AI IN SCIENCE TUTORIALS

Working with AI

Image Analysis using Large Language Models (LLMs)

SESSION 4 Using LLM for Super Resolution Analysis

Manuel F. Navedo · UC Davis School of Medicine

Learning objectives

By the end of this session, you'll be able to:

1. Recognize SMLM data as a point cloud, not an image — and prompt accordingly
2. Prompt an LLM to run cluster analysis on a real STORM data
3. Read and verify the code and outputs the model generates — stay accountable
4. Validate clusters against synthetic data or other analysis

Ambitious for one session — doable because the LLM does the heavy lifting while you stay in responsible charge

What this segment covers

1. **The conceptual pivot** From pixels to point clouds — the spine of the whole module
2. **Fundamentals** How localizations are made
3. **Execution** Cluster analysis with Claude writing the code
4. **Metrics menu** What you can measure, and which to trust
5. **Validation** A sample-data test

The conceptual pivot: from pixels to point clouds

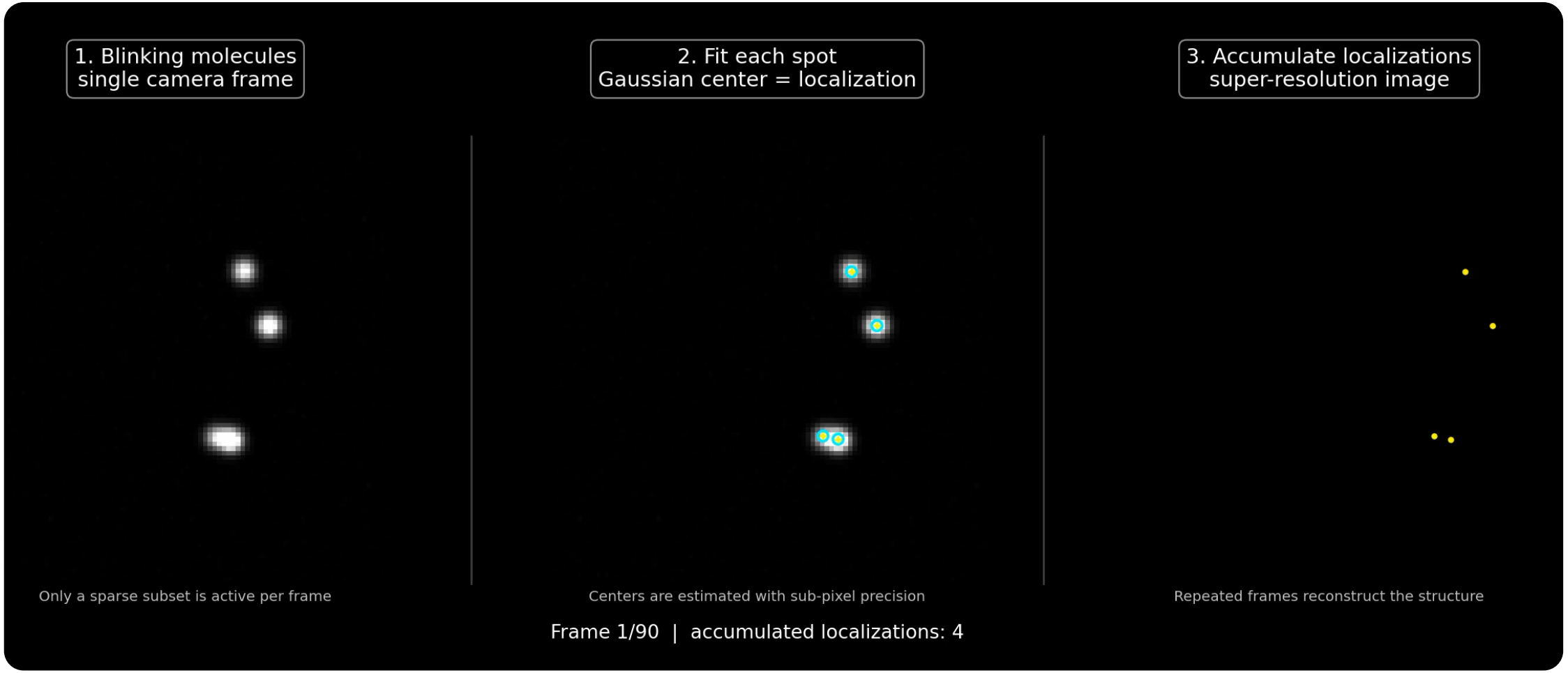
Conventional / earlier sessions

- The data IS an image
- A grid of pixels with intensity / bit depth
- You threshold intensity to find signal
- Resolution set by the pixel + diffraction limit

Localization microscopy (SMLM)

- The data is NOT an image
- A table of coordinates: x, y, (z), photons, frame, uncertainty
- There is no intensity to threshold
- Resolution set by localization precision + sampling

From Blinks to Super-Resolution: How Single-Molecule Localizations Are Made

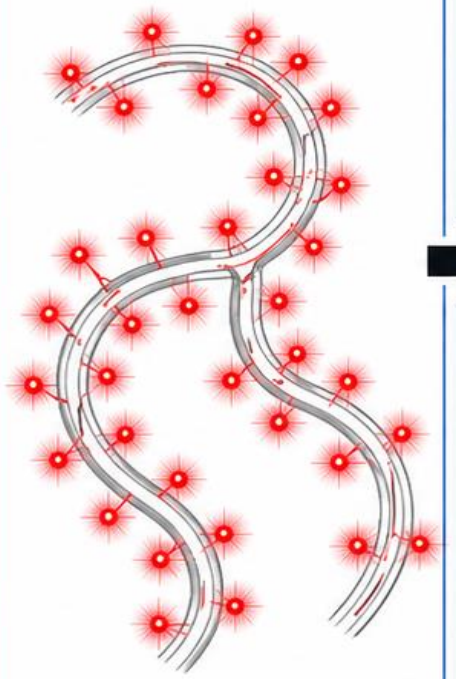


Created with ChatGPT GPT-5.5

How Super-Resolution Localizations Are Made

1 Labeled sample

Many far-red fluorophores (e.g., 647 dye) are attached to a structure.



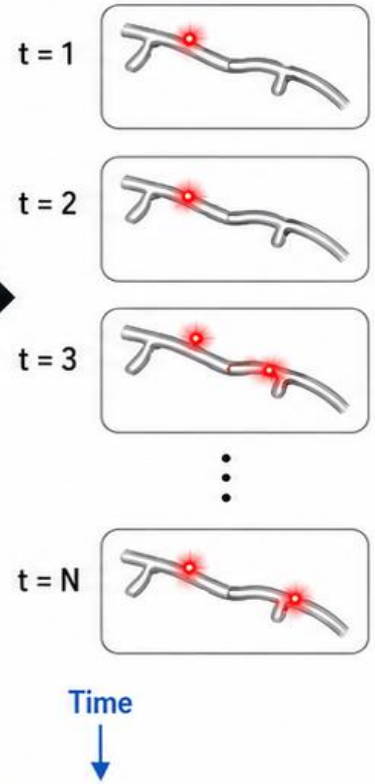
2 Conventional image

All fluorophores are on at once. Their light overlaps, producing a blurry, diffraction-limited image.

Diffraction-limited (~200–300 nm)

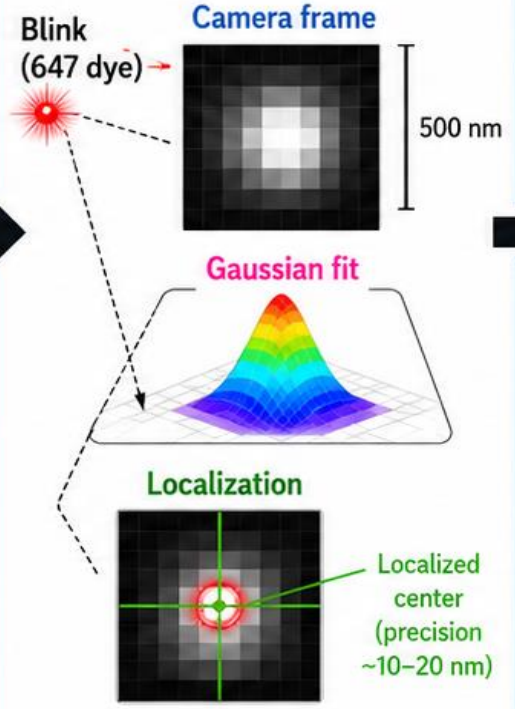
3 Sparse blinks over time

Only a few fluorophores are on in each camera frame.



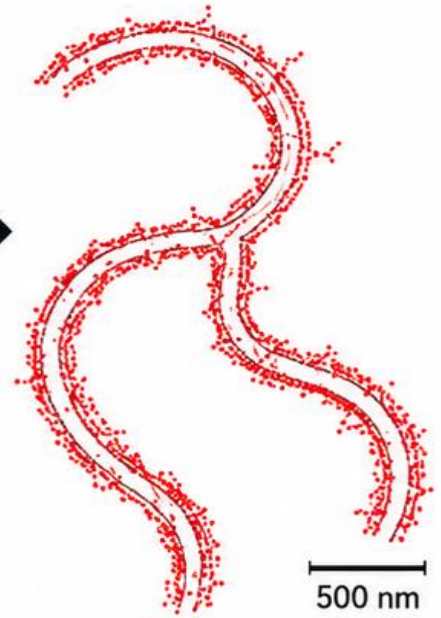
4 Detect each blink

Each blink appears as a fuzzy spot in one frame. Its center is localized precisely by fitting a Gaussian.



5 Reconstruct super-resolution image

Many localizations from many frames are combined to build a sharp, high-resolution image of the structure.



Super-resolved (~10–20 nm precision)

Single far-red fluorophores (e.g., 647 dye) blink on and off.
Each blink is detected, localized, and combined to reconstruct a super-resolved image.

Each dot = one localization (molecule position)

UCDAVIS HEALTH

Created with ChatGPT GPT-5.5

Fundamentals — how a localization is made

1

Blinking

Sparse fluorophores switch ON in any one frame (photoswitching)

2

Fitting

Each isolated PSF is fit (Gaussian / MLE) to sub-pixel center

3

Localization

Center + uncertainty stored as one row in a table

4

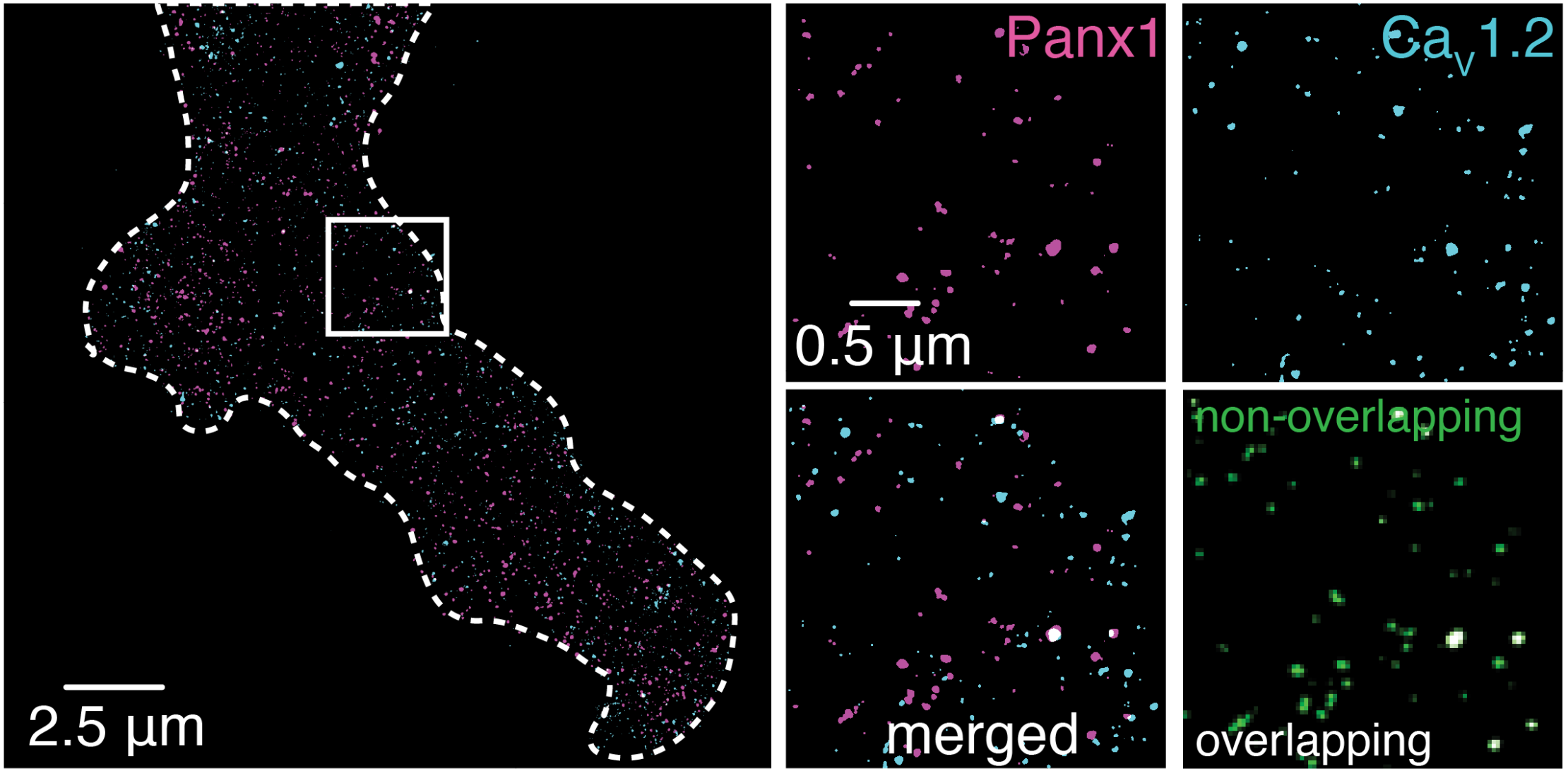
Table

Thousands of frames → a point cloud of all localizations

Pipeline: raw blinking movie → Commercial software / ThunderSTORM / Picasso → localization table

I can demo the movie→table step live; the prompt-driven analysis begins at the coordinates.

Super-Resolution Analysis: So Many Spots — Now What?



Hong, Martin-Aragon et al, Circ Res 2025

You have the image — now extract more from it

Your clusters are already reconstructed

From here it's the series' shared arc — applied to the image

1

Talk to the model

Describe your image and list what you want measured — before any code

2

Build the analysis

You and Claude build the script that measures the clusters you already have

3

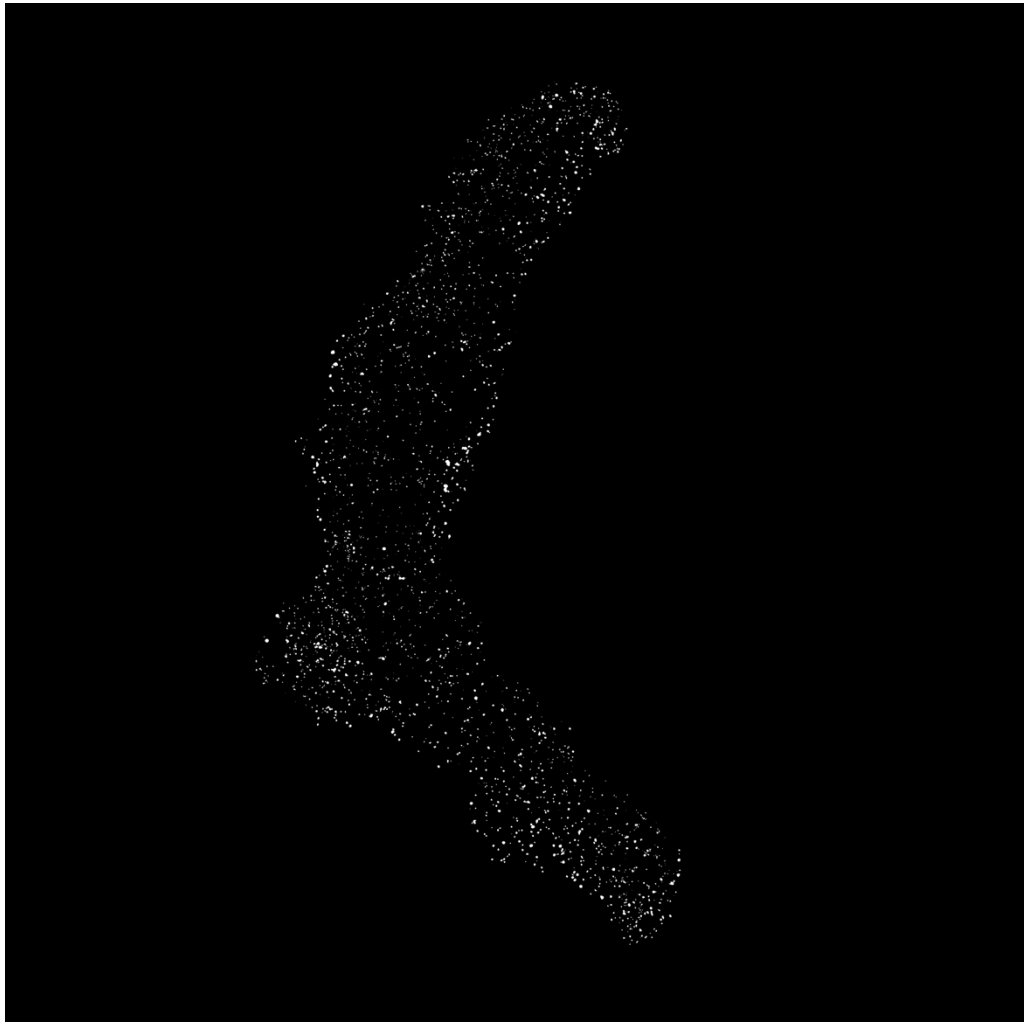
Validate it

Confirm it against a chance control — is the analysis real?

One running dataset the whole way through: your reconstructed SR image — clusters already established

You have the image — describe what you want

Arc · step 1 of 3 — before any code, spell out the analysis



Four dimensions of a cluster image

01 Sample

Organism, cell type, target protein(s), treatment — the biology you know

02 The image

Reconstructed SR image & clusters already established - nm/pixel calibration, channels, and is the background noisy (need a cell mask)?

03 Structure

Expected organization and scale — clusters? how spaced? one protein or two interacting?

04 Question

Cluster size? Cluster density? Intramolecular distance? Proximity? colocalization? — and what it's compared against

"The image" replaces "the raw data". Calibration, channels and a noisy background are things you describe here

THIN vs RICH: spell out what the script should do

THIN

“Analyze this image.”

Arbitrary threshold, no cell mask, whatever it guesses — and you can’t reproduce it

RICH — a spec list what you want; start step by step

1. **Detect the cell** — analyze only inside it (background can be noisy)
2. **Detect clusters by threshold** — they’re already established in the image
3. **Report size, density, coverage**
4. **Intramolecular distance** — cluster-to-cluster, one channel
5. **Two channels** — intermolecular distance, proximity ≤ 30 nm, colocalization
6. **Randomize as a control** — is the proximity beyond chance?

The RICH prompt is a checklist. Each line is a decision you can see, change, and report

TRY IT NOW · 5 min — write the spec for your image

In plain prose, describe your image and list — in order — what you want the script to measure

No code yet

1 Write · 4 min

Describe the image (calibration, channels, background), then ask to detect the cell edge

2 Swap · 1 min

Trade with a neighbor. Could Claude build it with no follow-up question?

WATCH FOR

nm/pixel calibration · cell detection (noisy background?) · how a cluster is thresholded · one channel or two channels · which distances · a chance control · reports

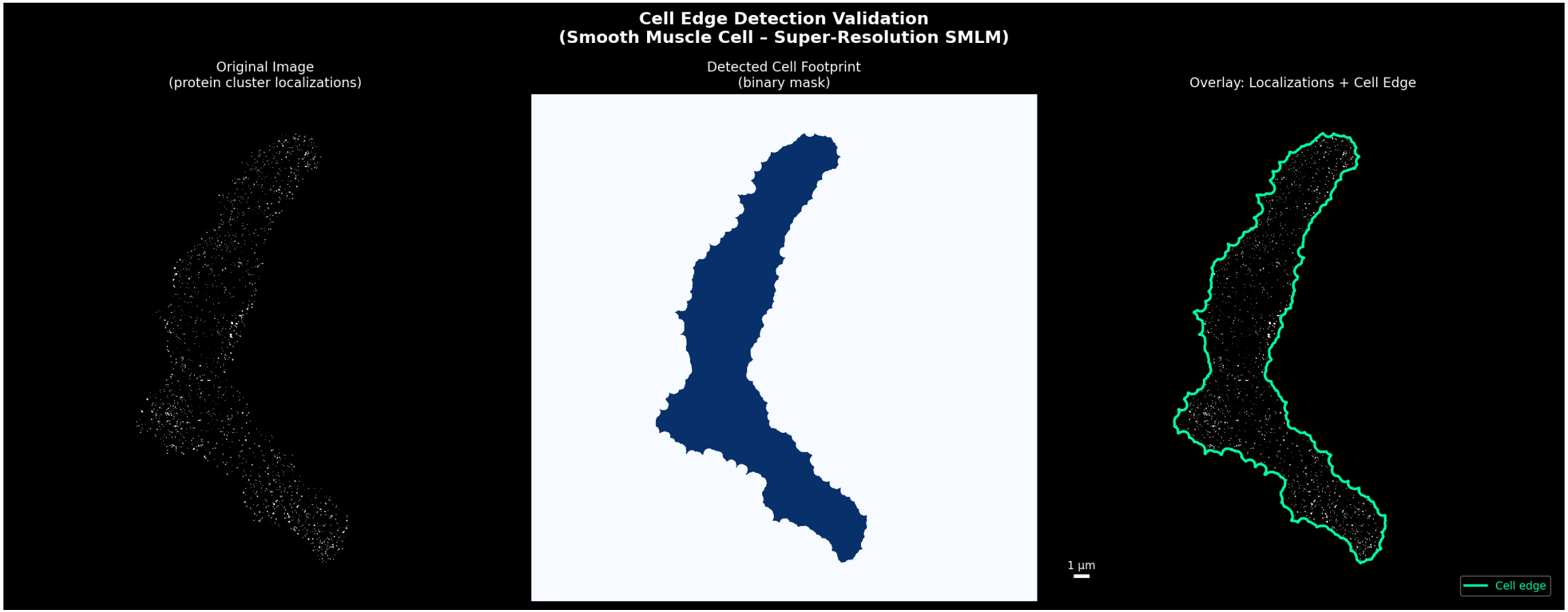
What was my prompt and what did I get?

This is a reconstructed super-resolution image of a smooth muscle cell. The pixel size is 5 nm/pixel. The white spots are clusters of protein X. I want to analyze only clusters within the cell footprints. First, detect the cell edge and show me an image for validation of the cell edge detection.

Reading the file-reading skill to understand how to process the uploaded image
Now let me process the image with Python to detect the cell edge:
Ran 5 commands, read a file
Here's the validation figure. Here's what was done and the key stats:
Method: The cell edge was detected using a morphological approach suited to sparse SMLM data:
1.Binarize – threshold the localization image (pixel > 10)
2.Dilate – expand spots with a disk kernel (~400 nm radius) to bridge the gaps between sparse clusters and merge them into a continuous filled region
3.Fill holes – close any remaining interior voids
4.Erode – shrink back by the same amount to recover the true cell boundary without the dilation padding
5.Largest connected component – select the single cell body
6.Contour tracing – extract the boundary polyline
Cell footprint statistics:
•Area: ~193.6 µm²
•Spatial extent: ~3.1–36.9 µm (long axis) × ~9.8–26.5 µm (short axis) — consistent with a spindle-shaped smooth muscle cell
•The teal contour in the overlay panel traces the detected cell edge
Does the contour look right to you? If it's too tight (cutting off peripheral clusters) or too loose (including background), I can adjust the dilation radius. Once you're happy with it, I can proceed to mask the localizations and analyze only the intra-cellular clusters.

What was my prompt and what did I get?

This is a reconstructed super-resolution image of a smooth muscle cell. The pixel size is 5 nm/pixel. The white spots are clusters of protein X. I want to analyze only clusters within the cell footprints. First, detect the cell edge and show me an image for validation of the cell edge detection.



Refining the edge detection

The contour looks super tight, and it could be loosened a bit.

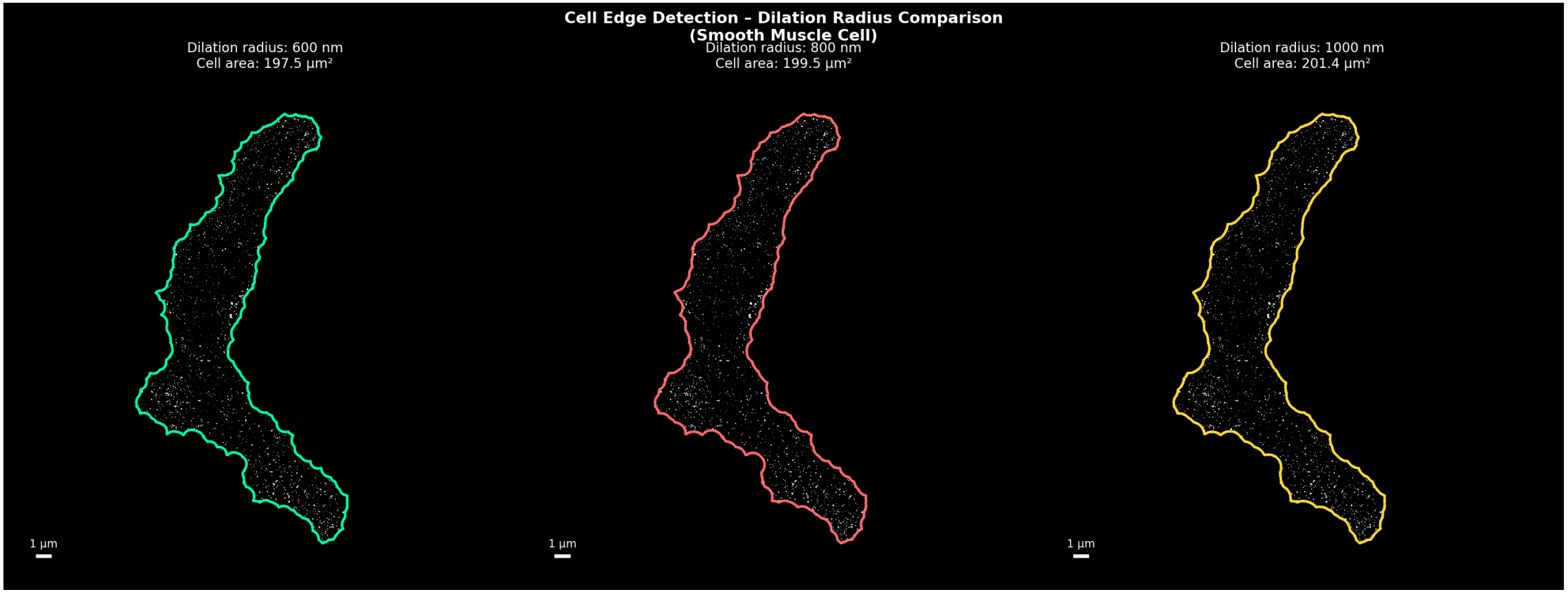
Here are three options with progressively looser contours — the dilation radius controls how far the boundary extends beyond the outermost localizations:

- **600 nm** — moderate loosening, hugs the cluster distribution fairly closely
- **800 nm** — a comfortable margin around all visible spots
- **1000 nm** — the most generous boundary, well outside the peripheral clusters

Which of these looks right to you, or would you like to try a value in between? Once you confirm, I'll lock in that contour and move on to filtering clusters within the cell footprint.

Refining the edge detection

The contour looks super tight, and it could be loosened a bit.



Build the analysis – create your own order

1. **Detect the cell** analyze only inside — ignore noisy background
2. **Detect clusters** by threshold; they're already established
3. **Measure** size, density, intra- / inter-molecular distance, morphology
4. **Control** randomize positions — is it beyond chance?
5. **Reports** create reports for easy transfer to statistical softwares and analysis

This is exactly how the scripts progress — each step adds one capability the last one set up

The decisions you own

Cell mask

Include or exclude the noisy background — what region counts

Cluster threshold

How the intensity cut defines each cluster’s extent

Proximity threshold

What counts as “close” across channels — e.g. 30 nm

Calibration

nm / pixel — it sets every distance you report.

Set them on purpose, report them, sweep the sensitive ones. A threshold is a decision, not a default

TRY IT NOW · 5 min — write the spec for your image

In plain prose, describe your image and list — in order — what you want the script to measure

No code yet

1 Write · 4 min

Describe the analysis that you want to obtained – e.g. cluster size, and density

2 Swap · 1 min

Trade with a neighbor. Could Claude build it with no follow-up question?

WATCH FOR

nm/pixel calibration · cell detection (noisy background?) · how a cluster is thresholded · one channel or two channels · which distances · a chance control · reports

What was my prompt and what did I get?

I want to detect clusters only inside the cell. I want to measure their size, the cluster density inside the cell, the intramolecular distance between them, and the morphology.

Q: Which dilation radius looks best for the cell boundary?
A: 1000 nm

Great, locking in 1000 nm. Let me now run all the analyses.

Here's the full analysis. Key findings:

Cluster Size

•Mean diameter: **61.6 ± 28.1 nm**, median 56.4 nm — consistent with nanoscale protein complexes resolved by SMLM

Cluster Density

•Cell footprint: **201.4 µm²**

•**1,900 clusters** inside the cell → **9.43 clusters/µm²**

Inter-cluster Distance (NND)

•Mean: **195.0 ± 119.9 nm**, median **178.9 nm** — the distribution is right-skewed, indicating most clusters have a close neighbour but a subset are more isolated (visible as the warm-coloured dots in the map)

Morphology (based on 885 blobs matched to individual connected components)

•**Circularity 0.921 ± 0.165** — clusters are predominantly round/compact (1.0 = perfect circle)

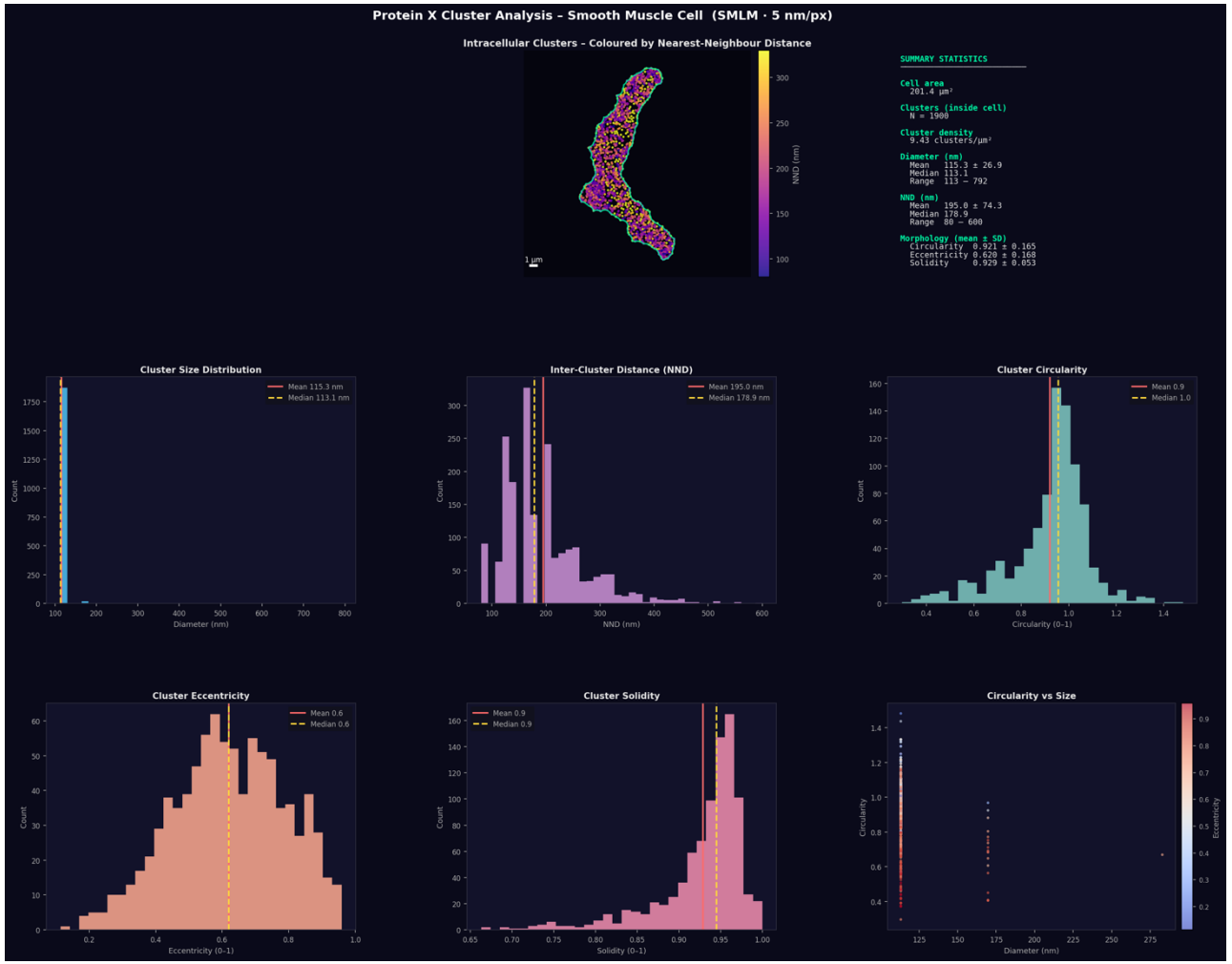
•**Eccentricity 0.620 ± 0.168** — a moderate degree of elongation on average

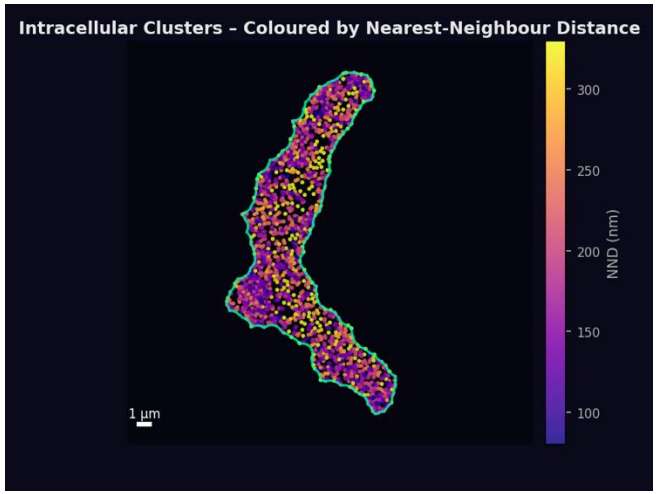
•**Solidity 0.929 ± 0.053** — very solid/convex shapes with few concavities

The scatter plot (bottom right) shows that smaller clusters tend to span the full circularity range, while larger ones are more consistently circular.

What was my prompt and what did I get?

I want to detect clusters only inside the cell. I want to measure their size, the cluster density inside the cell, the intramolecular distance between them, and the morphology.





SUMMARY STATISTICS

Cell area
201.4 µm²

Clusters (inside cell)
N = 1900

Cluster density
9.43 clusters/µm²

Diameter (nm)
Mean 115.3 ± 26.9
Median 113.1
Range 113 – 792

NND (nm)
Mean 195.0 ± 74.3
Median 178.9
Range 80 – 600

Morphology (mean ± SD)
Circularity 0.921 ± 0.165
Eccentricity 0.620 ± 0.168
Solidity 0.929 ± 0.053

UCDAVIS HEALTH

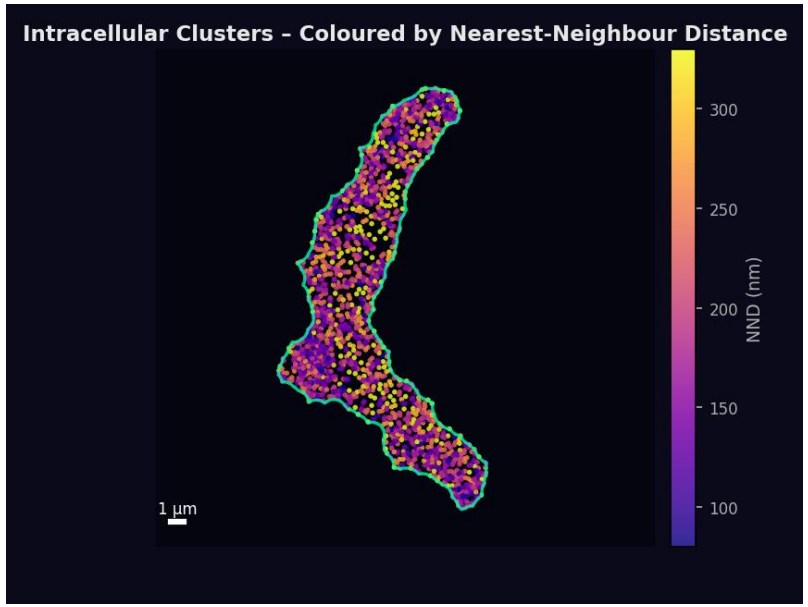
Would you like any of these analyses extended — e.g. spatial statistics (Ripley's K / L-function to test clustering), pairwise all-to-all distance distributions, or **export of cell/per-cluster measurements to a CSV**?

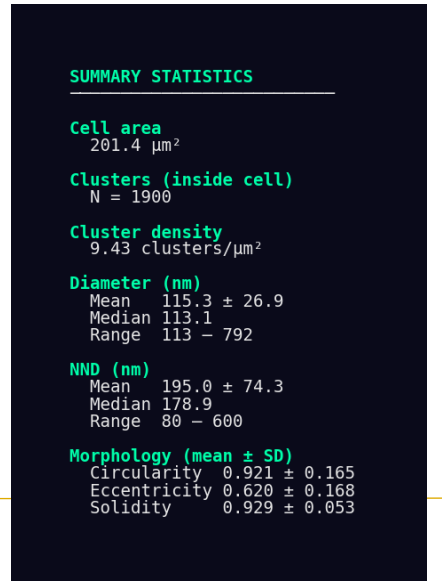
The result you can read: the map is the interface

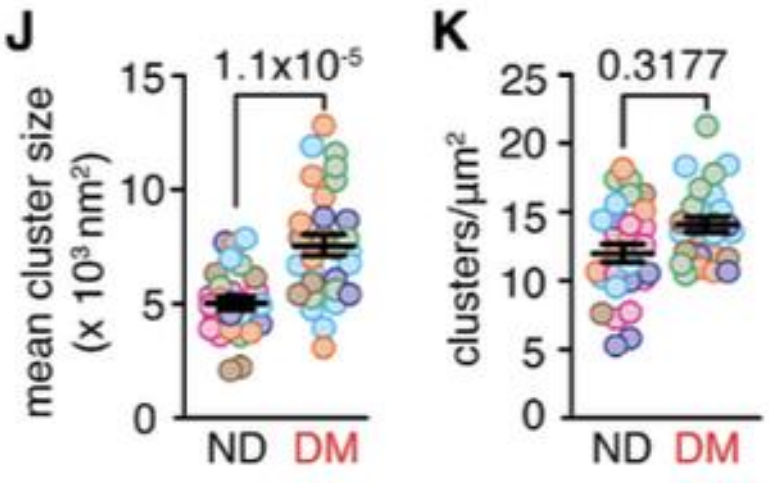
You don’t have to read the code. You read the map — and that you can do

- Clusters coloured by distance match what you see → right, however the code looks
- Colours land on background or miss real clusters → wrong, however elegant

Code literacy is optional - Biological judgment is not







Martin-Aragon et al, Circ Res 2022

Beyond size & density: what the image still holds

Commercial software gives you

ThunderSTORM · Picasso · vendor tools

- Cluster size
- Cluster density
- Cluster count

What you extract with a Claude script

from the same reconstructed image

- Intramolecular distance (one channel)
- Intermolecular distance (two channels)
- Proximity within ~30 nm
- Colocalization / overlap

Size & density are solved. The value is the spatial relationships — pulled from the picture you already have

Validation – Is it real? The randomization control

Compare your results with prior data from your group and others

Compare your result to the same clusters placed at random

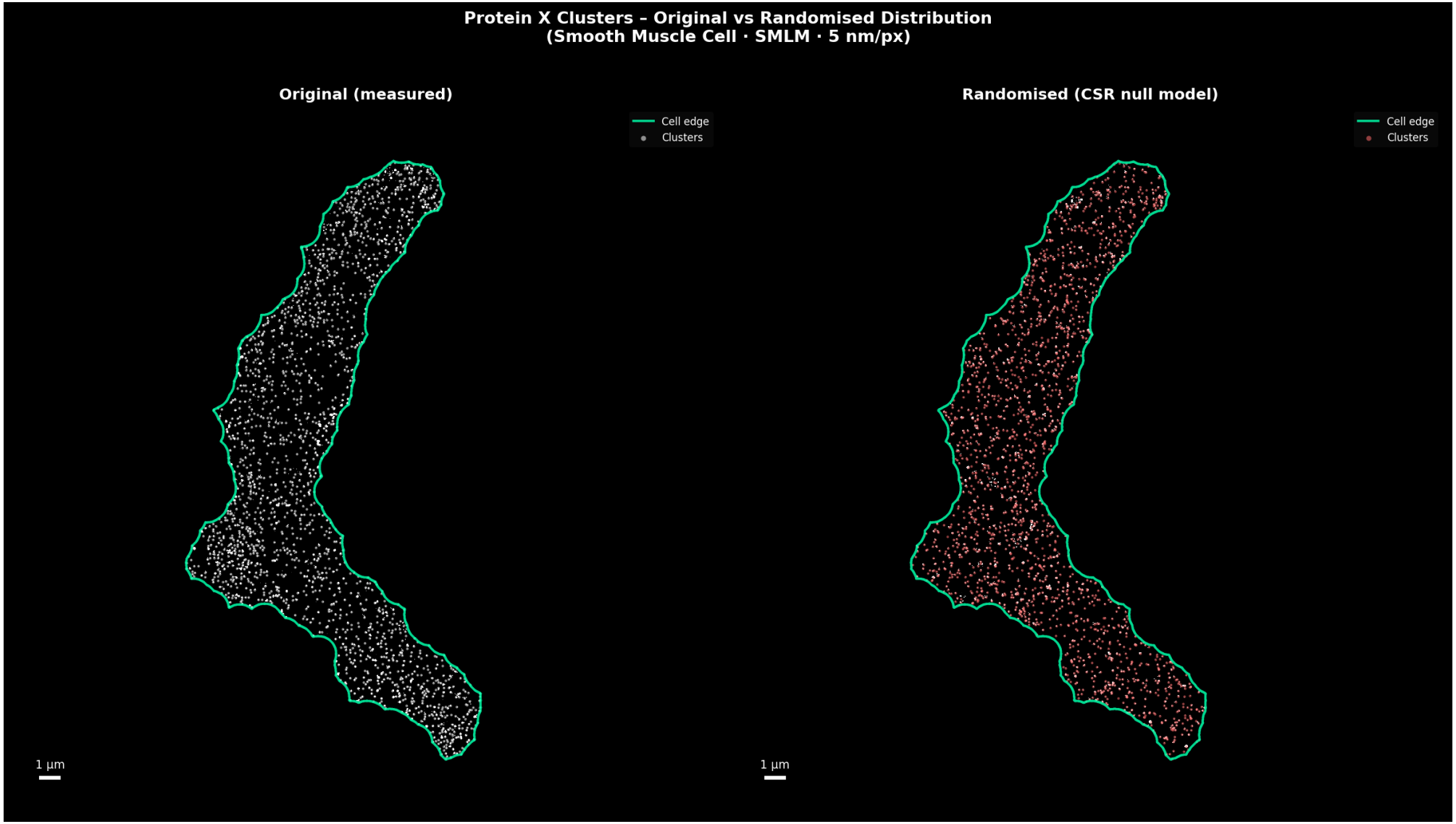
S **Shuffle** Randomize cluster positions inside the same cell boundary

R **Recompute** Run the exact same proximity / colocalization on the shuffled data

C **Compare** Observed ≫ randomized → real. Observed ≈ randomized → chance

You know what to do. Ask AI to create an image of the same cell where clusters are randomly distributed

Randomized images



UC DAVIS HEALTH

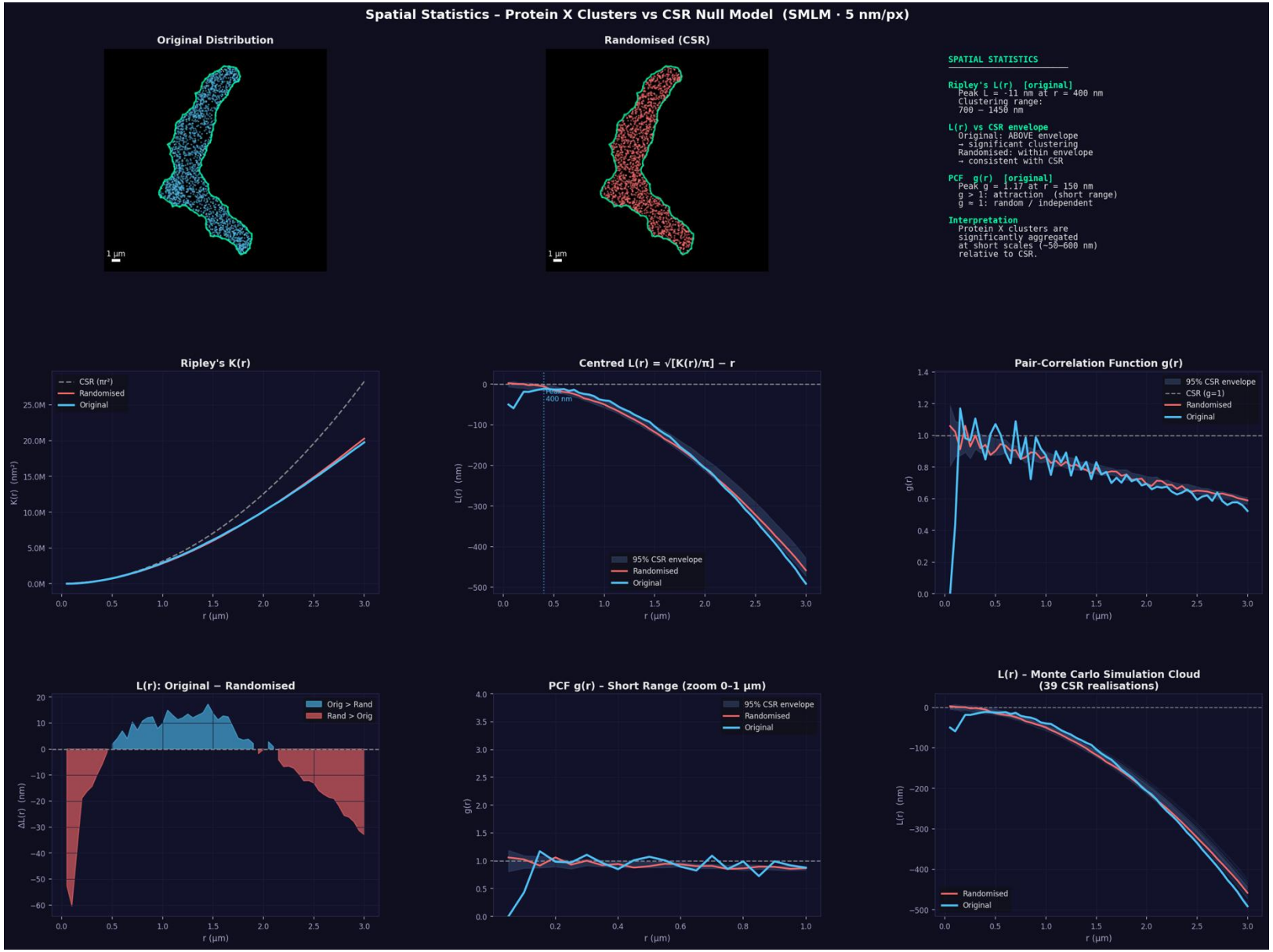
This randomised image is the standard null model used for spatial statistics tests like Ripley's K/L-function or pair-correlation function — would you like to run those comparisons between the original and randomised distributions?

Comparison

Yes, perform the analysis!

Conclusion

Protein X clusters are **non-randomly distributed** inside the smooth muscle cell. They exhibit significant short-range aggregation (clustering scale ~50-300 nm, peak ~250 nm) relative to a complete spatial randomness null model, suggesting cooperative or scaffold-mediated organization at the nanoscale



Ask AI to reanalyze the randomized data and compare it with your experimental data; create CSV report

Built with Claude — one guided script, yours to extend

- **Not in the commercial menu** — intramolecular / intermolecular distance, proximity, colocalization — Claude wrote them
- **One guided script** — single- or two-channel mode, config at the top, hands-on prompts inside
- **Extend by asking** — “add a radial distribution,” “report per-cluster shape,” “sweep the proximity threshold, make a script that can be run outside AI”
- **Reproducible** — script + calibration + thresholds + randomization = a re-runnable protocol

This is LLM-forward: the model builds the analysis commercial tools don’t ship — you verify it and own the result

Recap: four principles — our anchor was Reproducibility

- **R** **Reproducibility** — your protocol re-runs — image, script, parameters, versions ← ***today's anchor***
- **A** **Accountability** — you own the clusters and the claim — the model can't
- **T** **Transparency** — the spec, thresholds and parameters are visible, not hidden
- **V** **Verification** — you checked against a randomized control

Each session highlights one. Ours is Reproducibility — a spec + script + parameters makes a fully re-runnable record