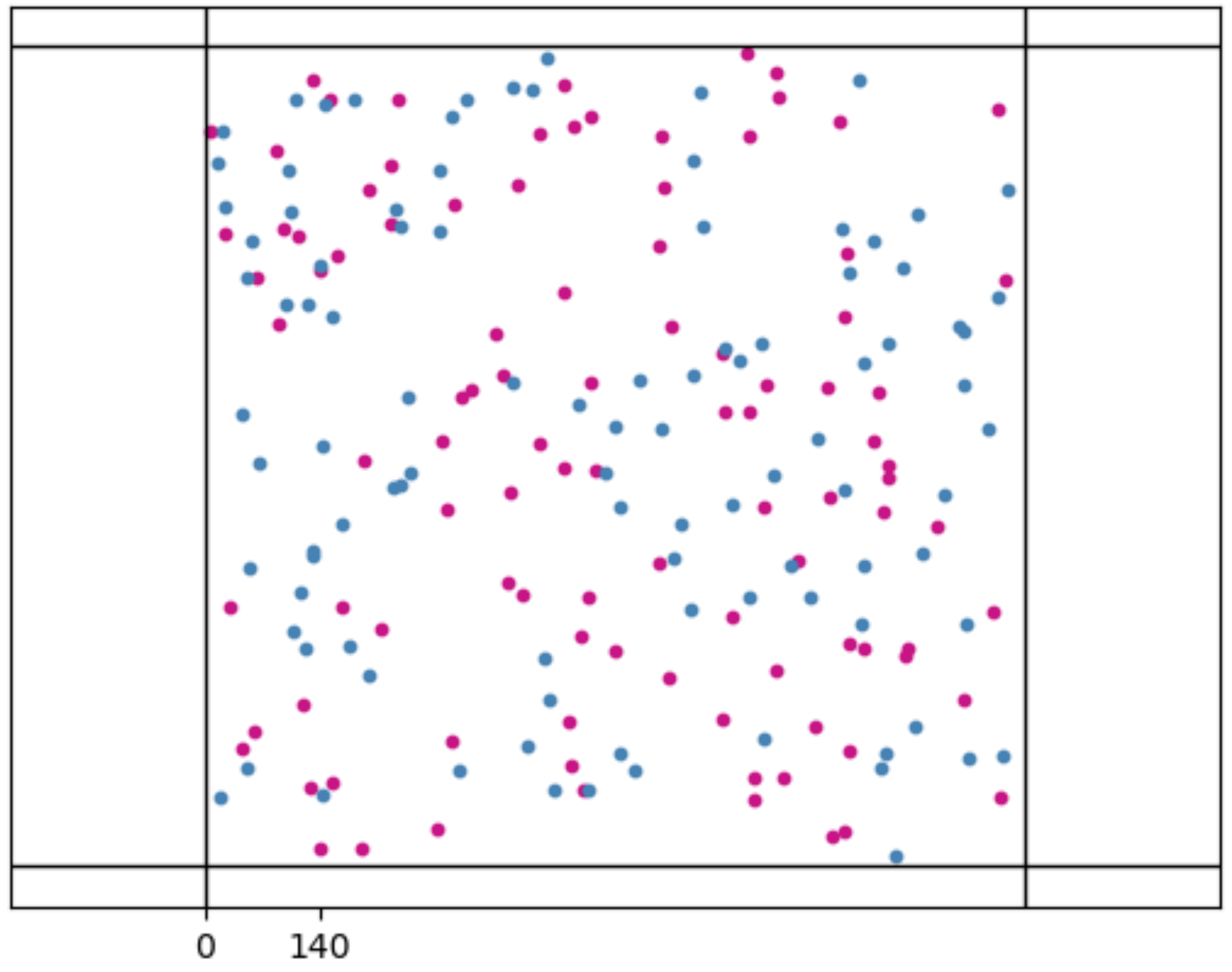
```
n = 1345345  # number of points in the dataset
area = 4056780  # area of the FOV

nulldist = getnulldist(
    n=n, area=area, radii=radii
)  # These are not good values for n and area. Change them!
```

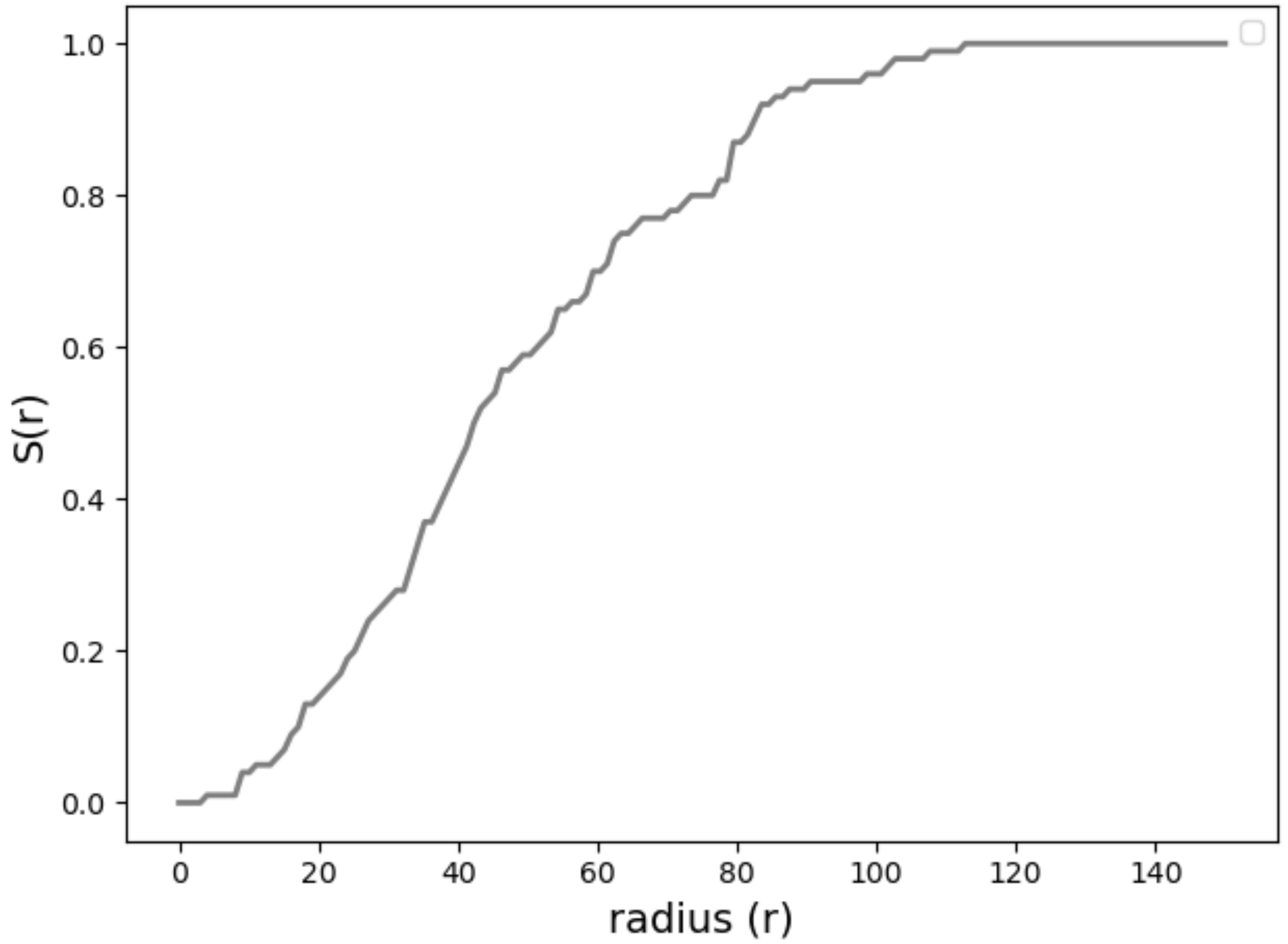
Results — the null distribution




In [1]: from microscopy import analysis
In [2]: img = analysis.imread("bobiac.tif")
In [3]: analysis.imshow(img)
In [4]:

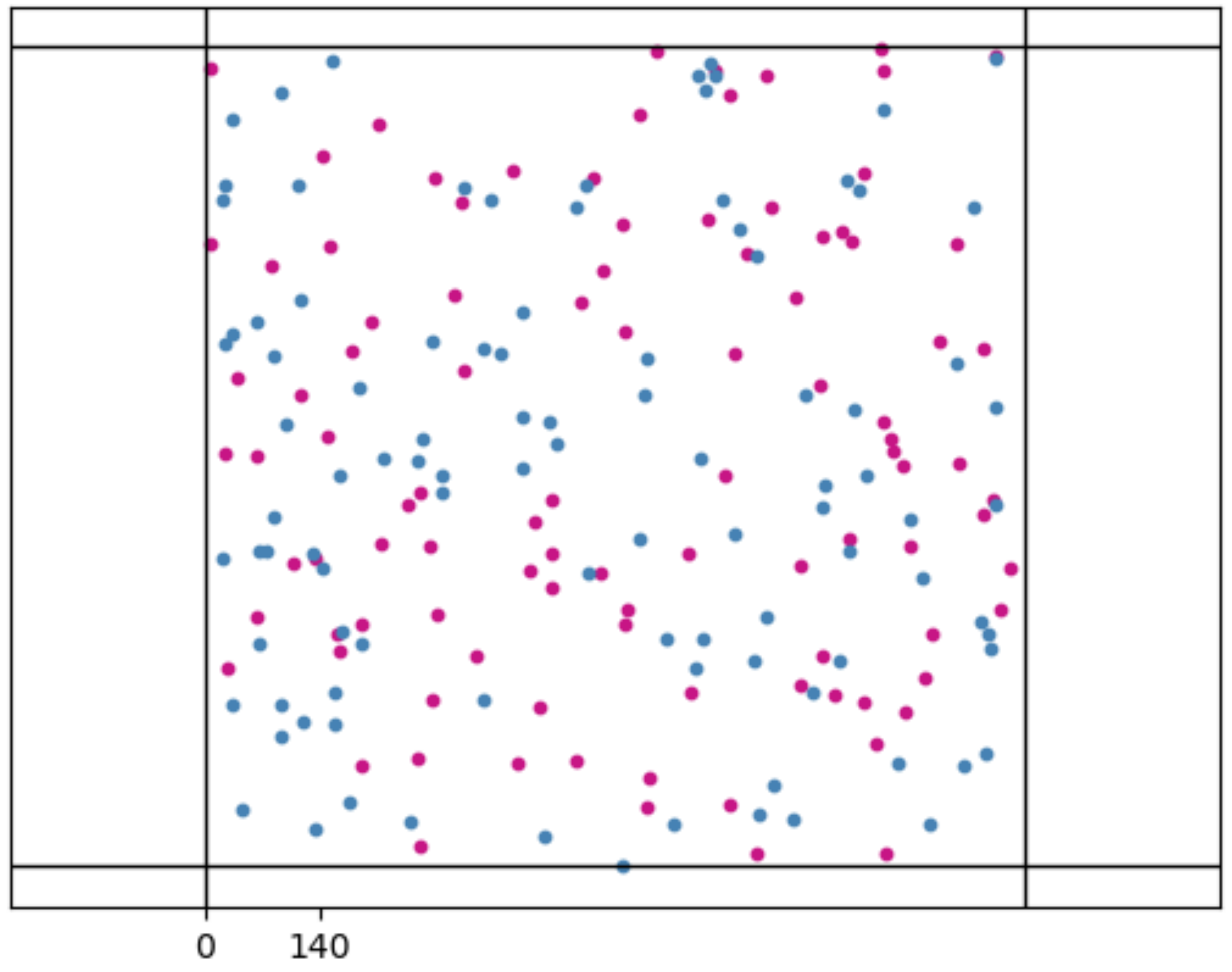
Monte-Carlo-based significance testing

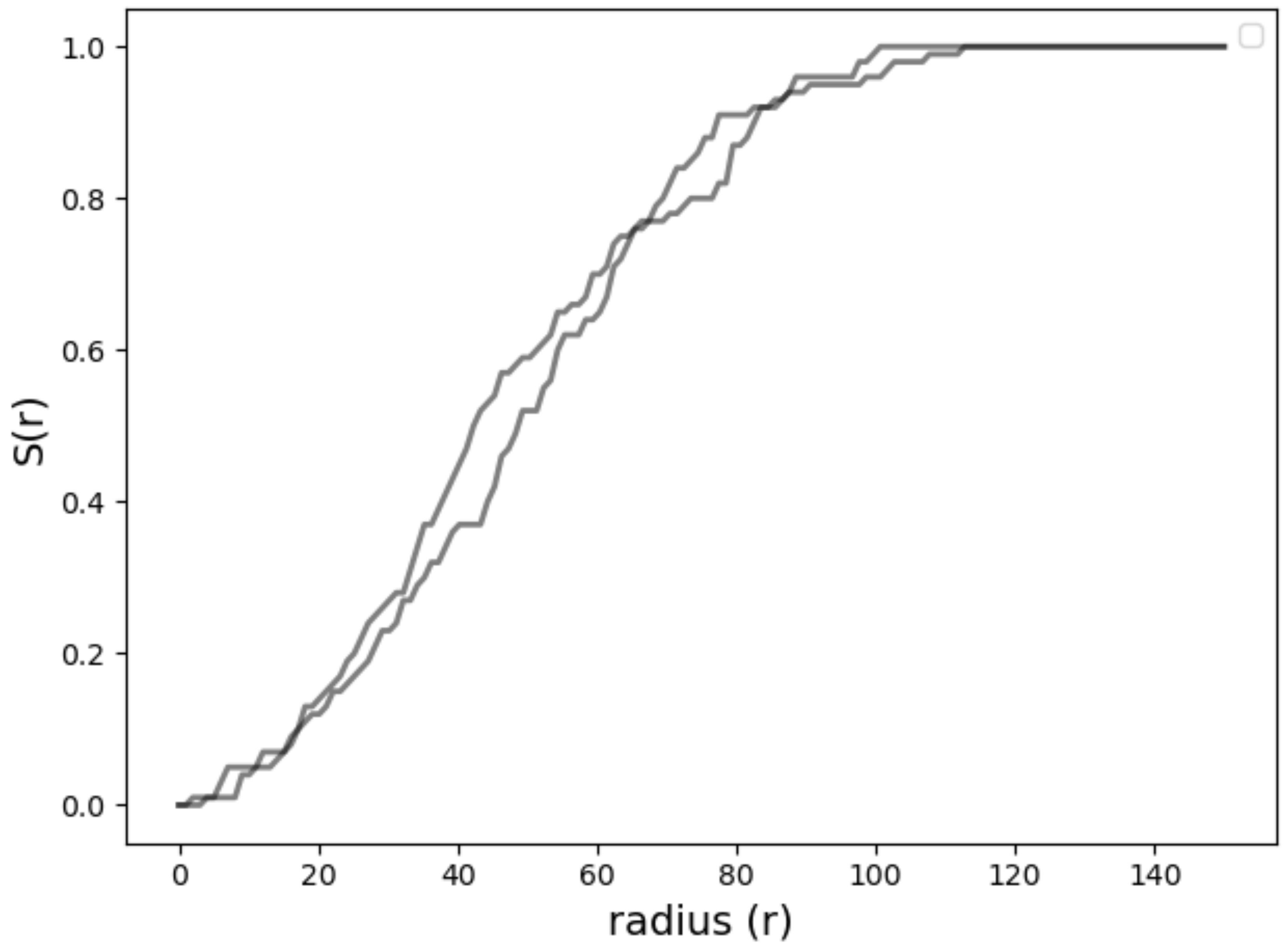




1

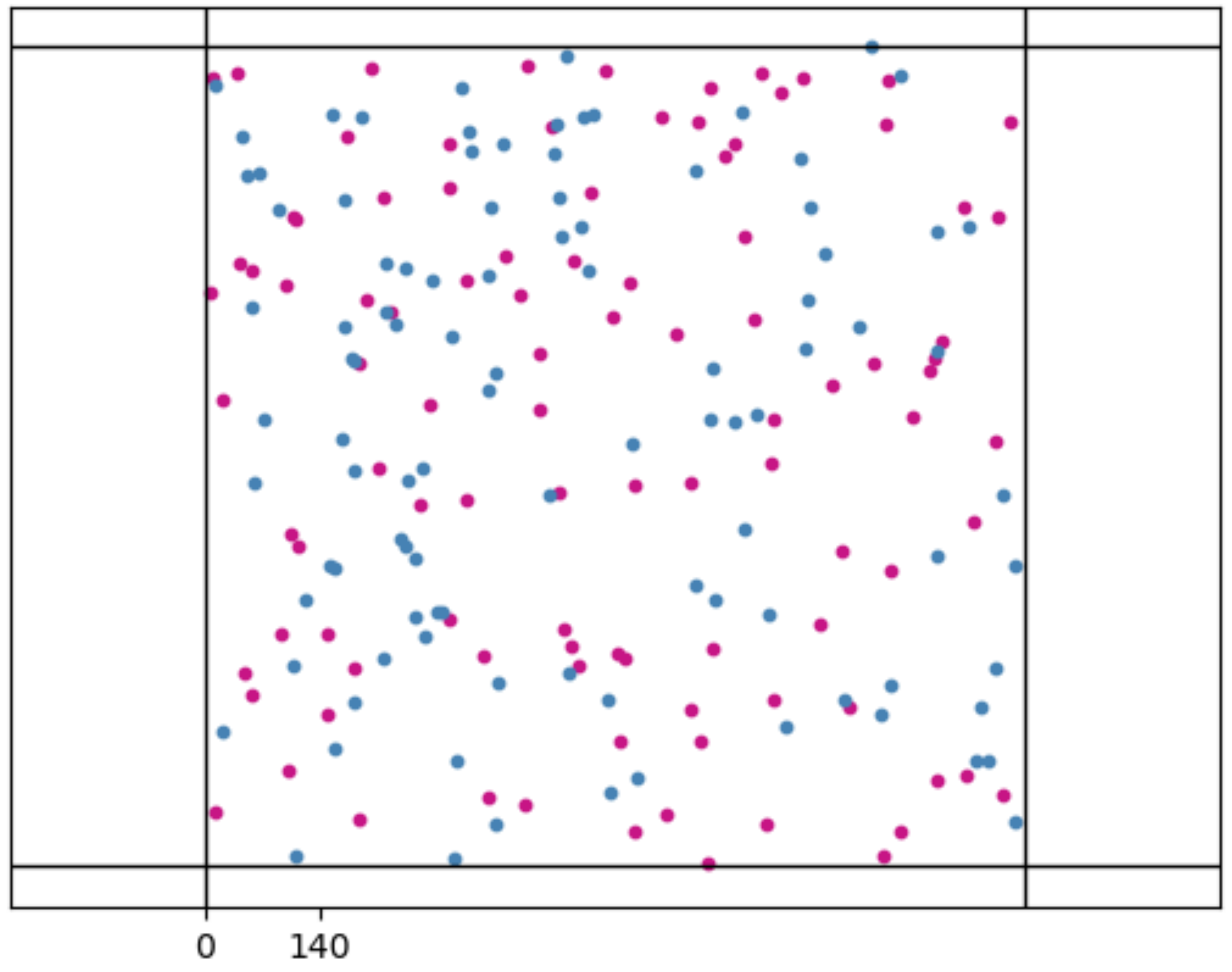
Monte-Carlo-based significance testing

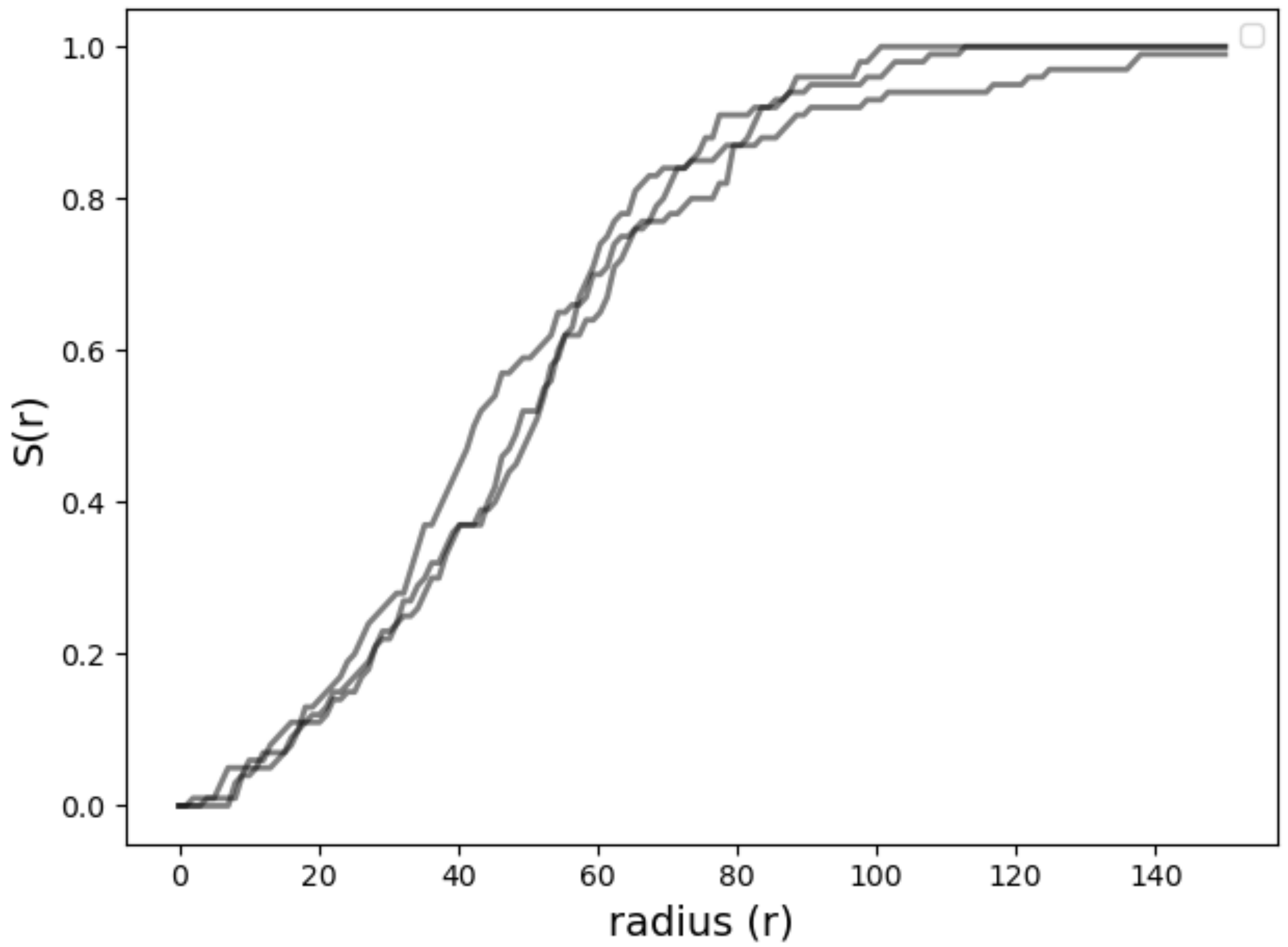




2

Monte-Carlo-based significance testing

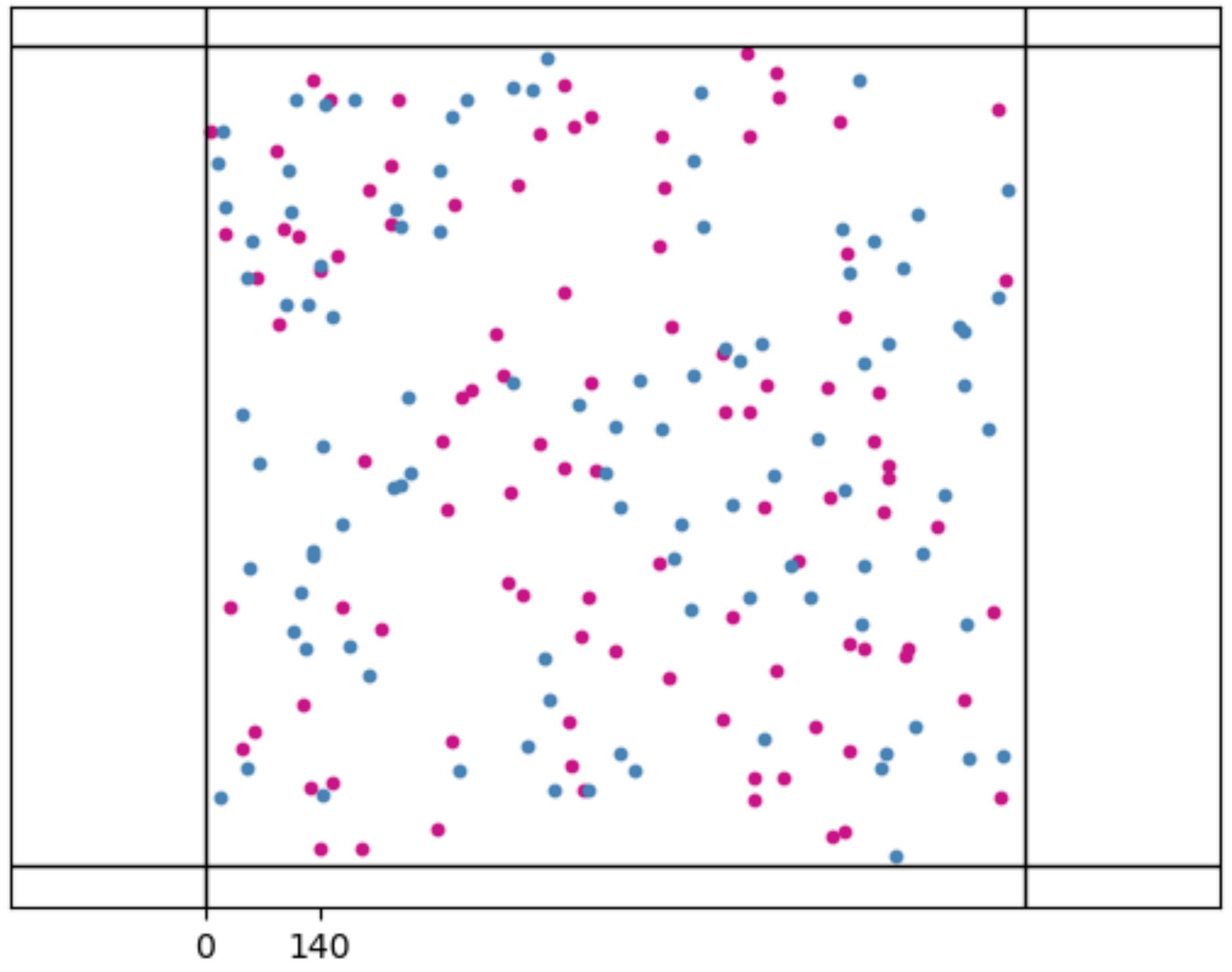


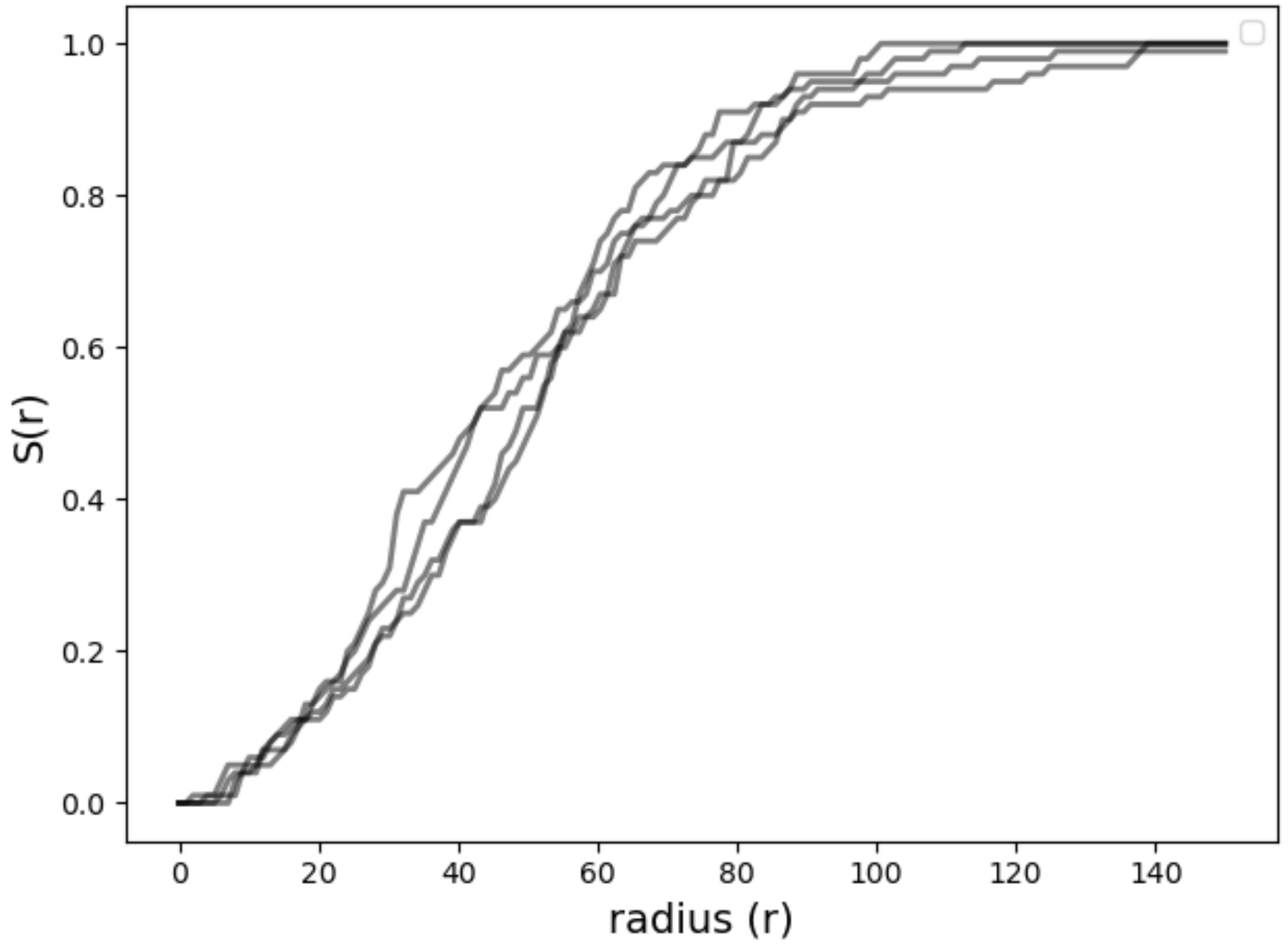


3

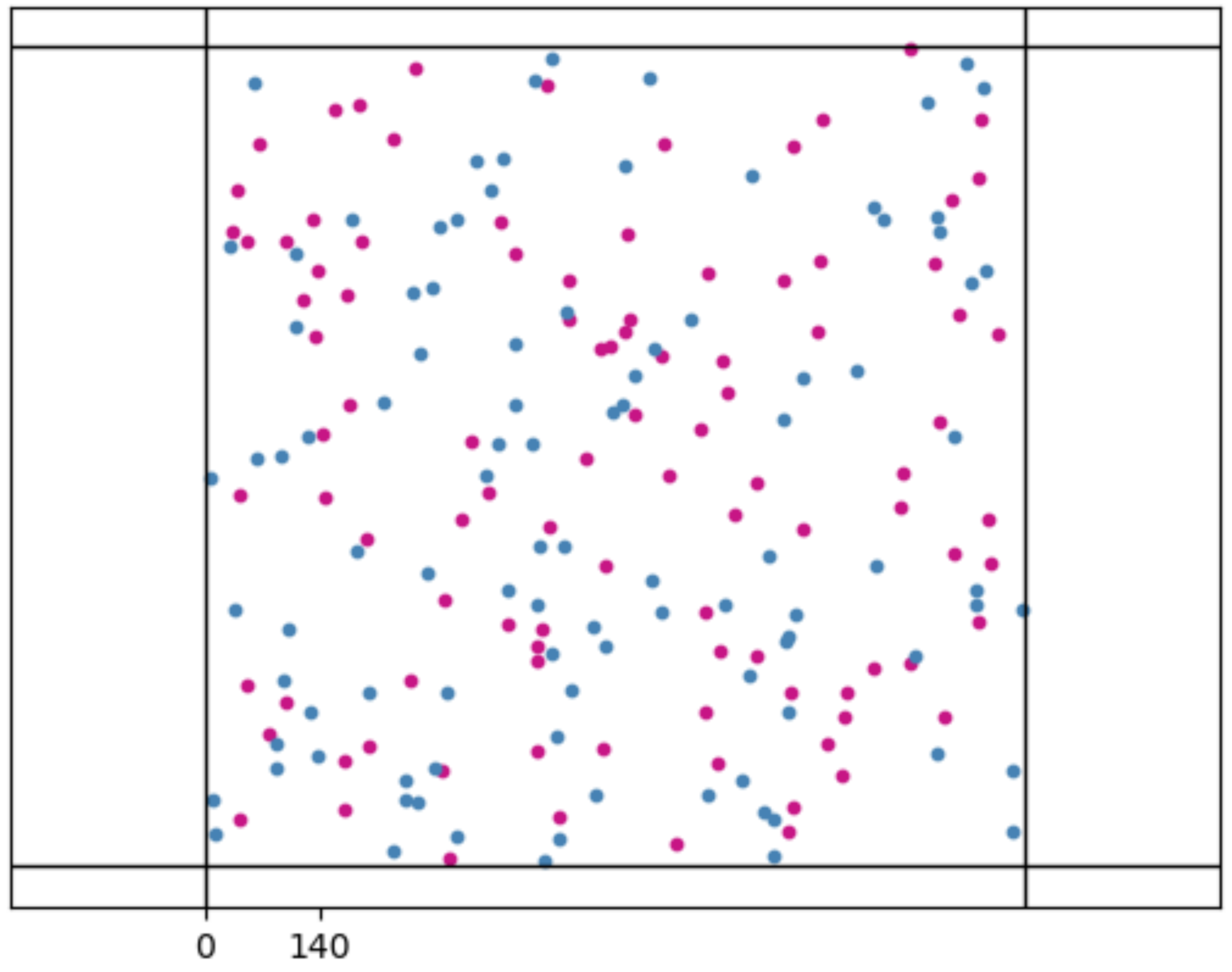
Monte-Carlo-based significance testing

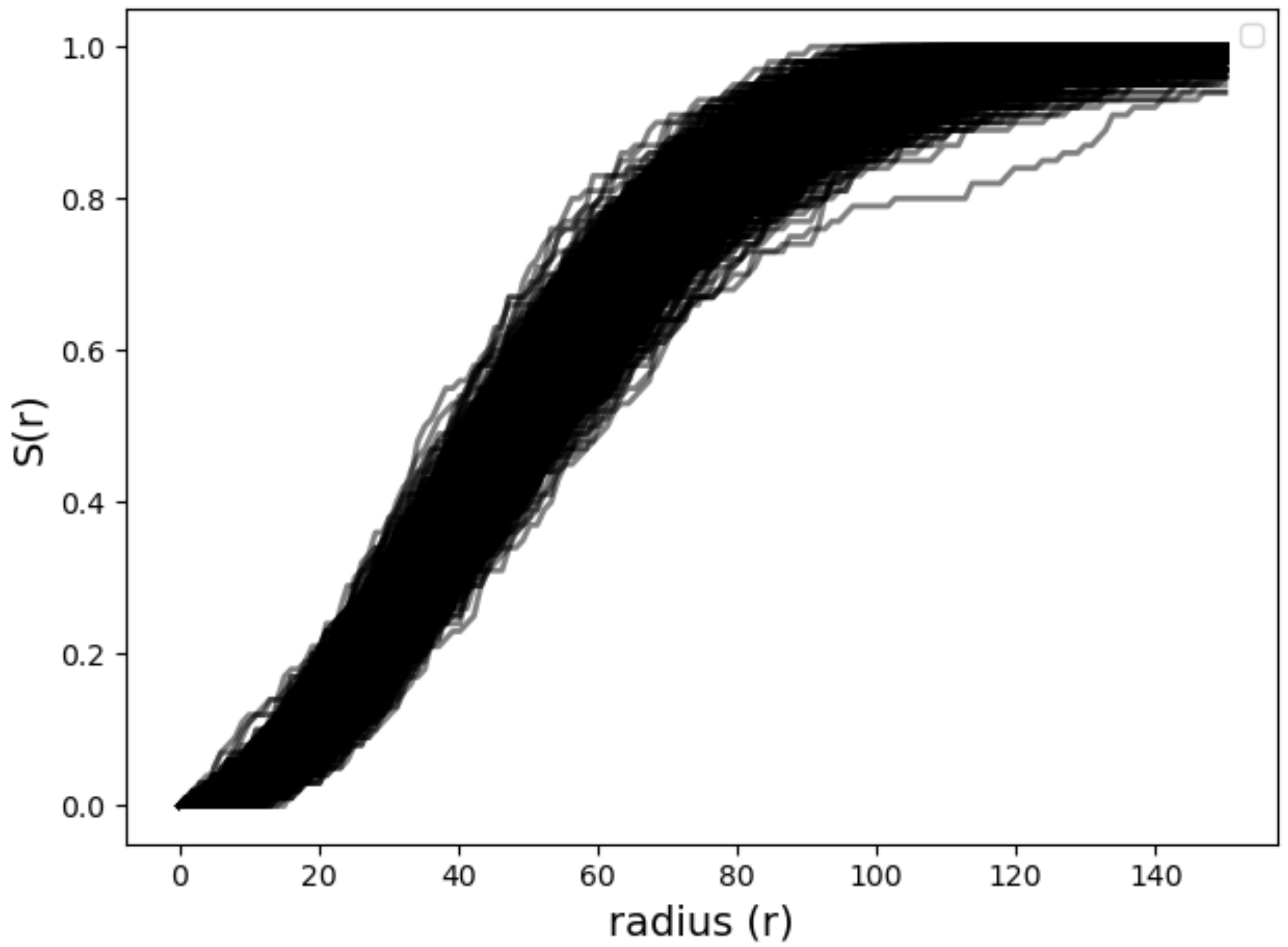
4





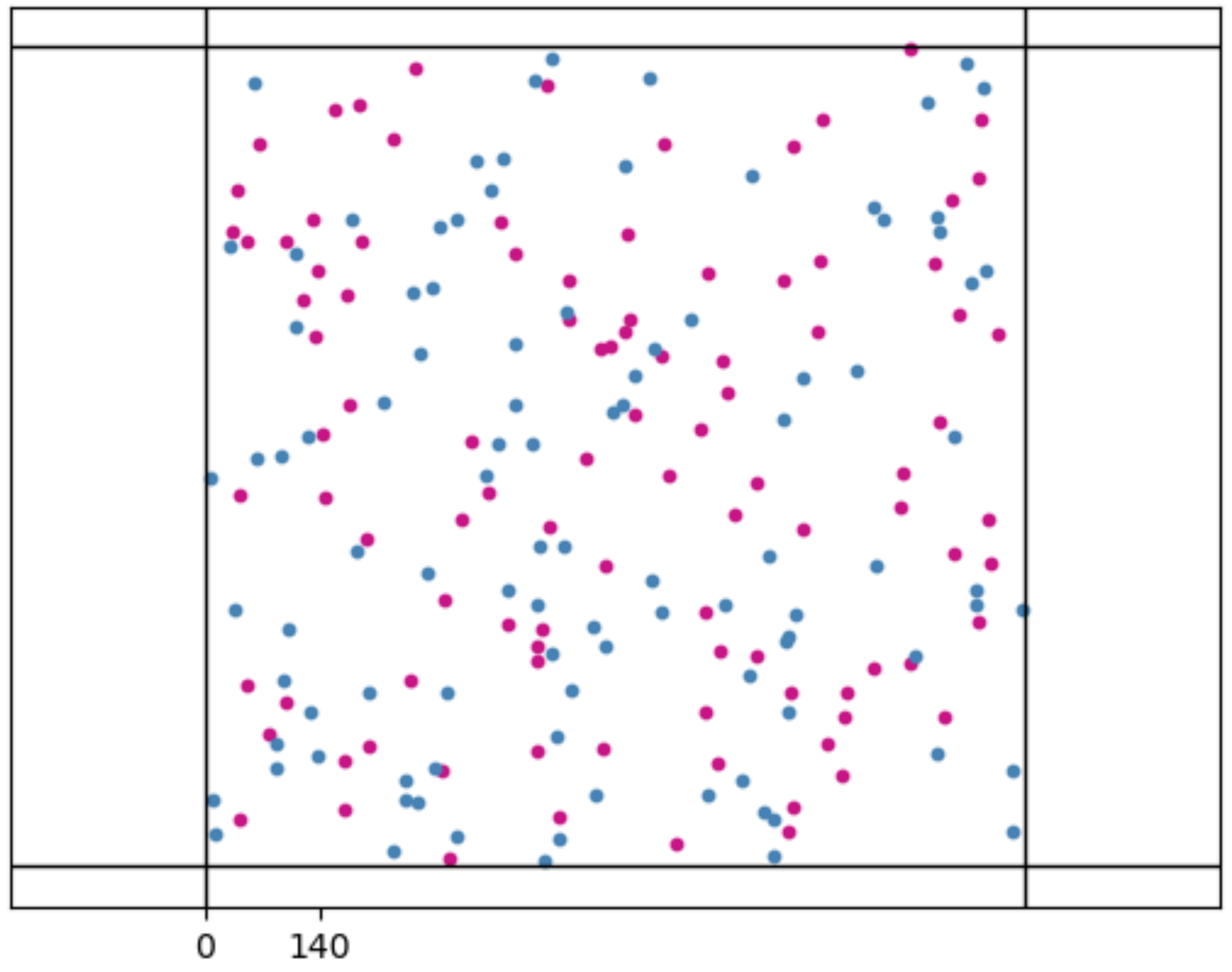
Monte-Carlo-based significance testing

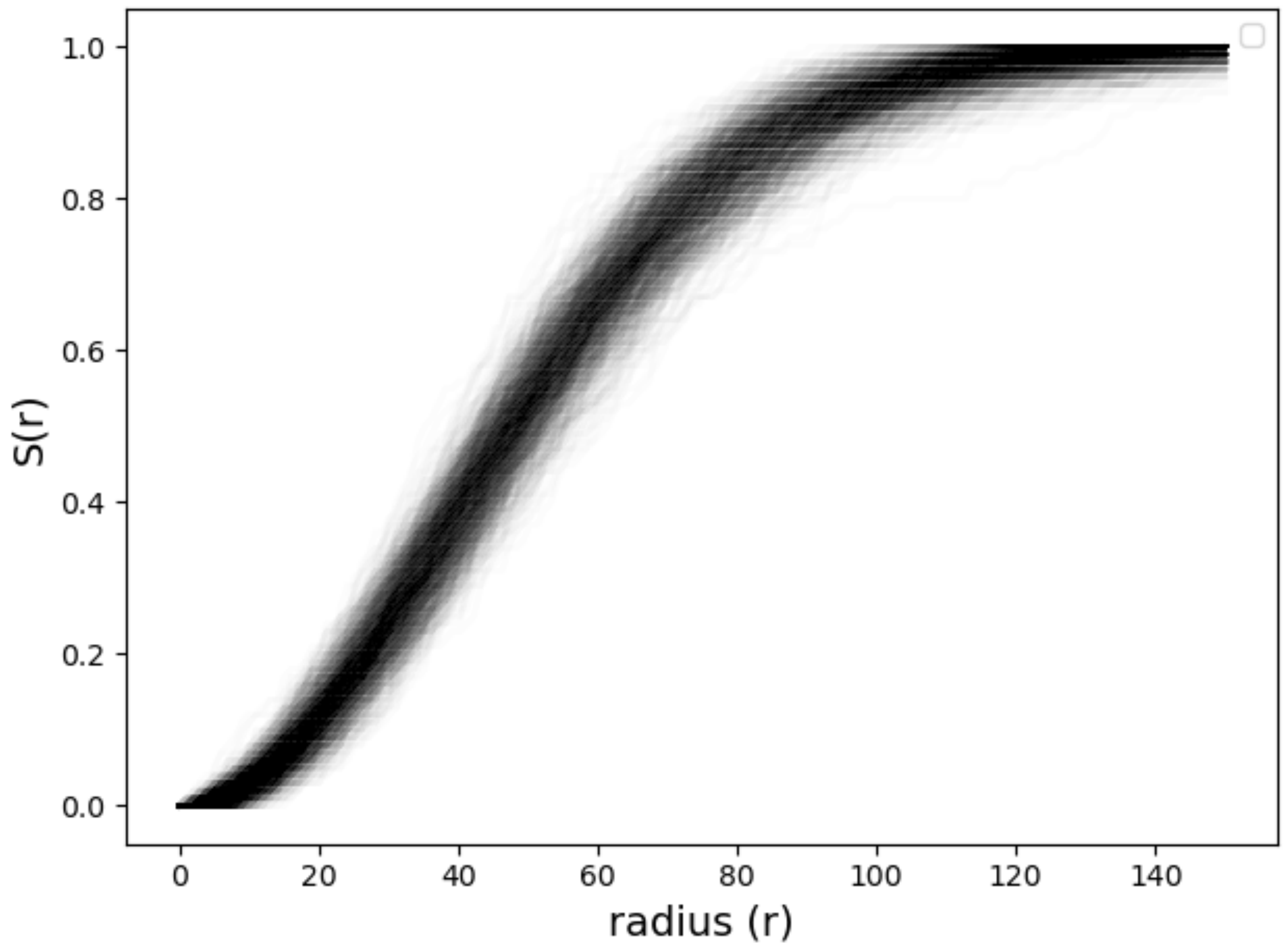




1000

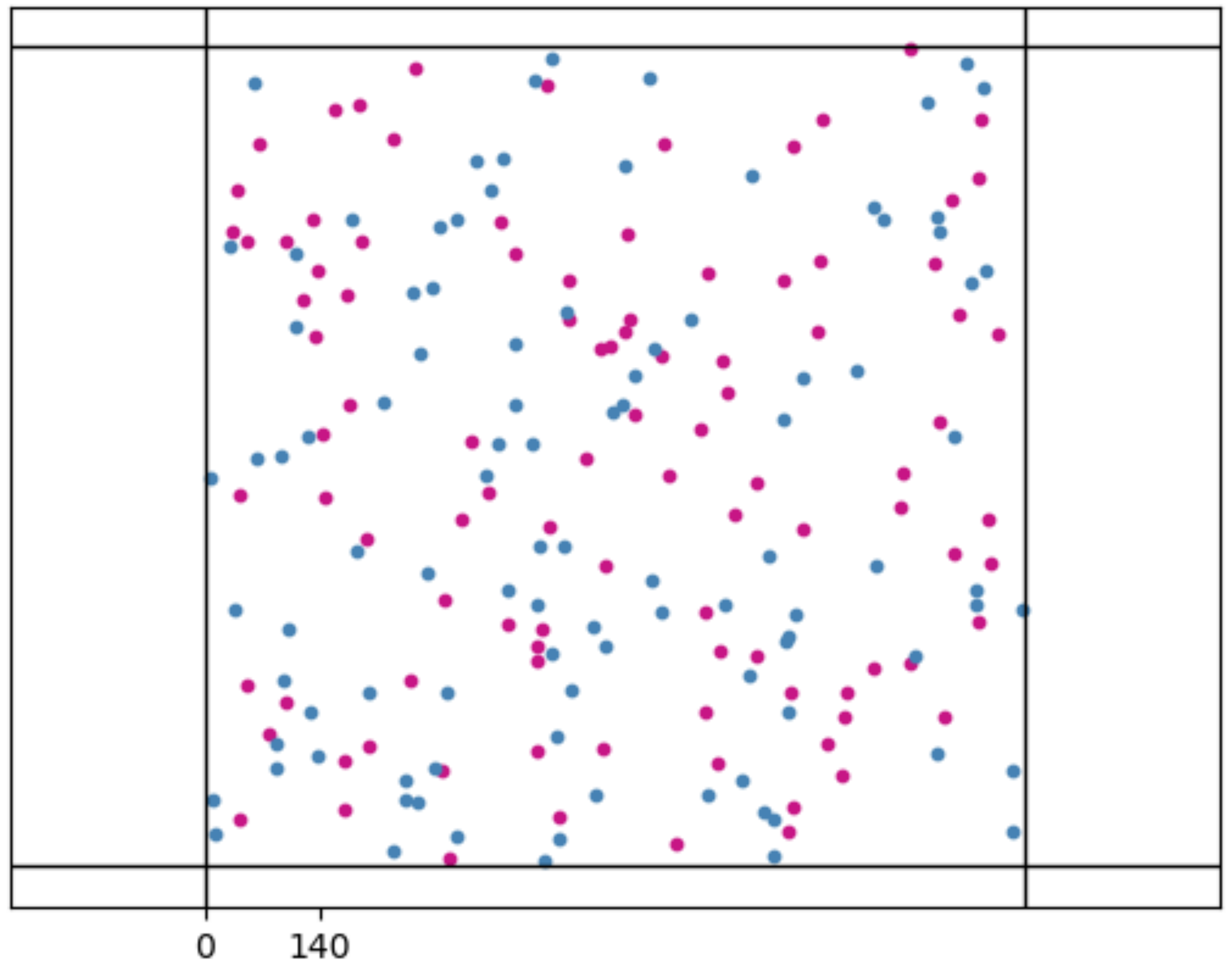
Monte-Carlo-based significance testing

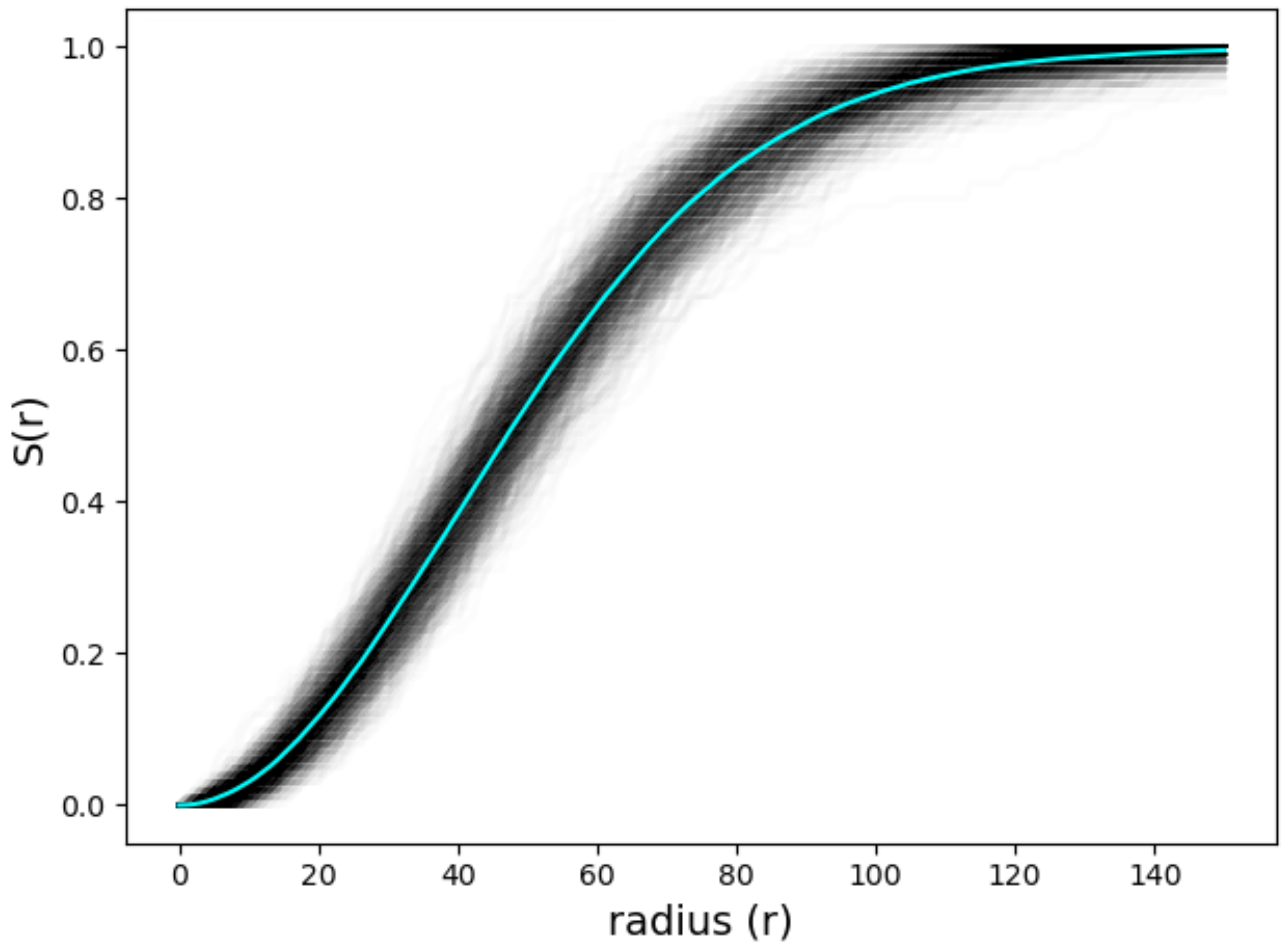




1000

Monte-Carlo-based significance testing

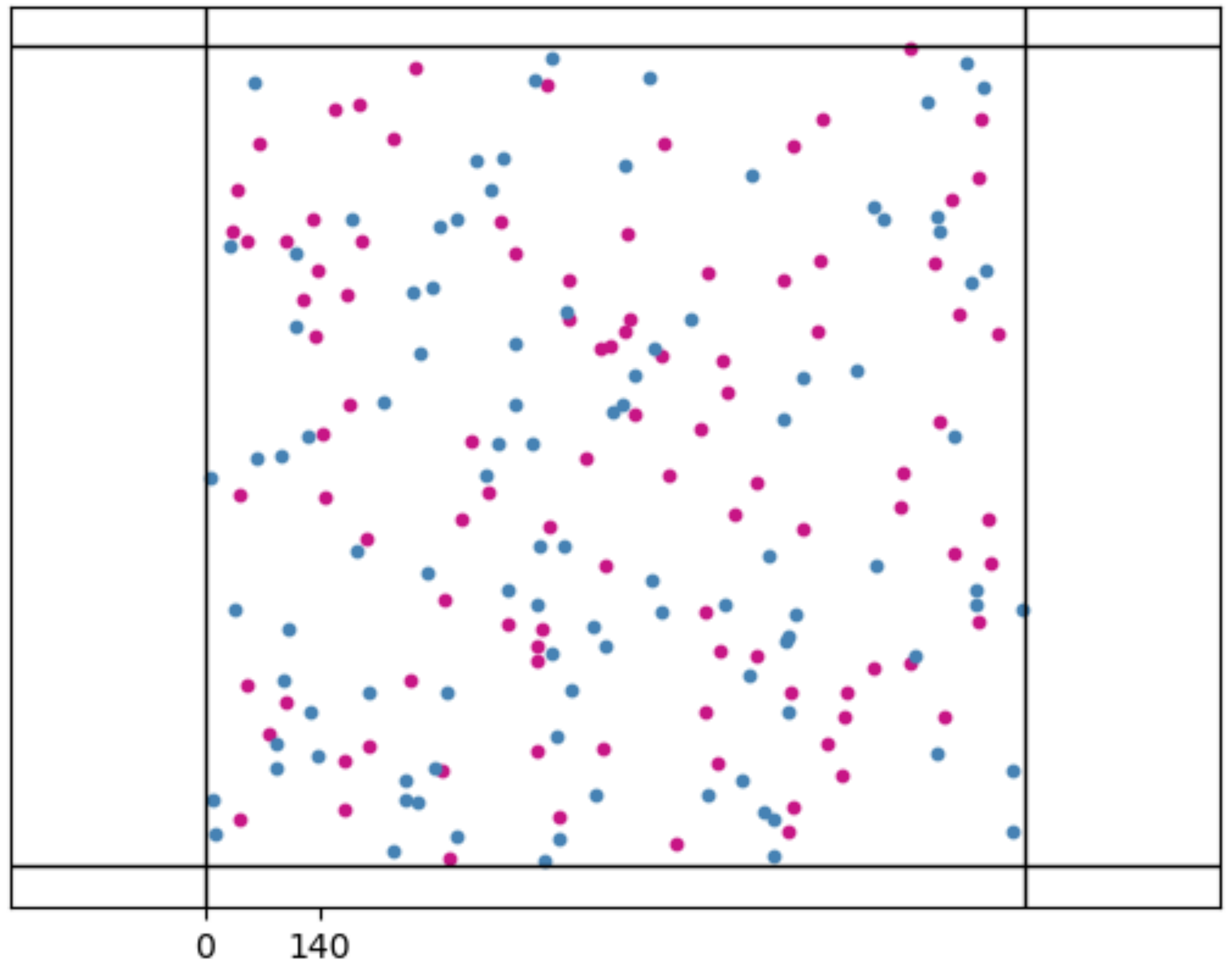


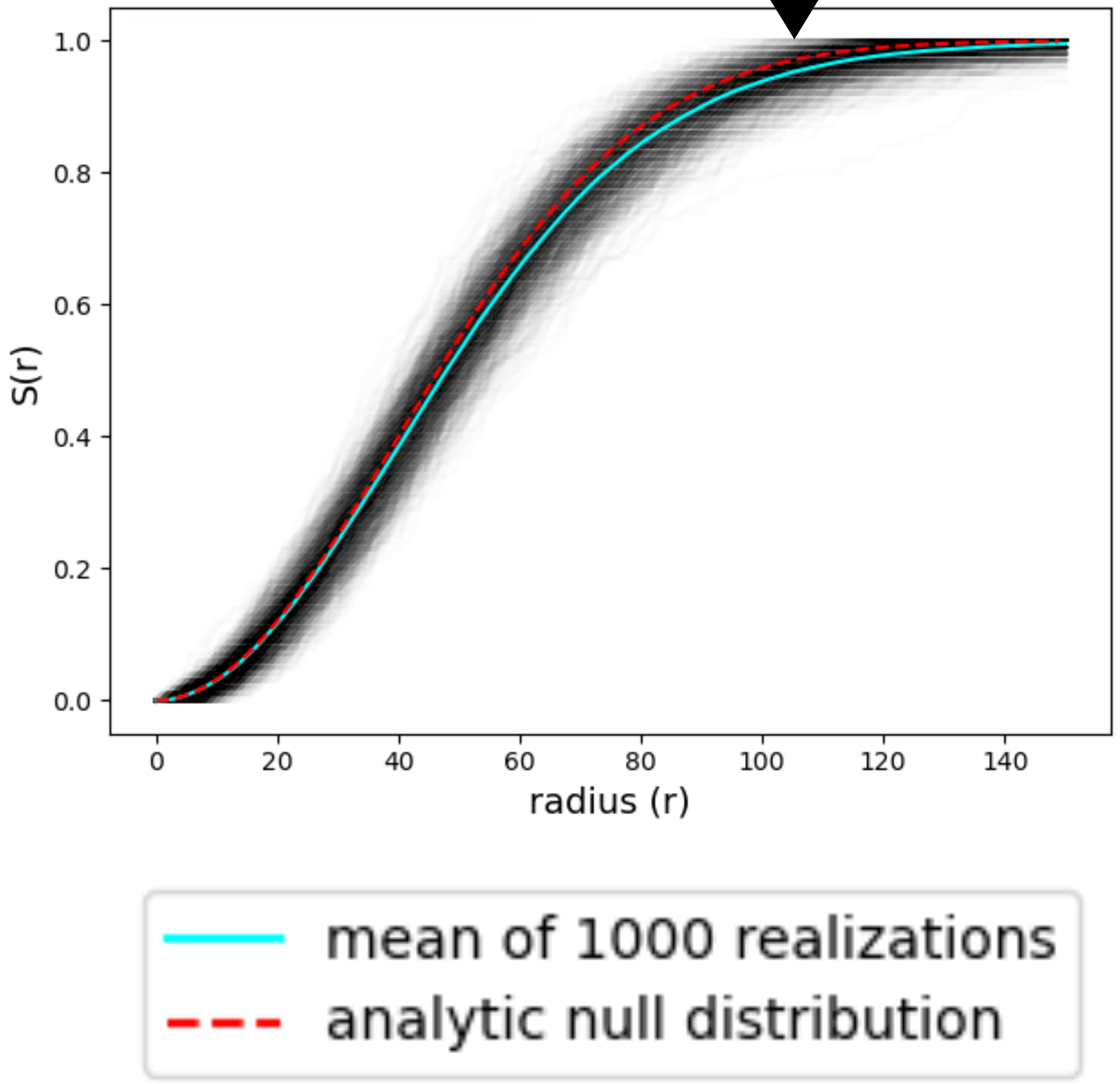


mean of 1000 realizations

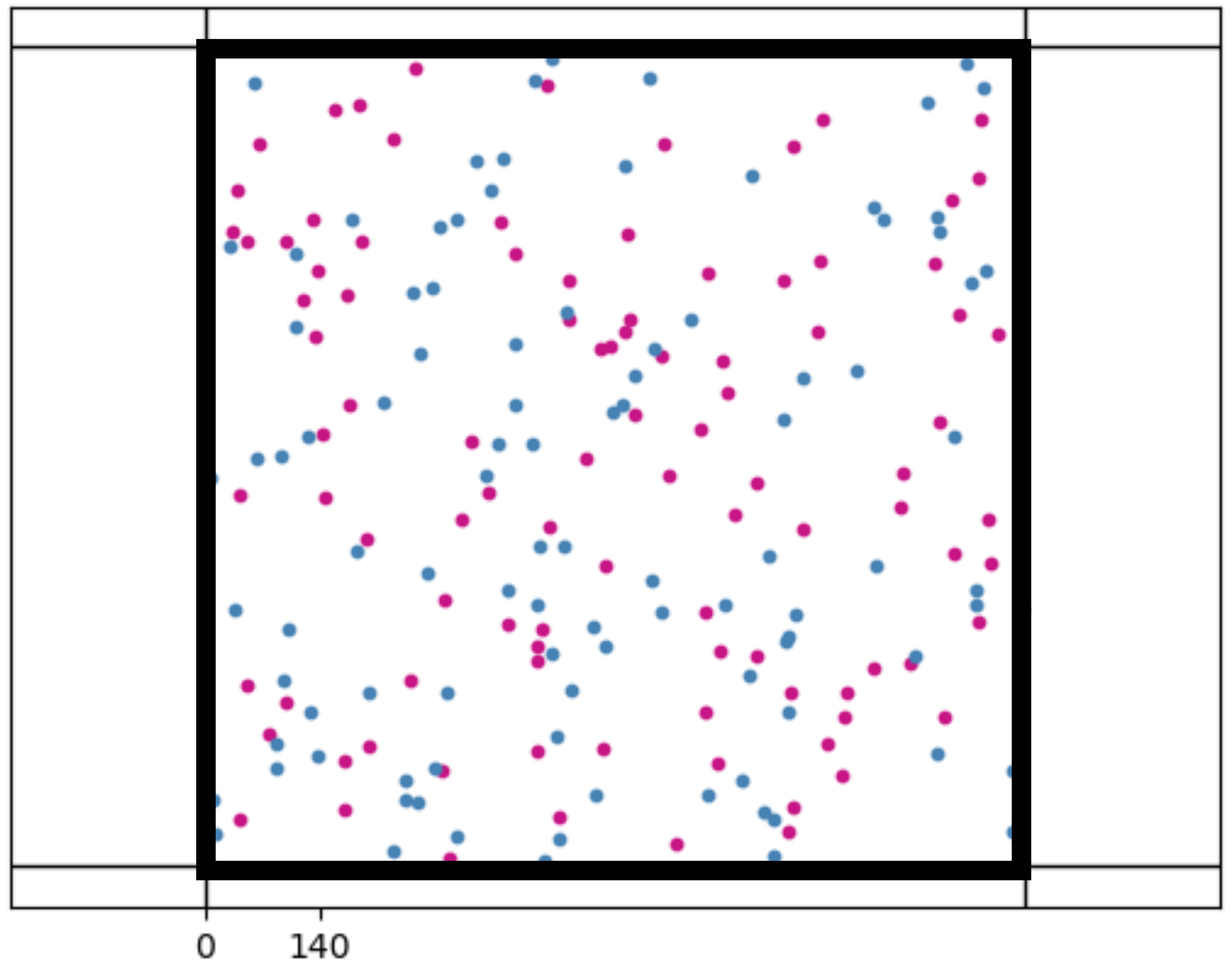
Monte-Carlo-based significance testing

Why?

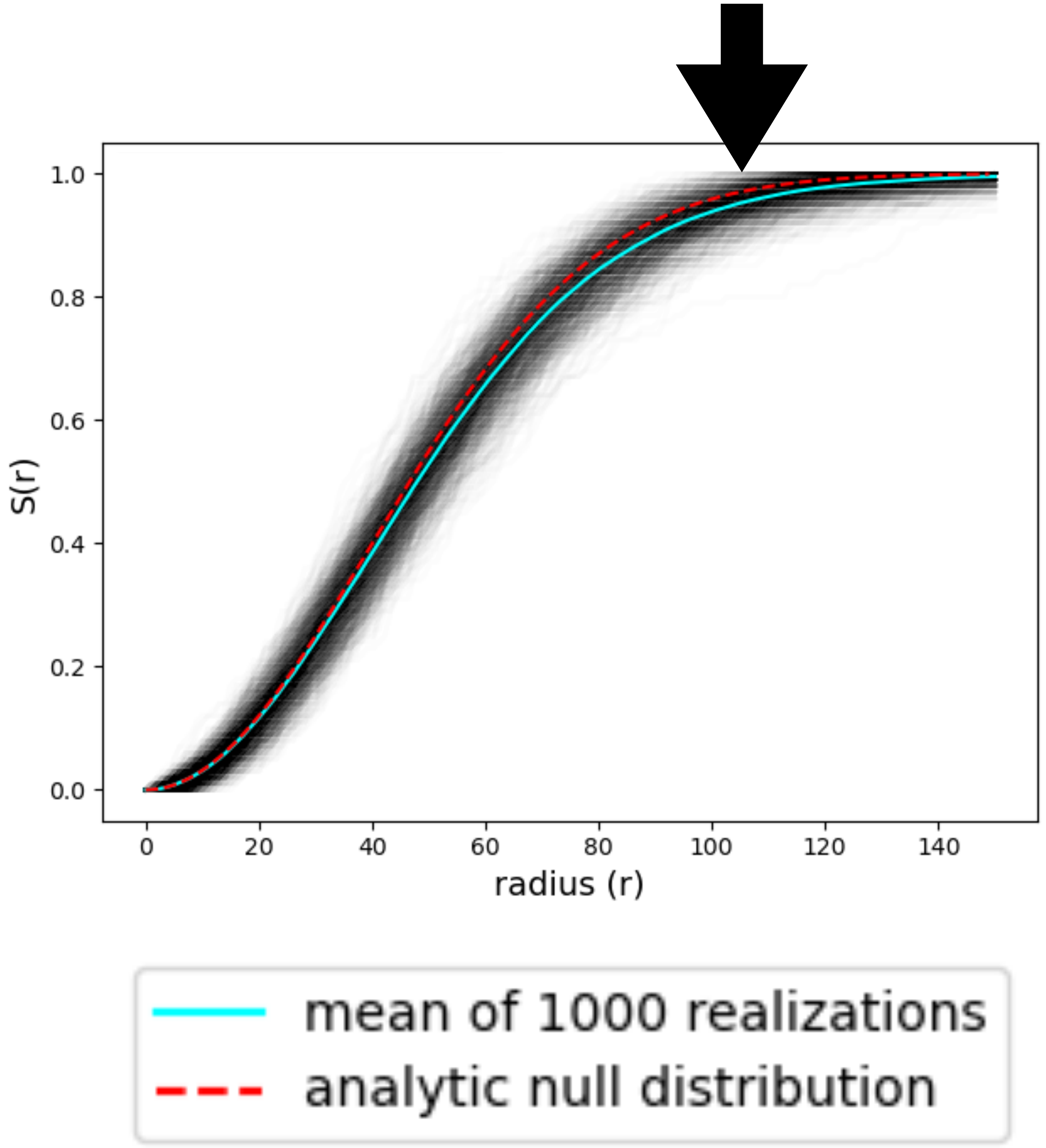




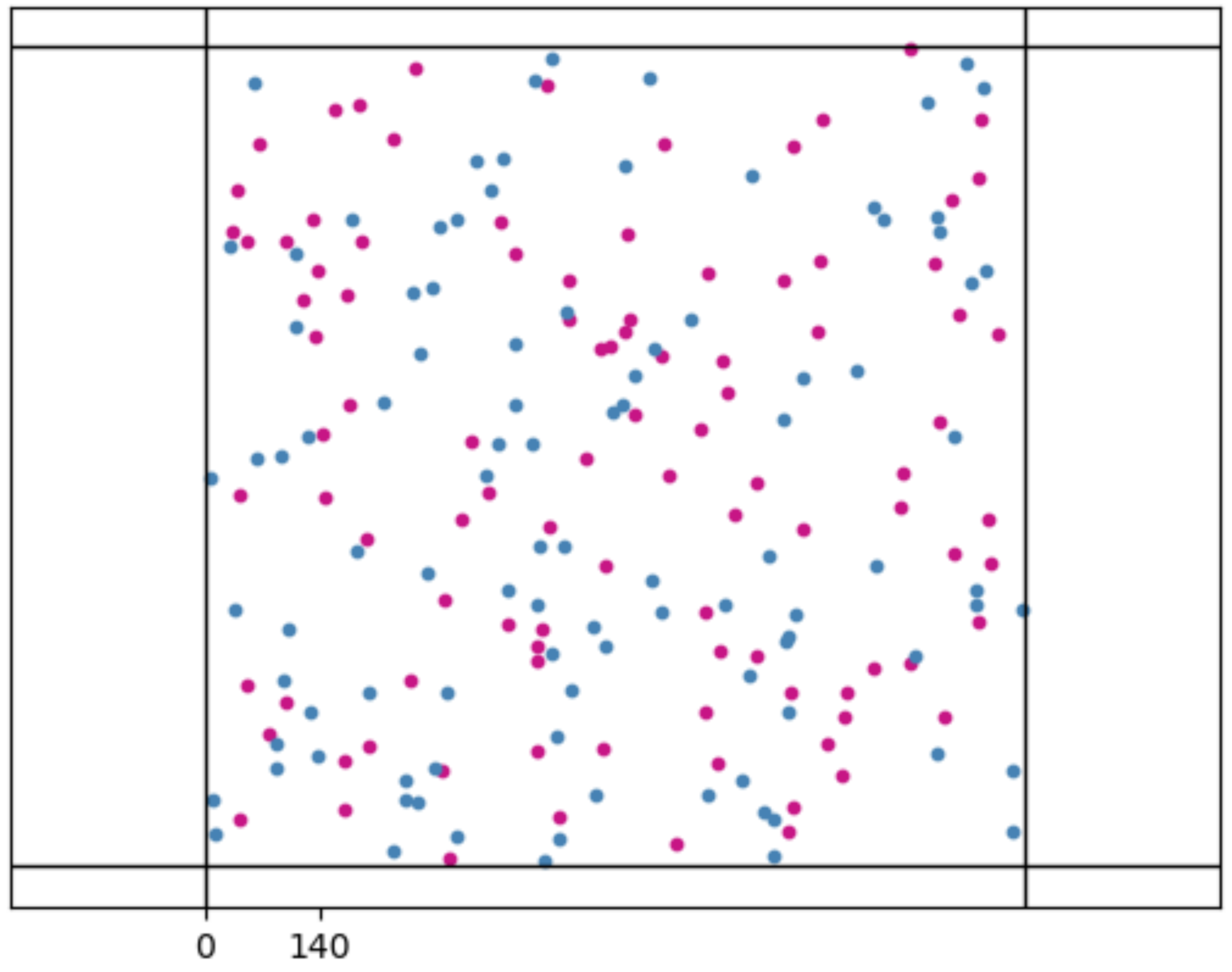
Monte-Carlo-based significance testing

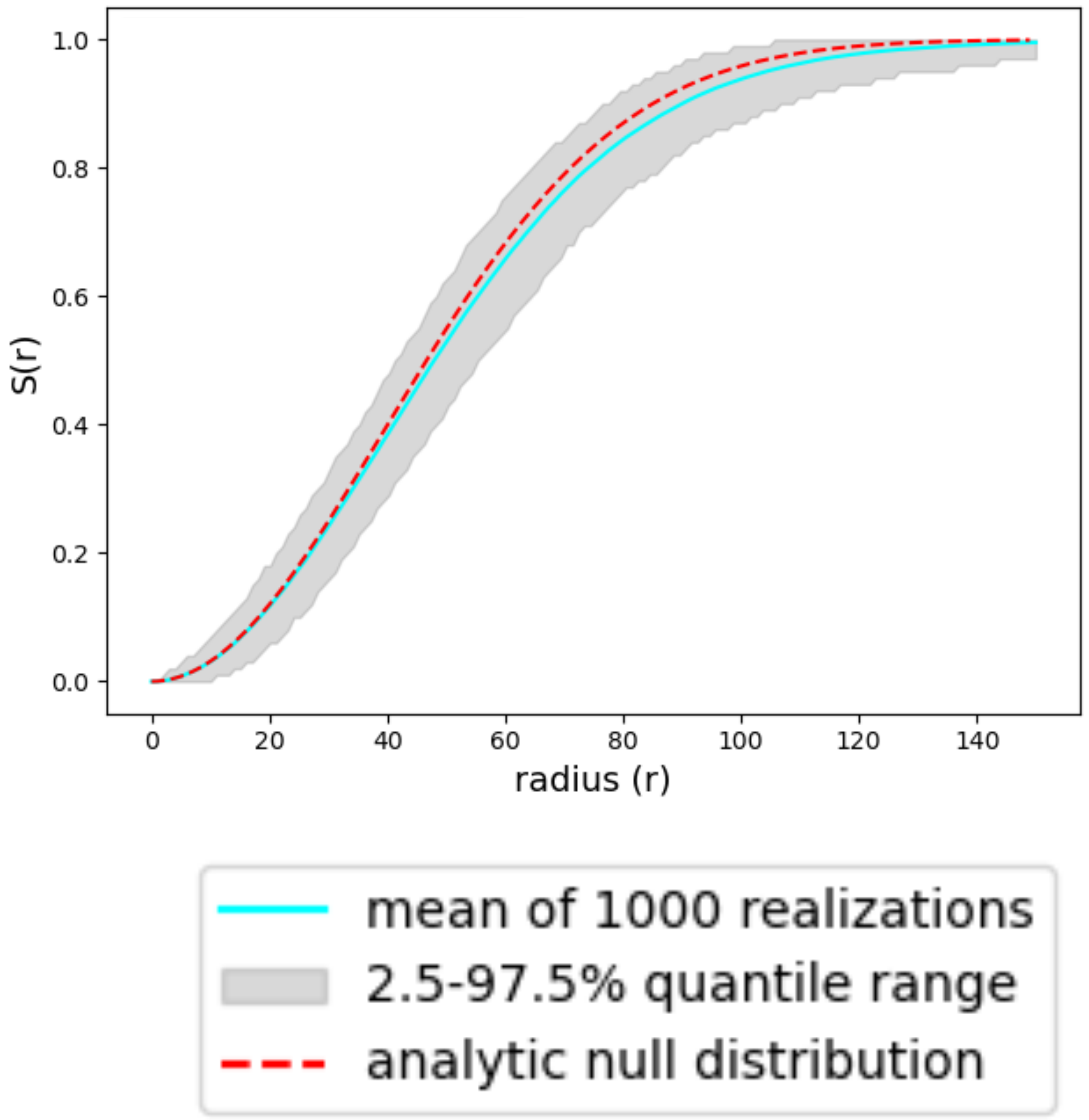


Why?

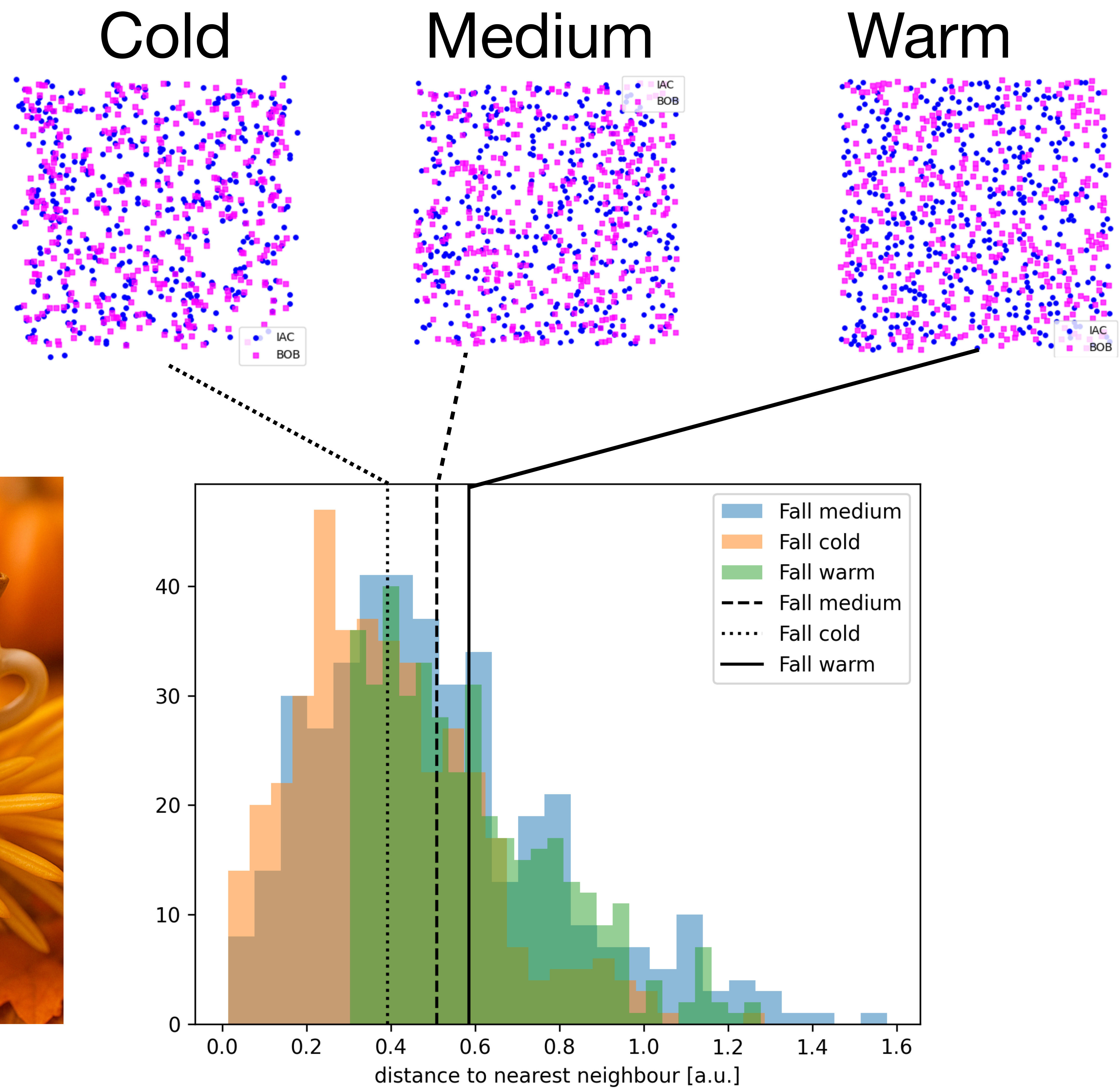


Monte-Carlo-based significance testing

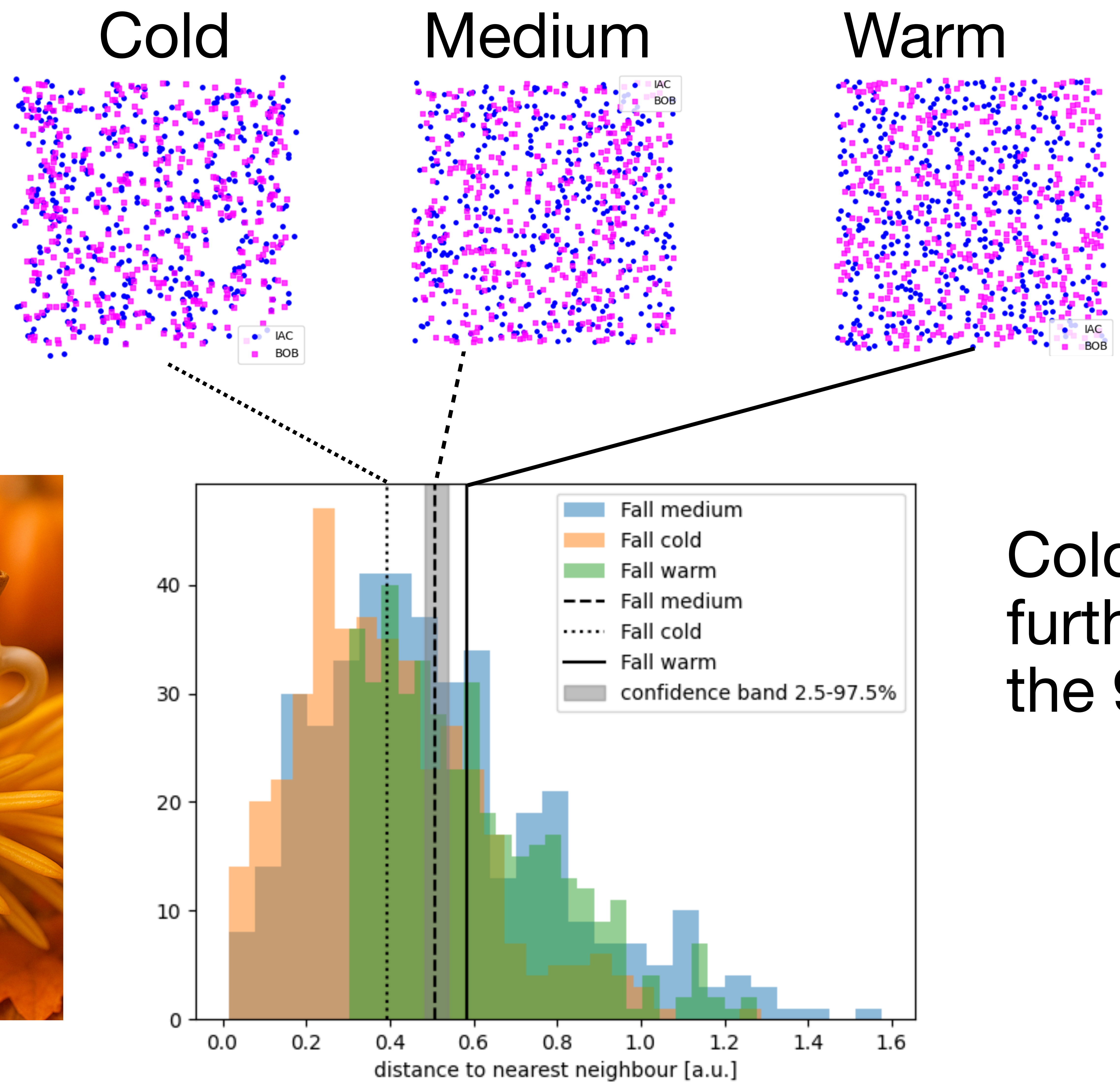




Results: Mean distance IAC -> BOB

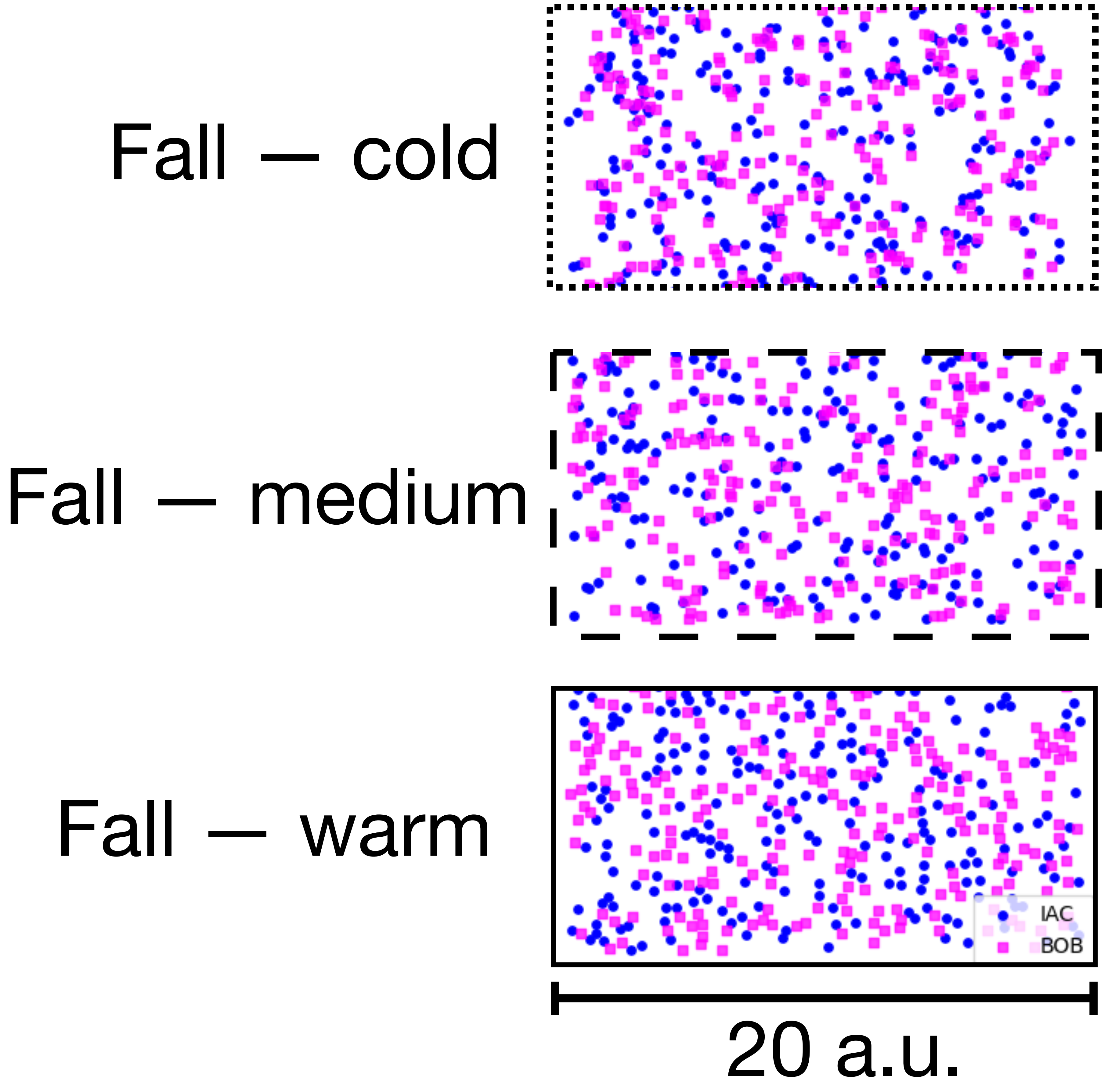


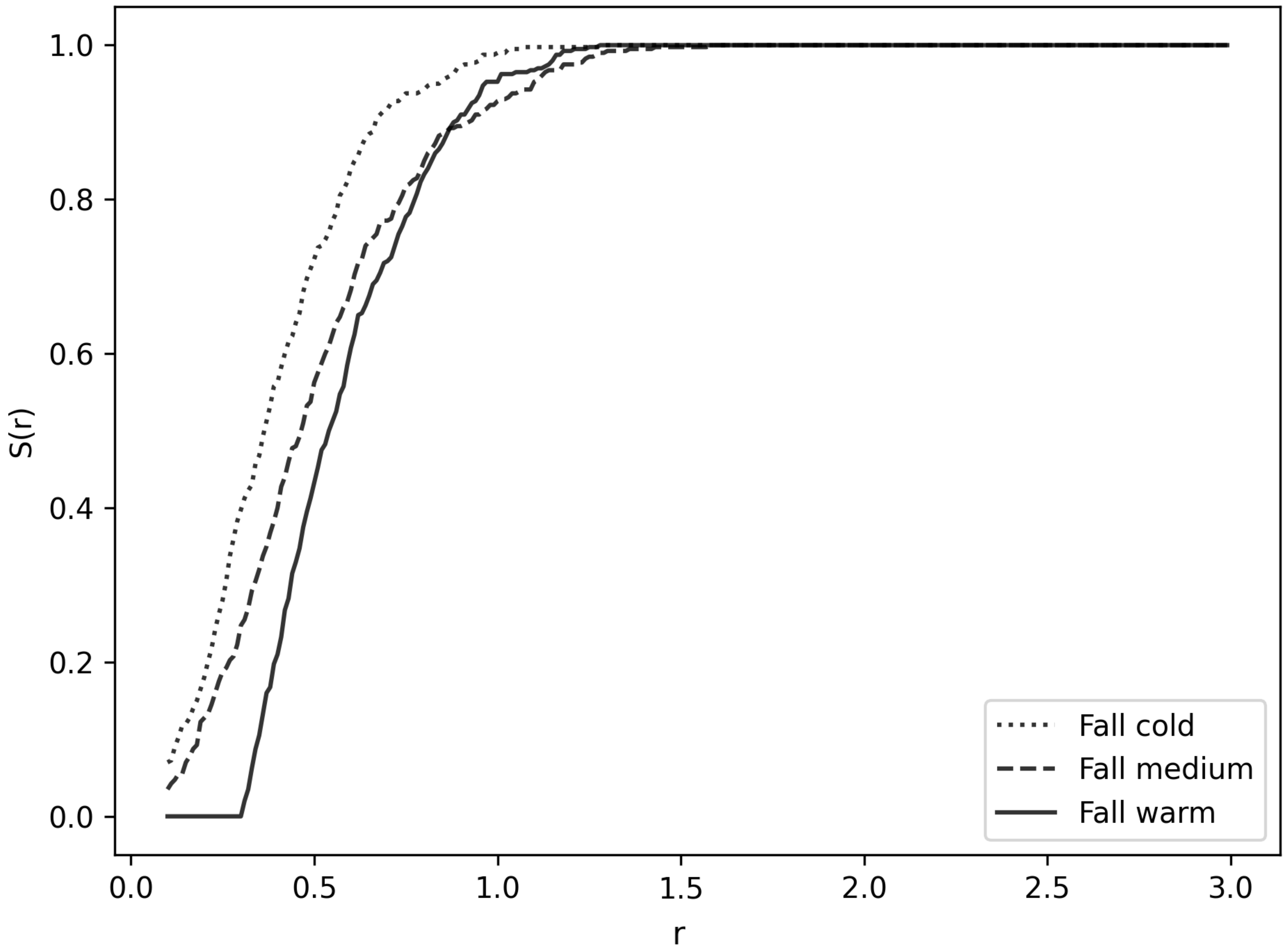
Results: Mean distance IAC -> BOB



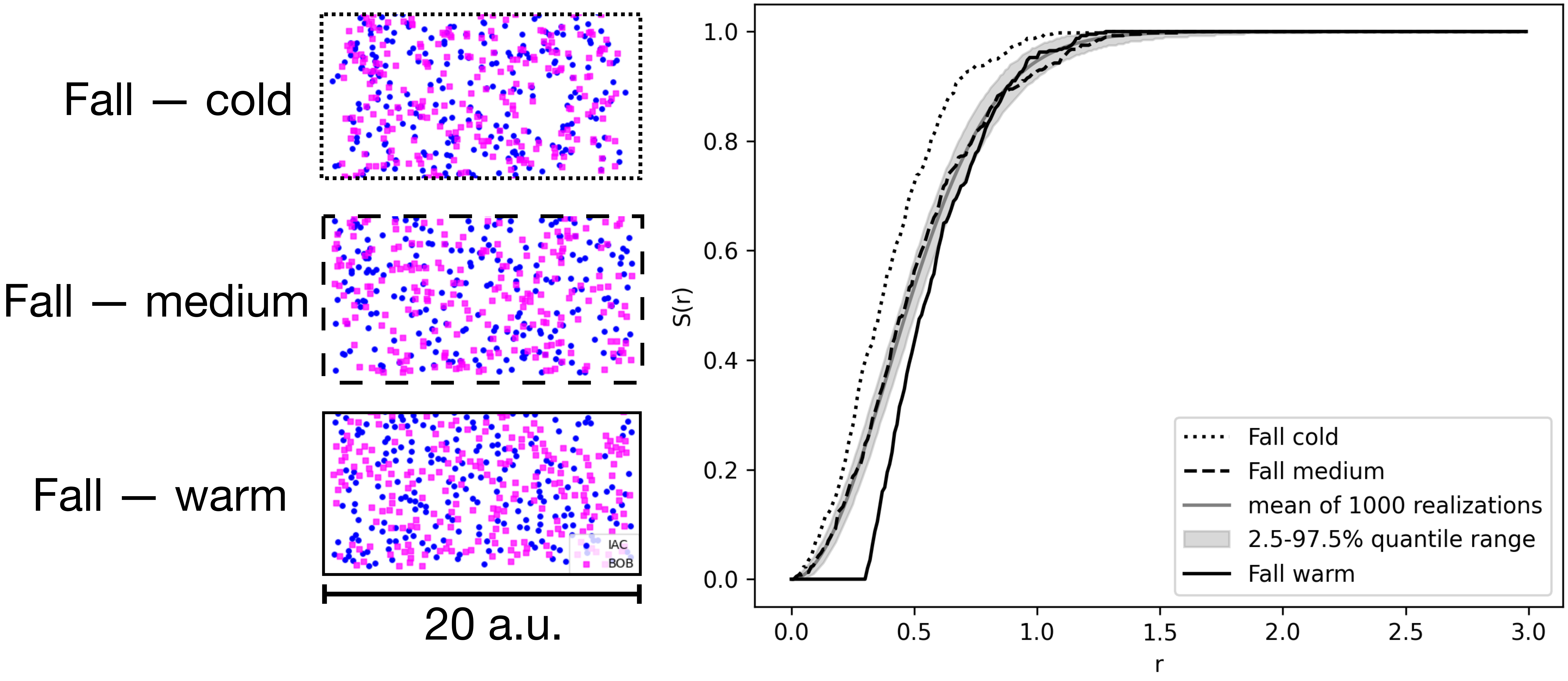
Cold/warm are closer/ further than random under the 95% significance level

Results: Nearest neighbor function



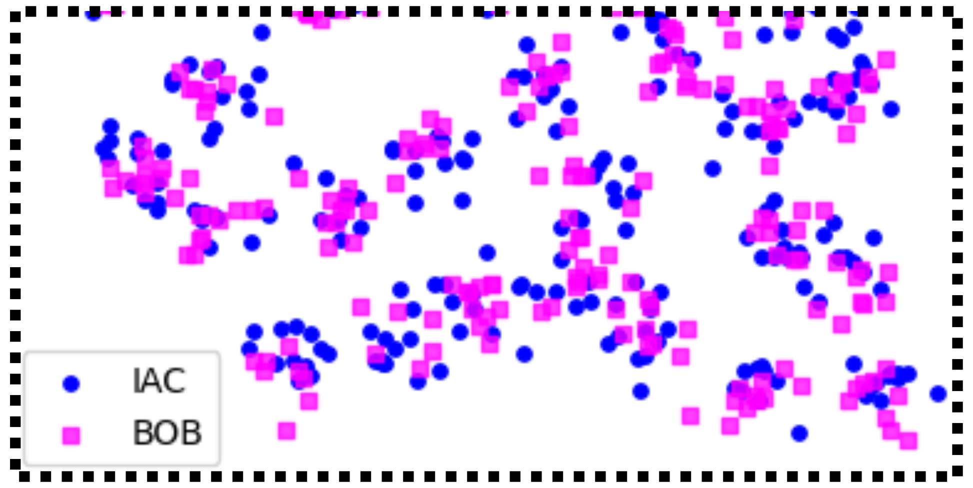


Results: Nearest neighbor function



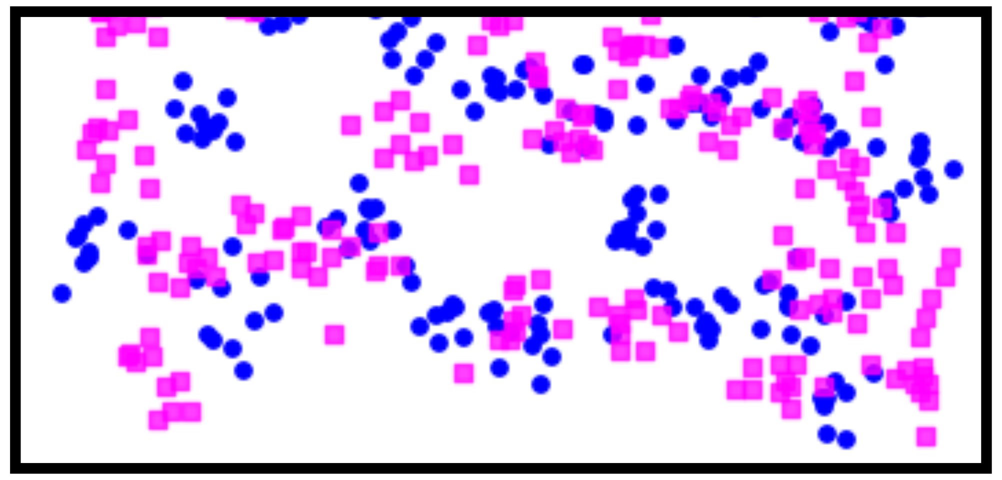
Results: Ripley’s K function

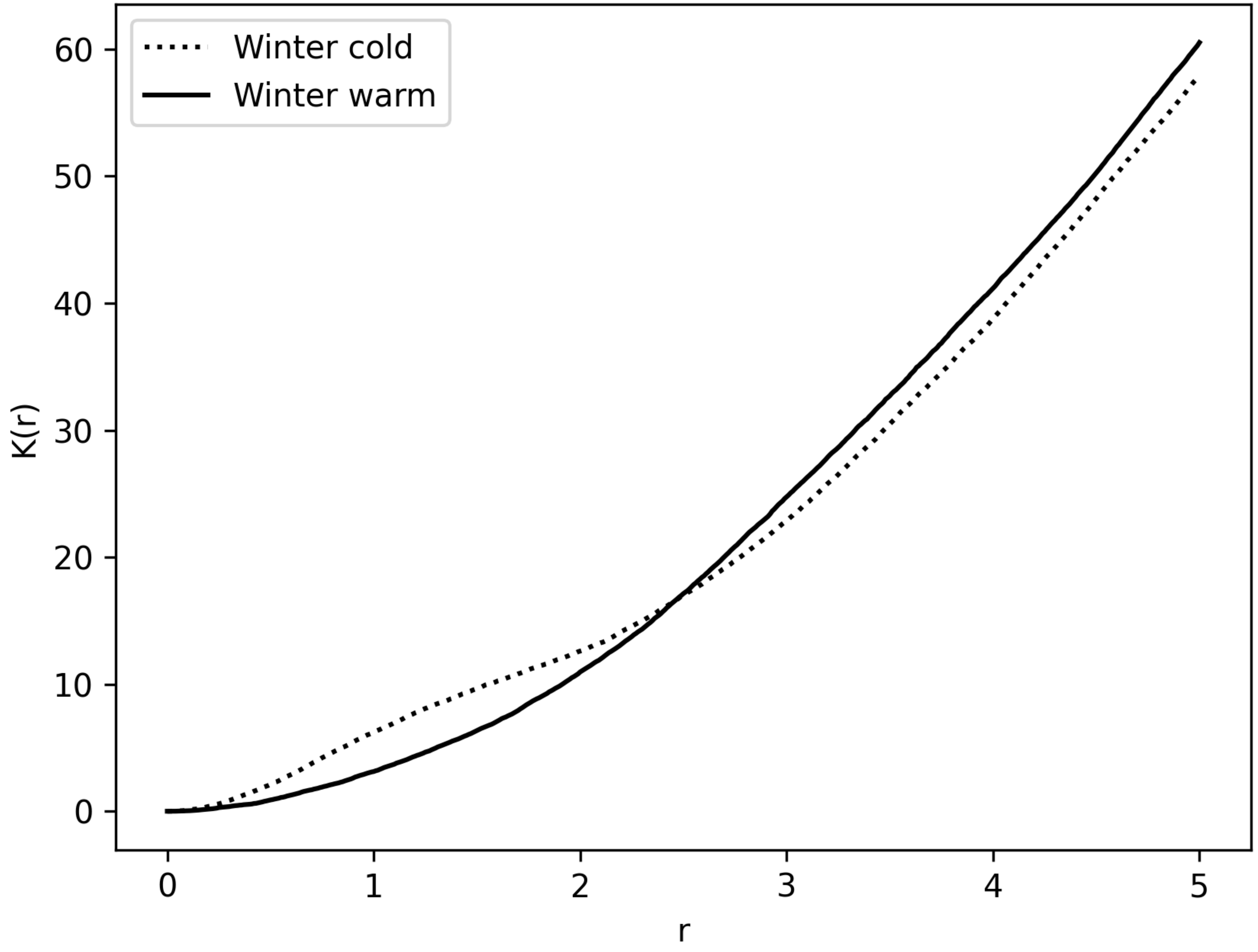
Winter — cold



Winter — medium

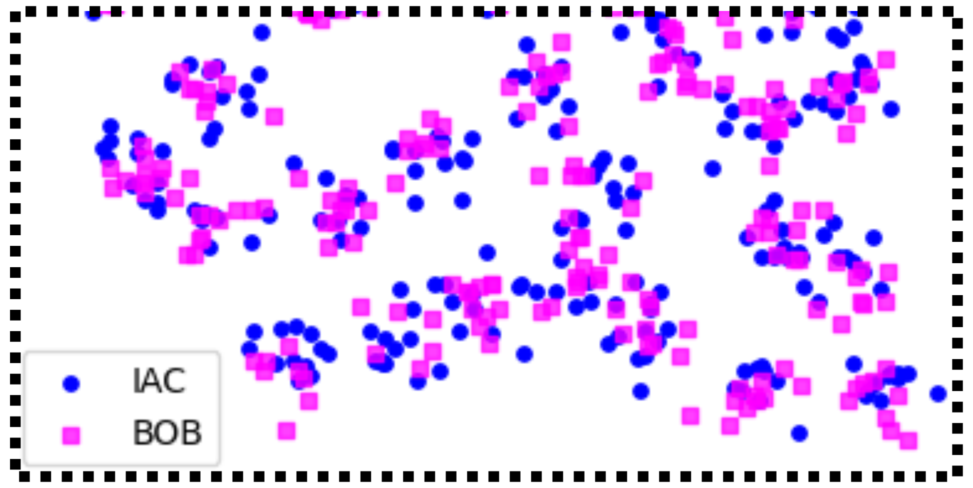
Winter — warm





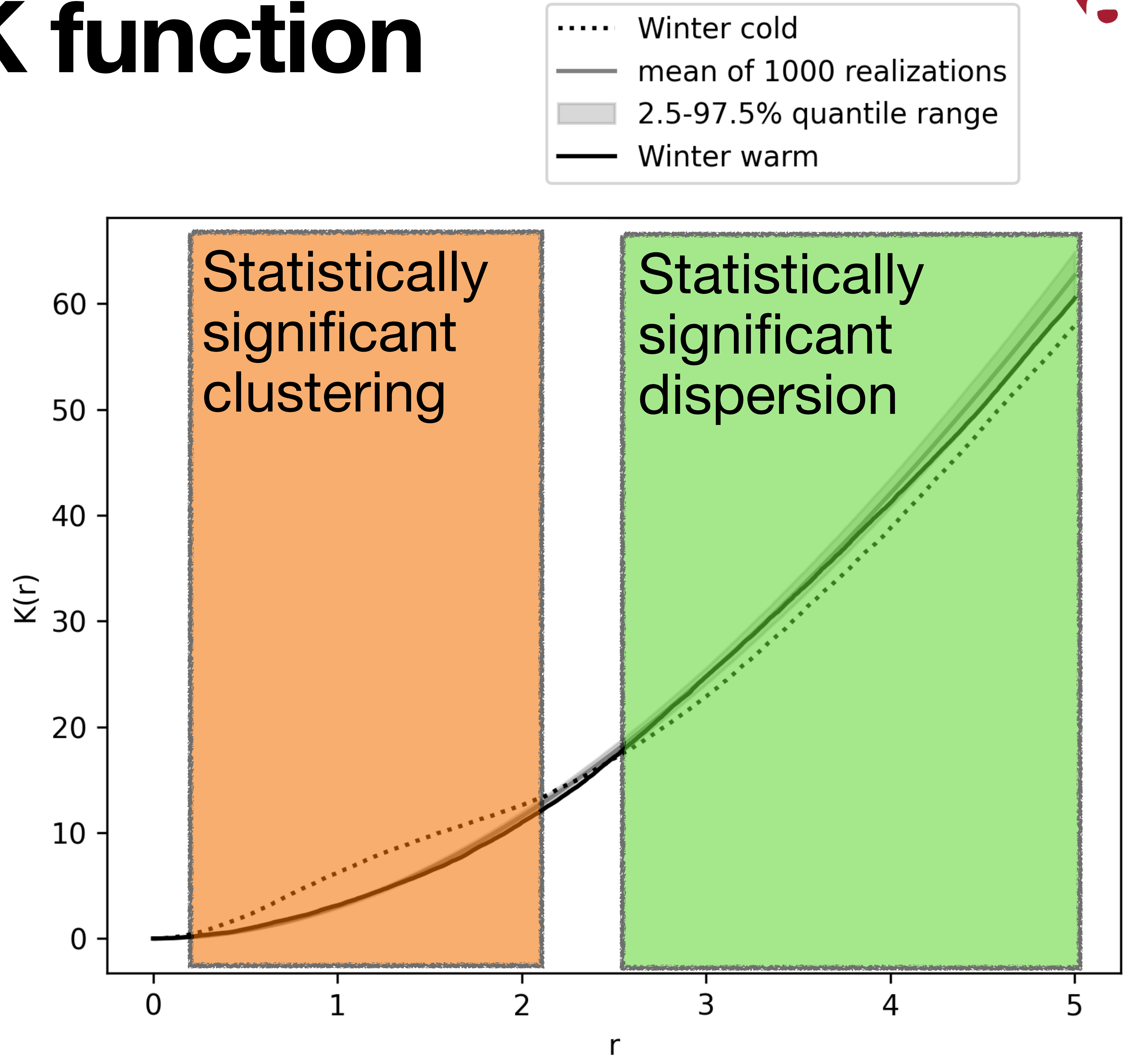
Results: Ripley’s K function

Winter — cold



Winter — medium

Winter — warm



The bigger picture: Monte-Carlo-based significance testing

- Just because your sample isn’t uniformly distributed doesn't mean it is biologically meaningful!
- It is possible to simulate hypotheses beyond uniform distributions.

Python notes

locan

Our implementation of Ripley's K was tested against the Locan library implementation:

https://locan.readthedocs.io/en/latest/tutorials/notebooks/Analysis_Ripley.html#

References

Lagache T, Sauvonnet N, Danglot L, Olivo-Marin JC. Statistical analysis of molecule colocalization in bioimaging. Cytometry A. 2015 Jun;87(6):568-79. doi: 10.1002/cyto.a.22629. Epub 2015 Jan 20. PMID: 25605428.

Ripley, B. D. “The Second-Order Analysis of Stationary Point Processes.” *Journal of Applied Probability*, vol. 13, no. 2, 1976, pp. 255–66. *JSTOR*, https://doi.org/10.2307/3212829. Accessed 13 July 2025.