Workshop on R and movement ecology:

Hong Kong University, Jan 2018

Eric Dougherty, Dana Seidel, Wayne Getz

Lecture 4

Space-time considerations

DEPARTMENT *of* ENVIRONMENTAL SCIENCE, POLICY, AND MANAGEMENT

Berkeley UNIVERSITY OF CALIFORNIA College of Natural Resources

Movement Ecology and Space-Time

δ

Movement Ecology and Space-Time

- space is always explicit — typically 2D-Euclidean

δ

Movement Ecology and Space-Time

- space is always explicit — typically 2D-Euclidean
- constructions from spatially explicit trajectories
  - home ranges (HRs)
  - utilization distributions (UDs)
  - resource selection functions (RSFs)
  - step selection functions (SSFs)

δ

Movement Ecology and Space-Time

- space is always explicit — typically 2D-Euclidean
- constructions from spatially explicit trajectories
  - home ranges (HRs)
  - utilization distributions (UDs)
  - resource selection functions (RSFs)
  - step selection functions (SSFs)
- grouping spatial locations under chunked time
  - constant
  - chunked into seasons (wet/dry, spring/summer/fall/winter, other)
  - diurnal stratifications (night/day, 4 equal centered around midday, other)

δ

Movement Ecology and Space-Time

- space is always explicit — typically 2D-Euclidean
- constructions from spatially explicit trajectories
  - home ranges (HRs)
  - utilization distributions (UDs)
  - resource selection functions (RSFs)
  - step selection functions (SSFs)
- grouping spatial locations under chunked time
  - constant
  - chunked into seasons (wet/dry, spring/summer/fall/winter, other)
  - diurnal stratifications (night/day, 4 equal centered around midday, other)
- time is explicit for point locations
  - points are in 2D space X 1D time volume
  - velocity transformed distance (velocity *v*):

δ

Movement Ecology and Space-Time

- space is always explicit — typically 2D-Euclidean
- constructions from spatially explicit trajectories
  - home ranges (HRs)
  - utilization distributions (UDs)
  - resource selection functions (RSFs)
  - step selection functions (SSFs)
- grouping spatial locations under chunked time
  - constant
  - chunked into seasons (wet/dry, spring/summer/fall/winter, other)
  - diurnal stratifications (night/day, 4 equal centered around midday, other)
- time is explicit for point locations
  - points are in 2D space X 1D time volume
  - velocity transformed distance (velocity *v*):
  - diffusion transformed distance (diffusion rate δ):

Movement Ecology and Space-Time

- space is always explicit — typically 2D-Euclidean
- constructions from spatially explicit trajectories
  - home ranges (HRs)
  - utilization distributions (UDs)
  - resource selection functions (RSFs)
  - step selection functions (SSFs)
- grouping spatial locations under chunked time
  - constant
  - chunked into seasons (wet/dry, spring/summer/fall/winter, other)
  - diurnal stratifications (night/day, 4 equal centered around midday, other)
- time is explicit for point locations
  - points are in 2D space X 1D time volume
  - velocity transformed distance (velocity *v*):

$$d : i \rightarrow j = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + v^2(t_i - t_j)^2}$$

  - diffusion transformed distance (diffusion rate δ):

Movement Ecology and Space-Time

- space is always explicit — typically 2D-Euclidean
- constructions from spatially explicit trajectories
  - home ranges (HRs)
  - utilization distributions (UDs)
  - resource selection functions (RSFs)
  - step selection functions (SSFs)
- grouping spatial locations under chunked time
  - constant
  - chunked into seasons (wet/dry, spring/summer/fall/winter, other)
  - diurnal stratifications (night/day, 4 equal centered around midday, other)
- time is explicit for point locations
  - points are in 2D space X 1D time volume
  - velocity transformed distance (velocity *v*):

$$d : i \rightarrow j = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + v^2(t_i - t_j)^2}$$

  - diffusion transformed distance (diffusion rate δ):

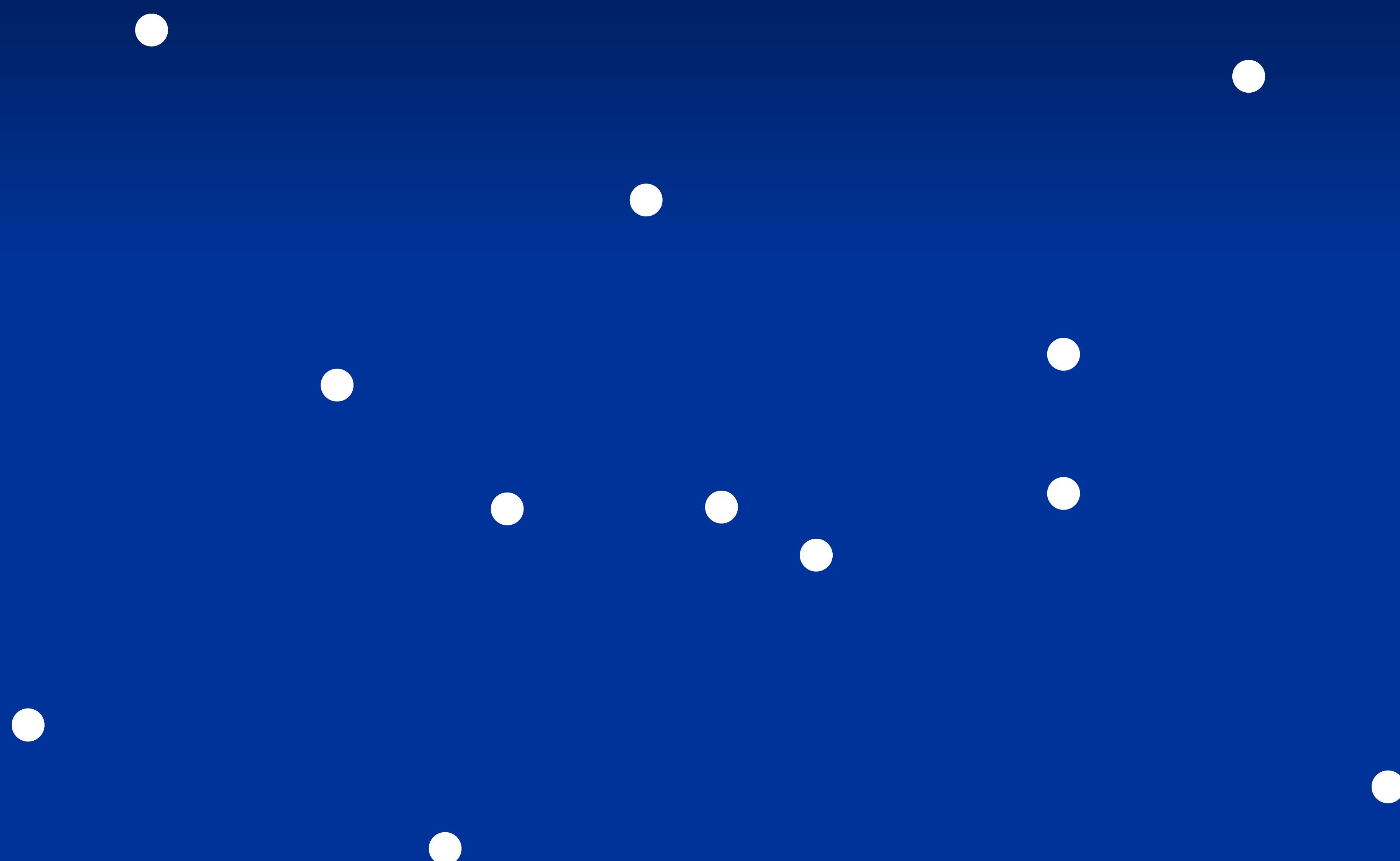
$$d : i \rightarrow j = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \delta^2(t_i - t_j)}$$

Local Convex Hull (LoCoH)
HR and UD construction

Using slides from
a presentation by Andy Lyons

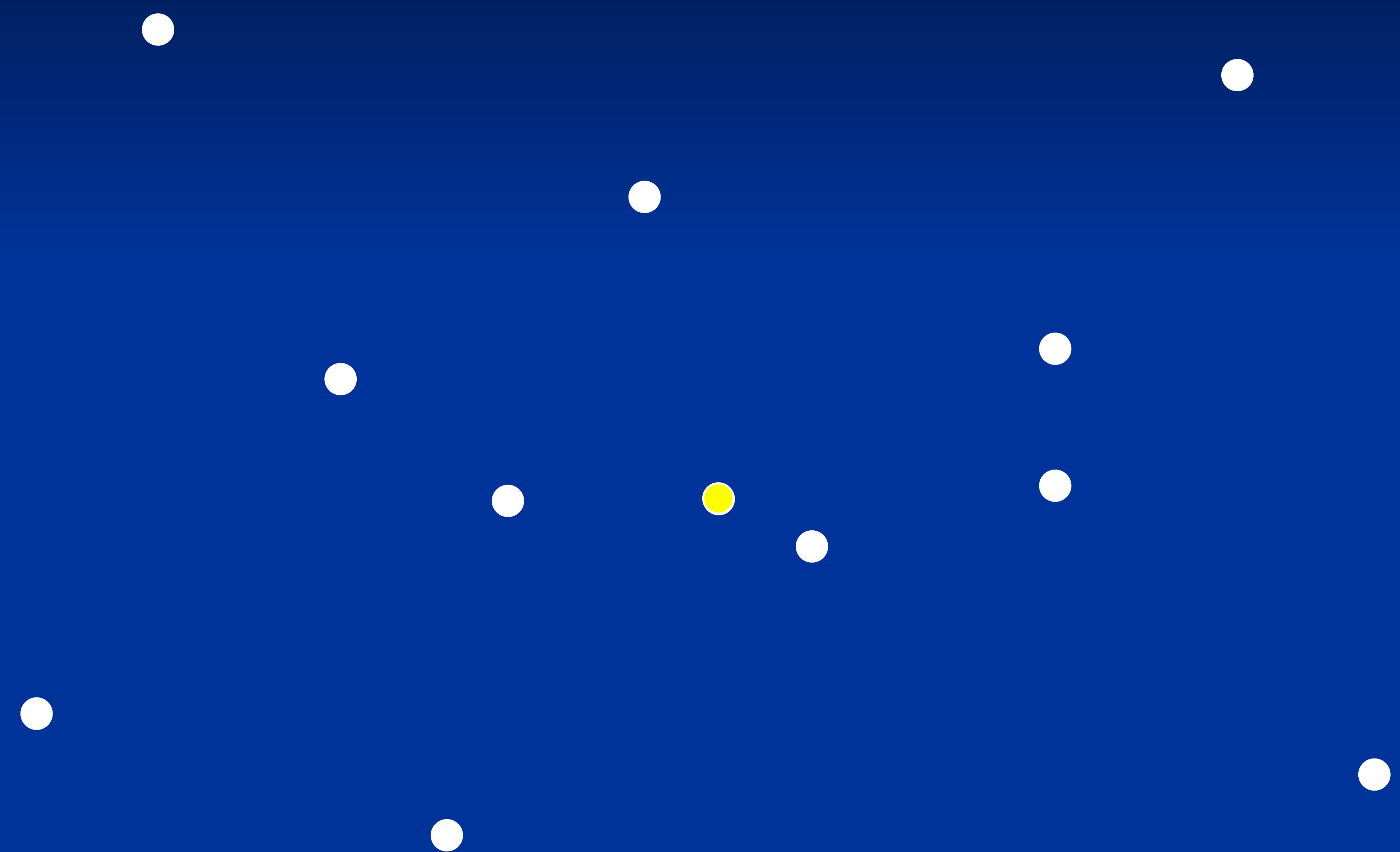
LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze



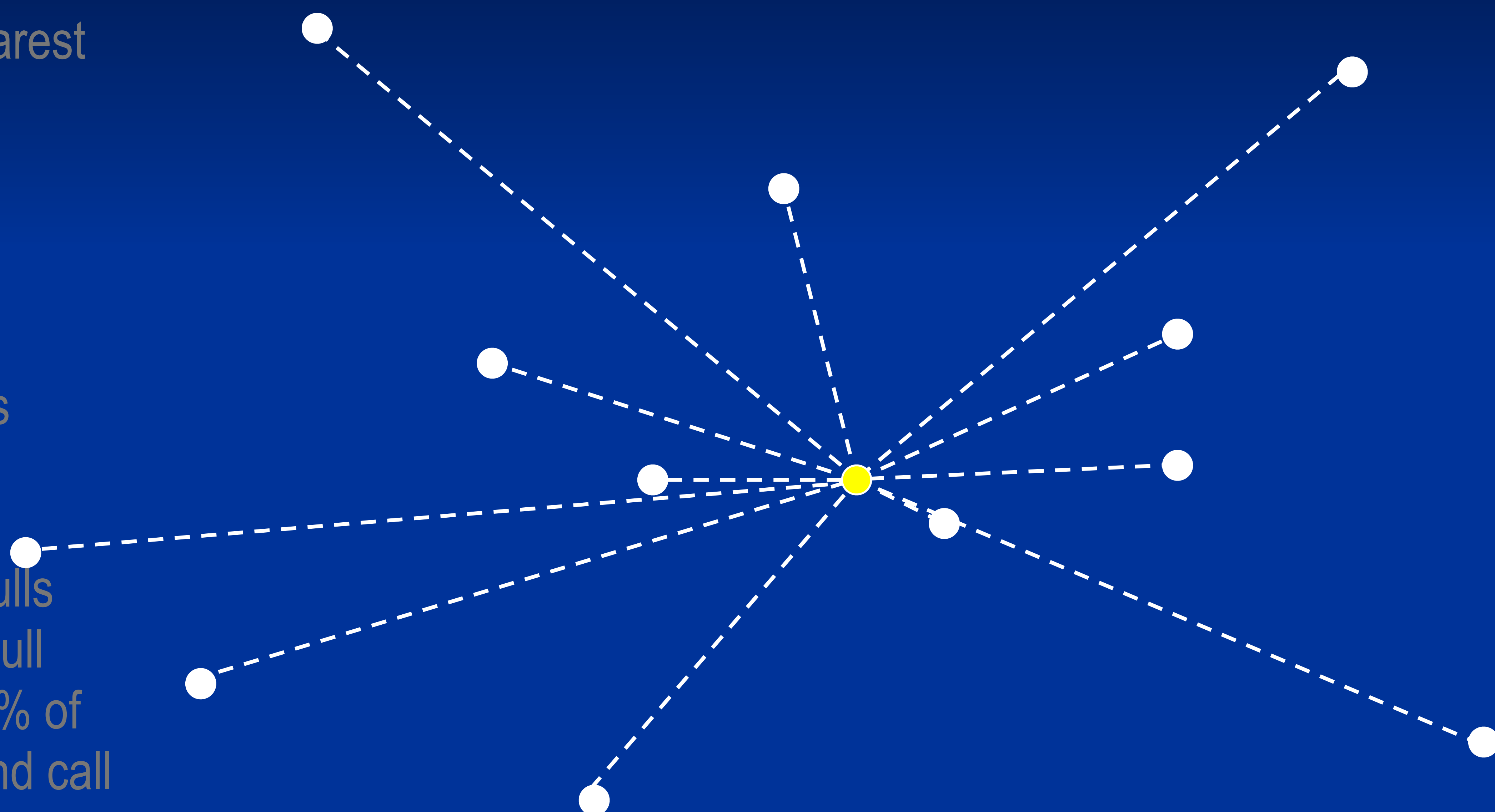
LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze



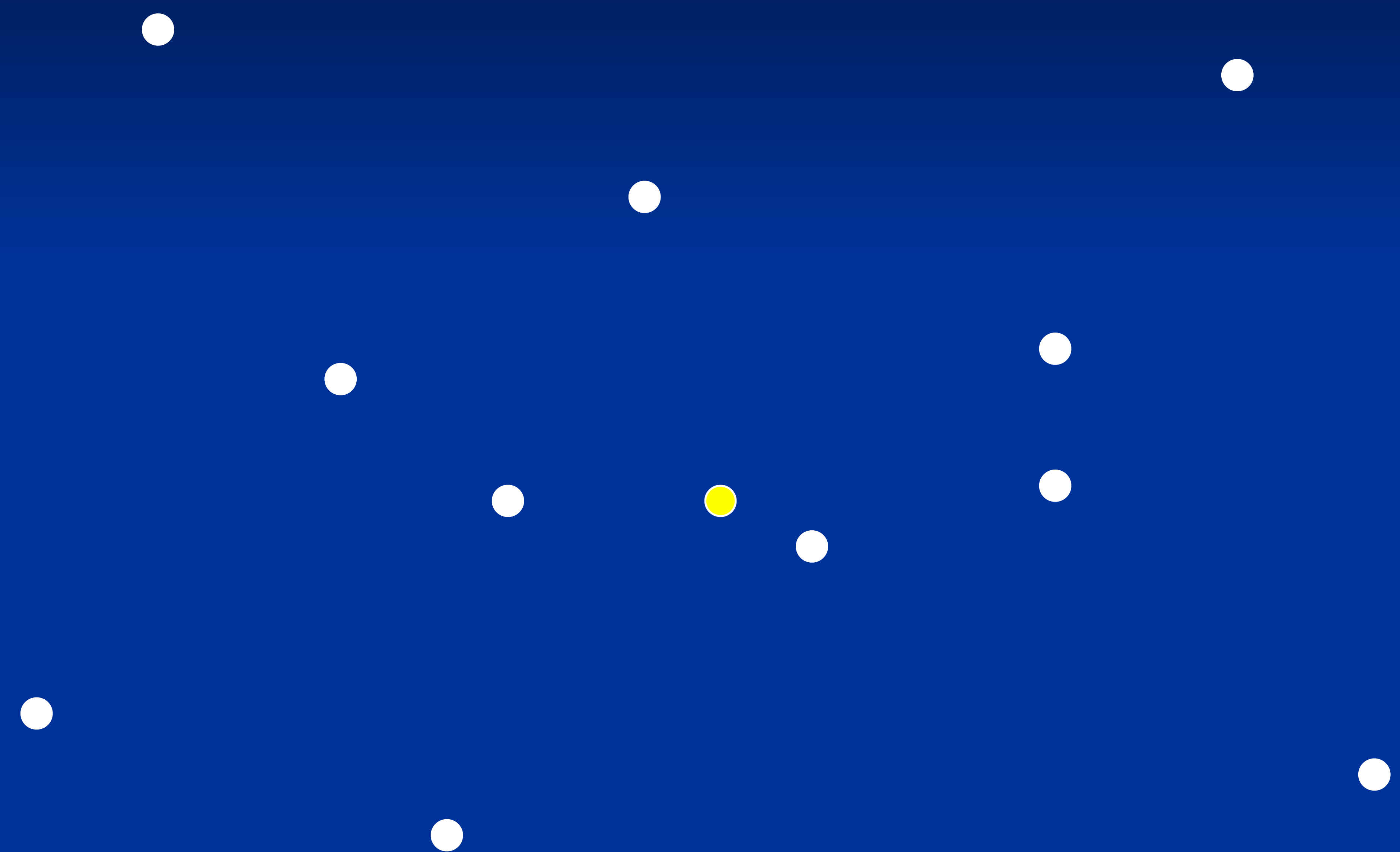
LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze



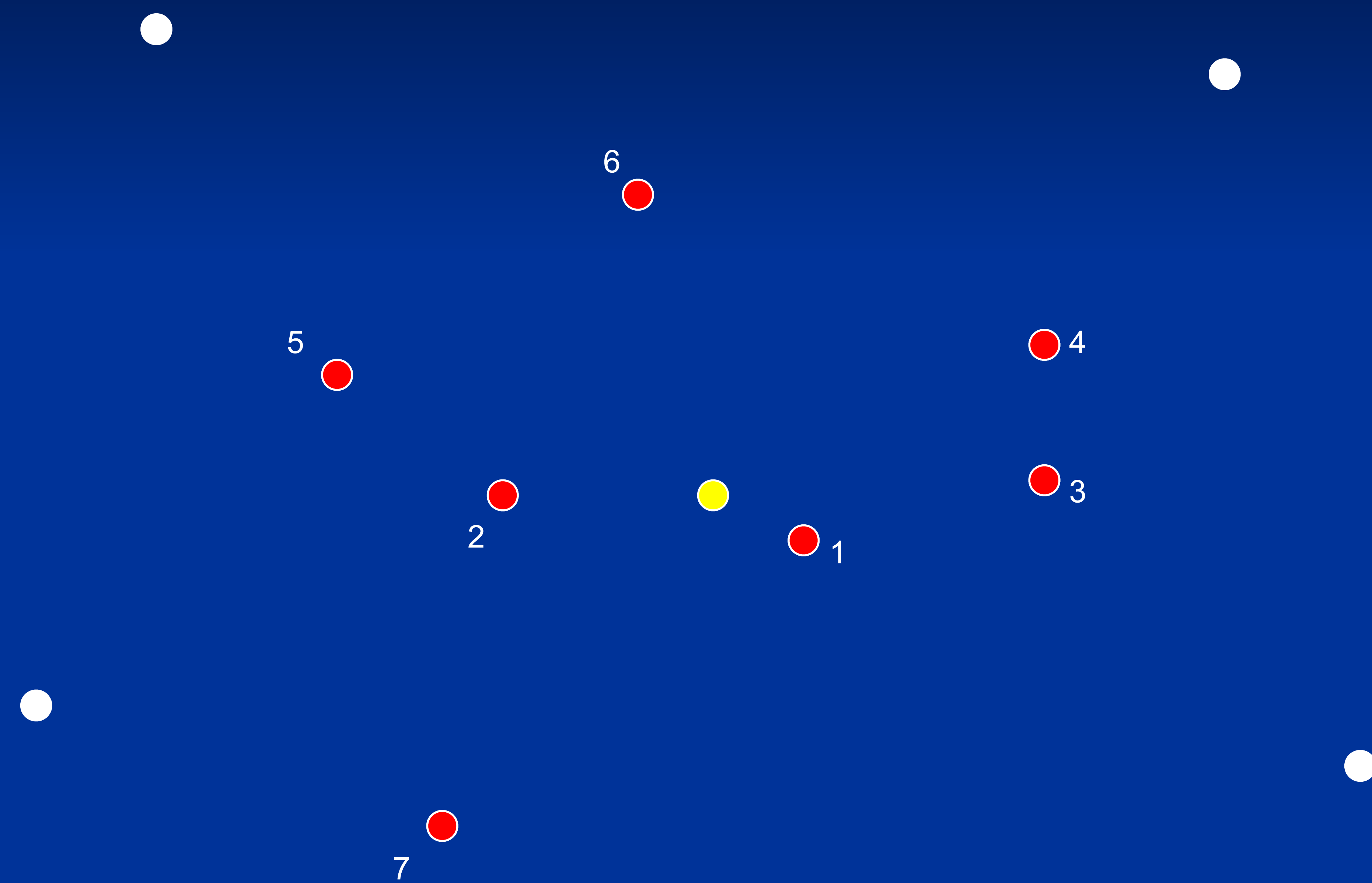
LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
    - k-method
    - r-method
    - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze



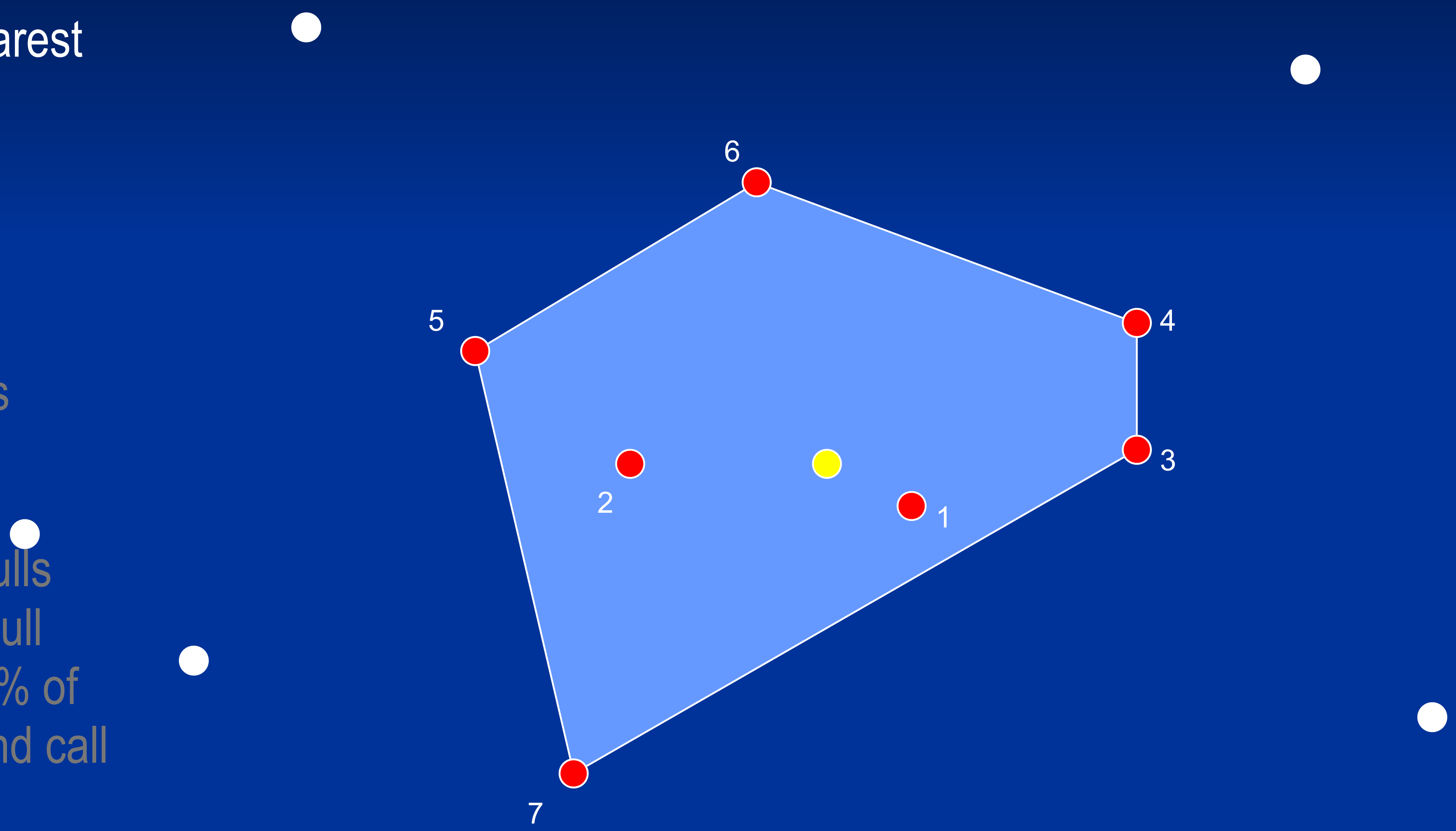
LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze



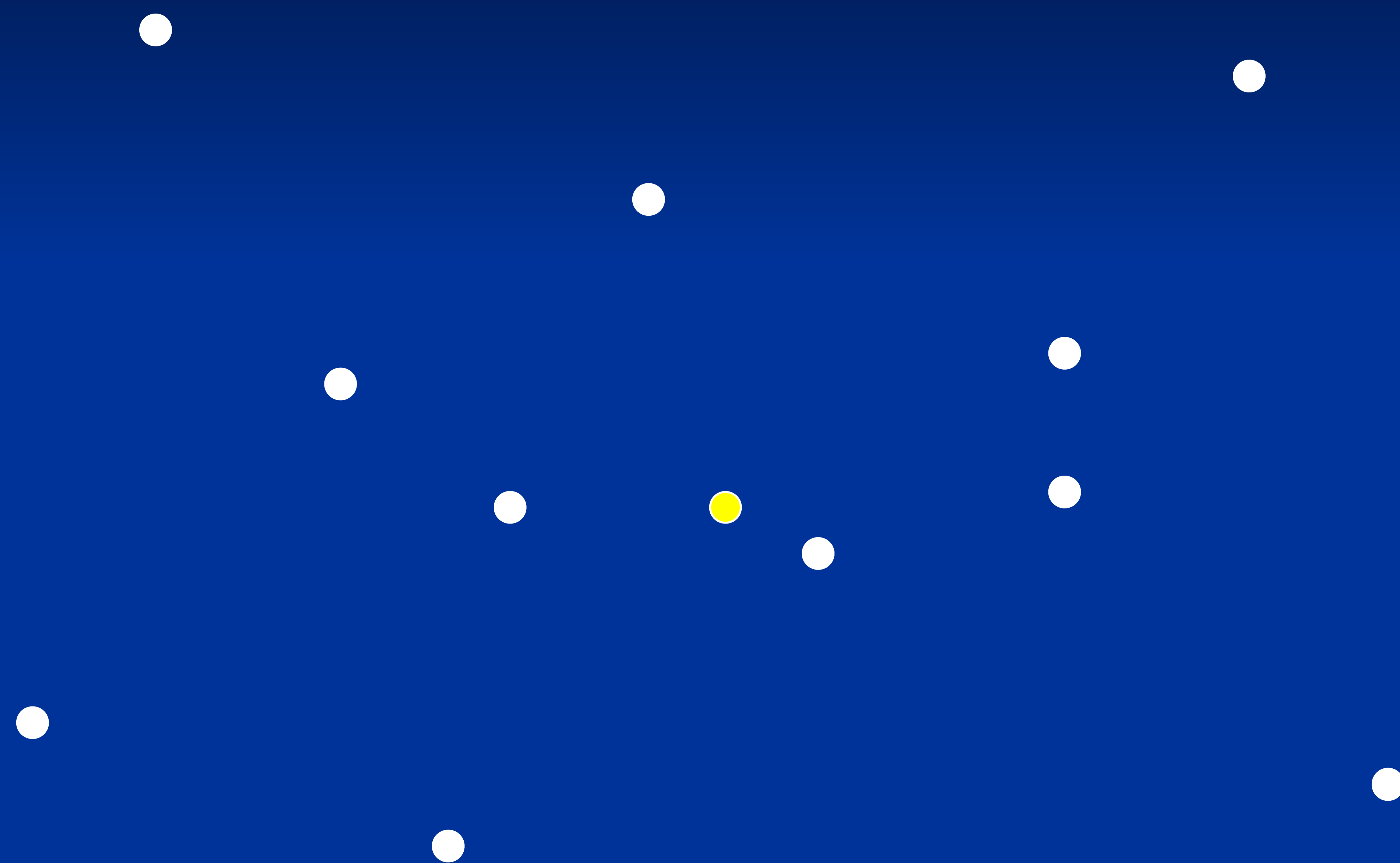
LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze



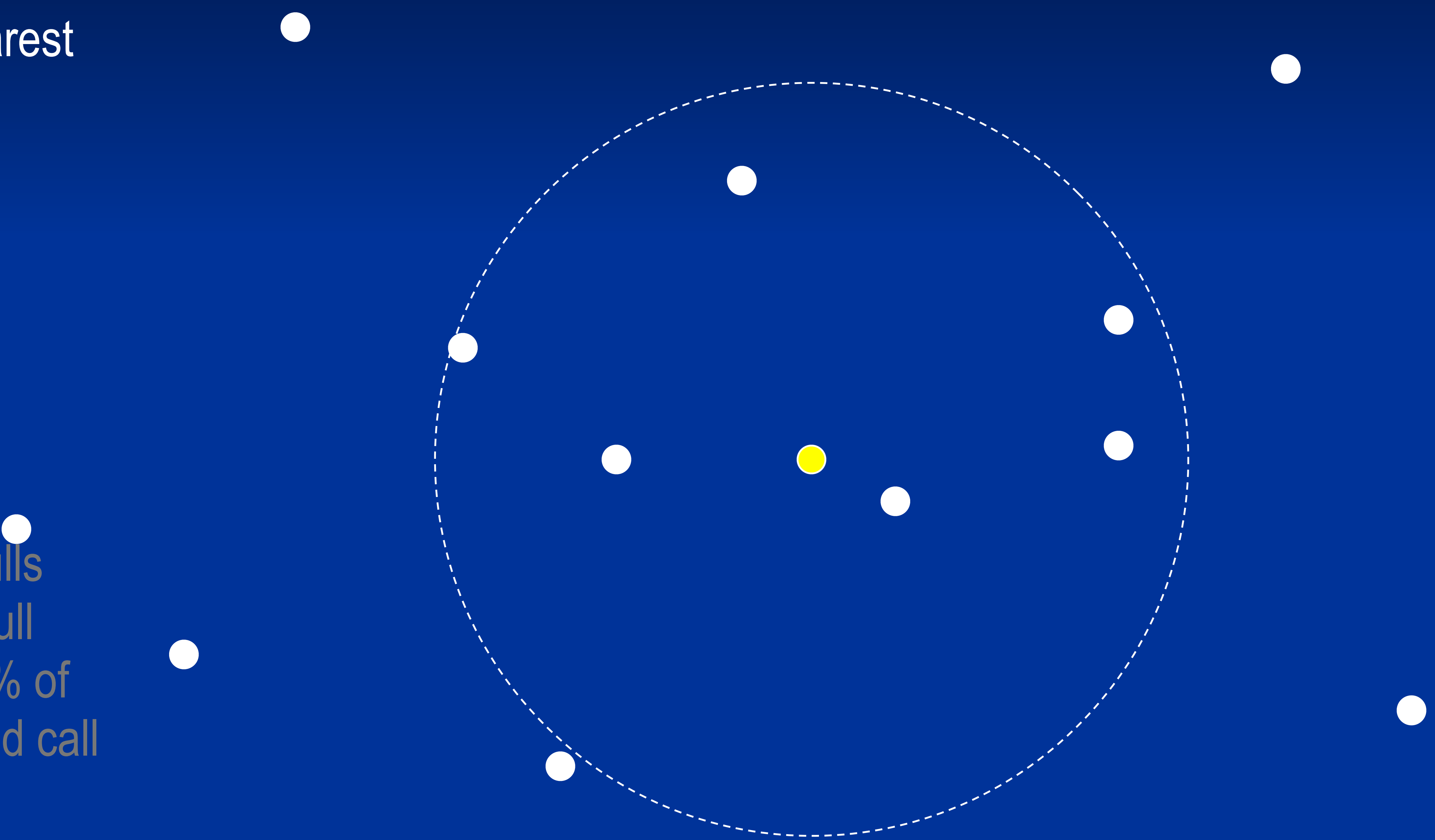
LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
    - k-method
    - r-method
    - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze



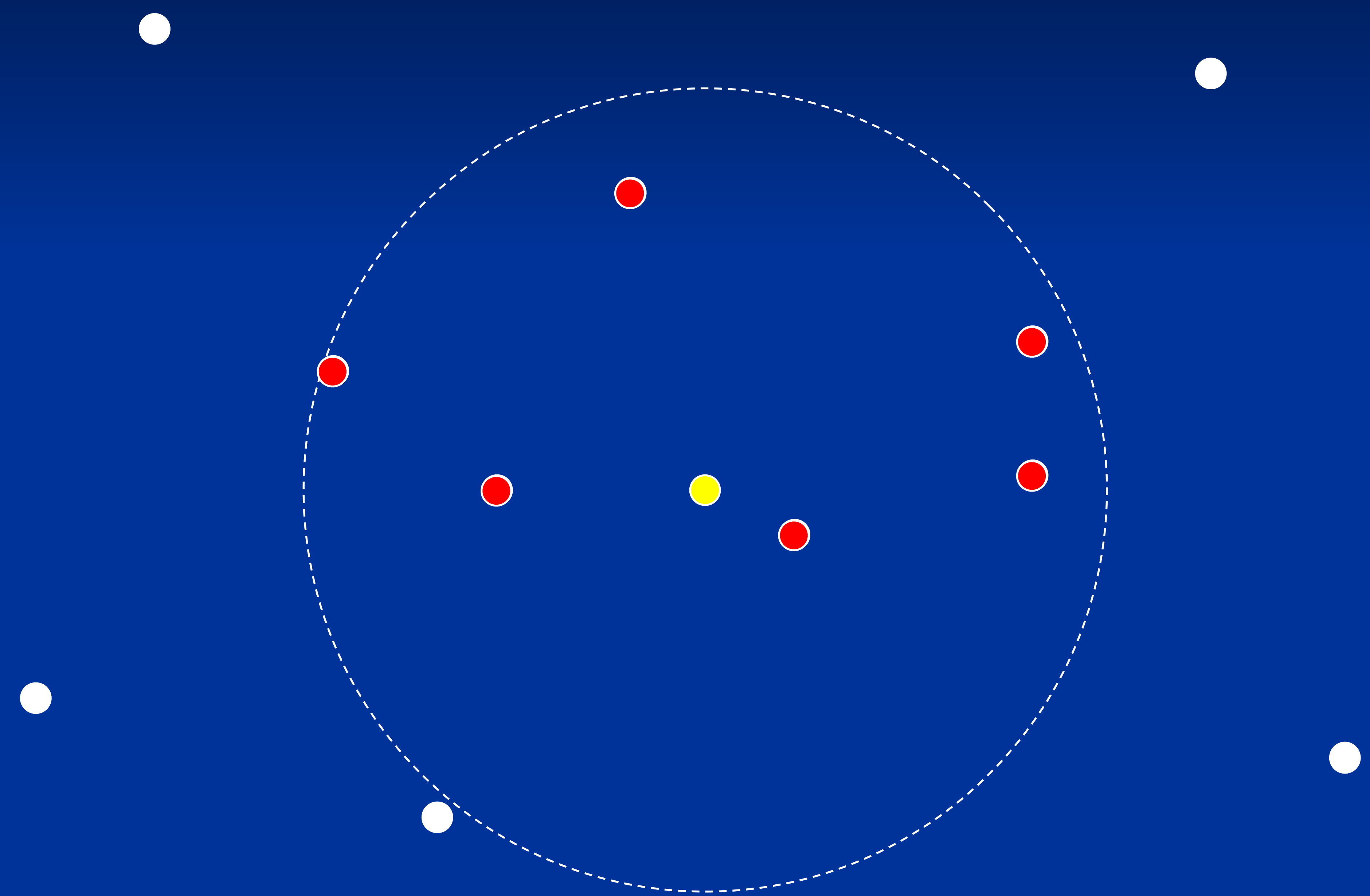
LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze



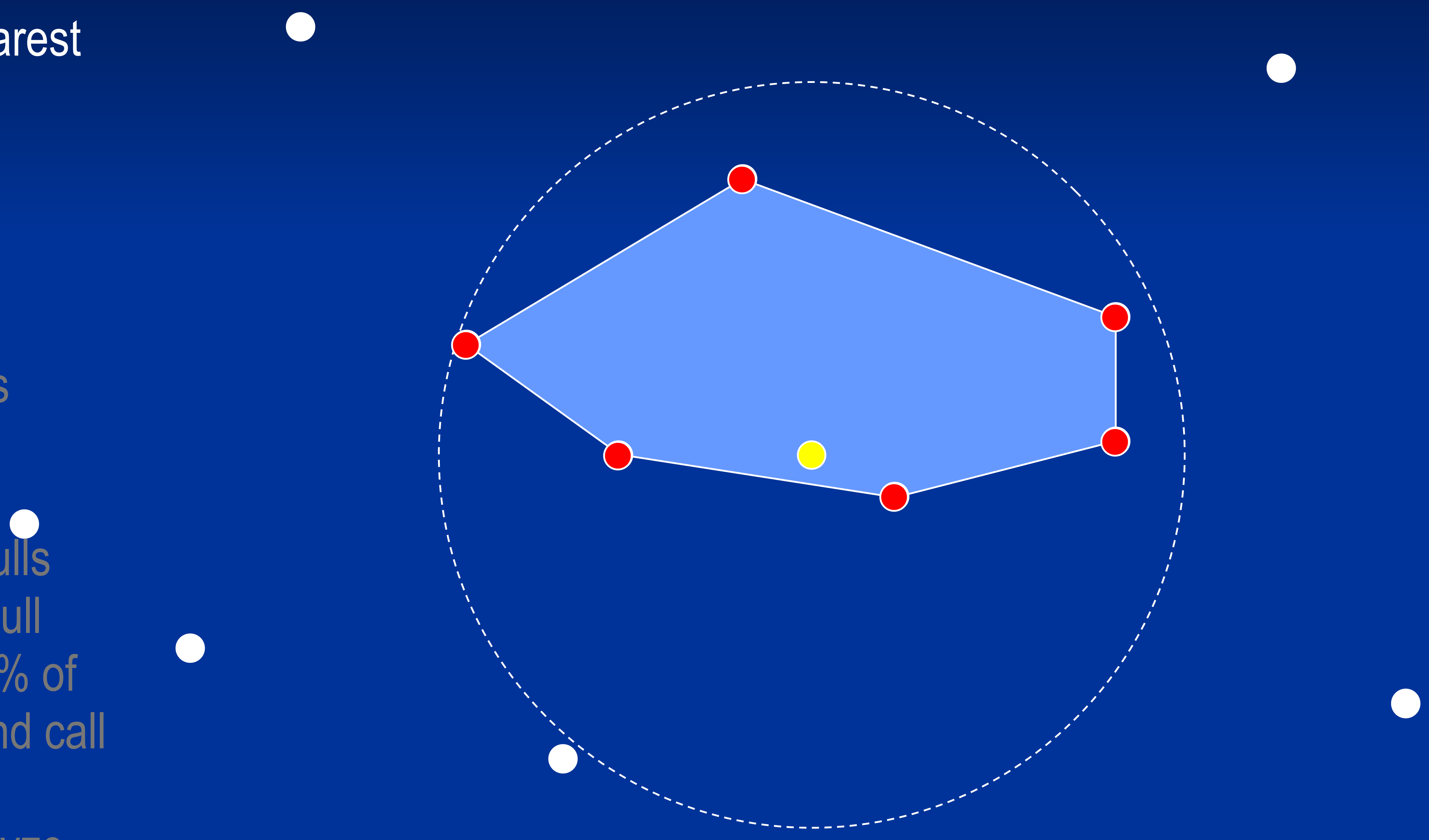
LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
    - k-method
    - r-method
    - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze



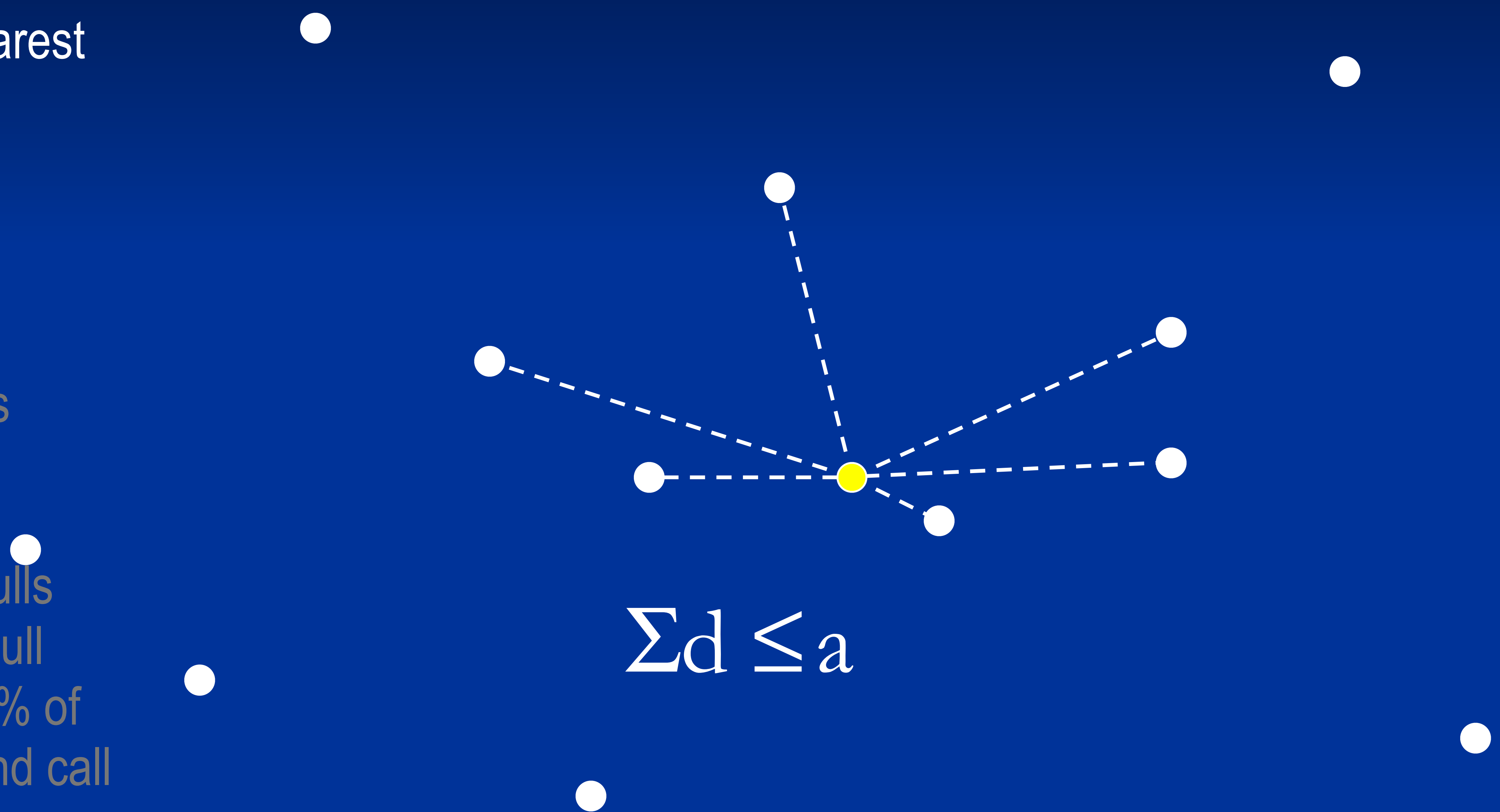
LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
    - k-method
    - r-method
    - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze



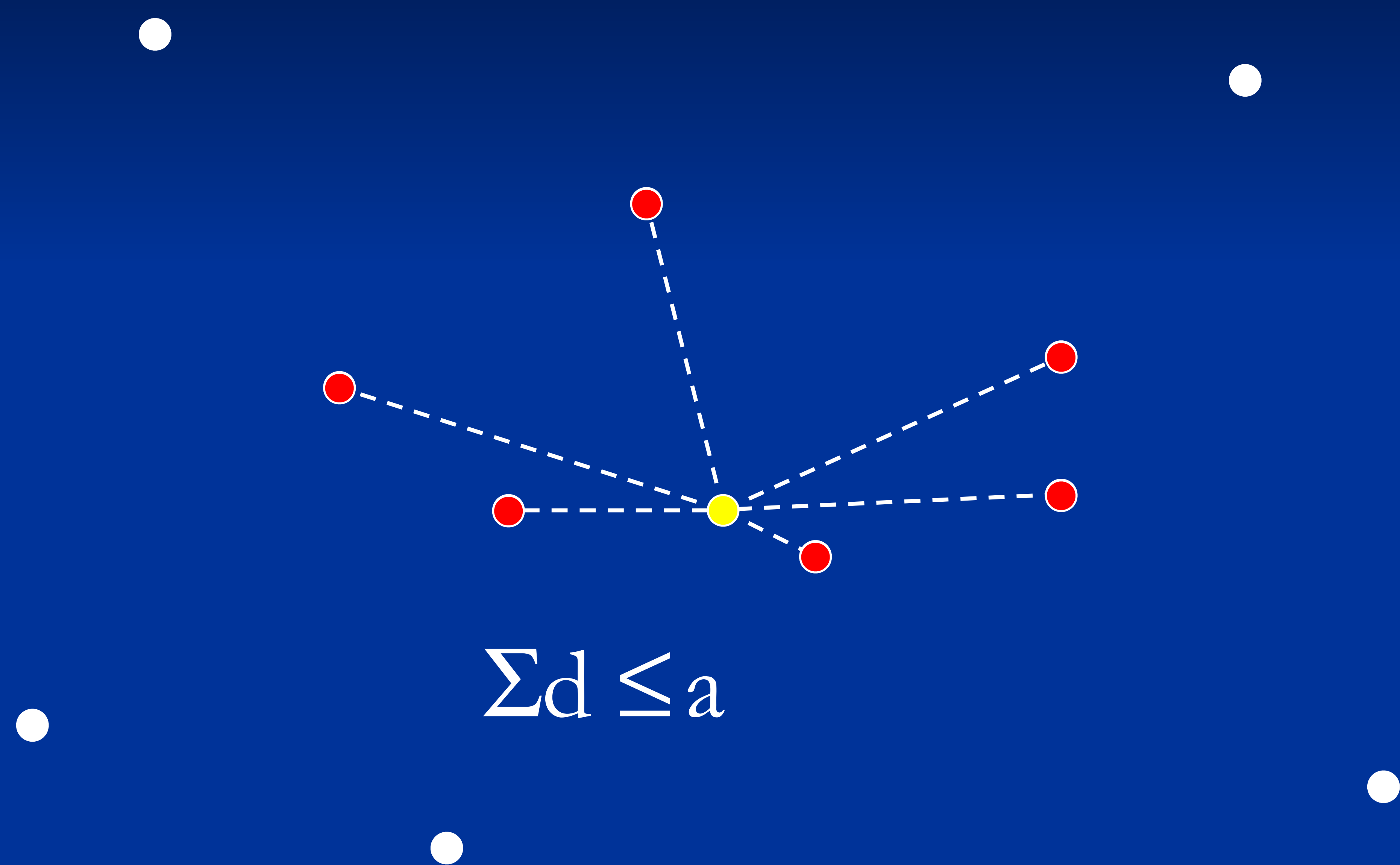
LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
    - k-method
    - r-method
    - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze



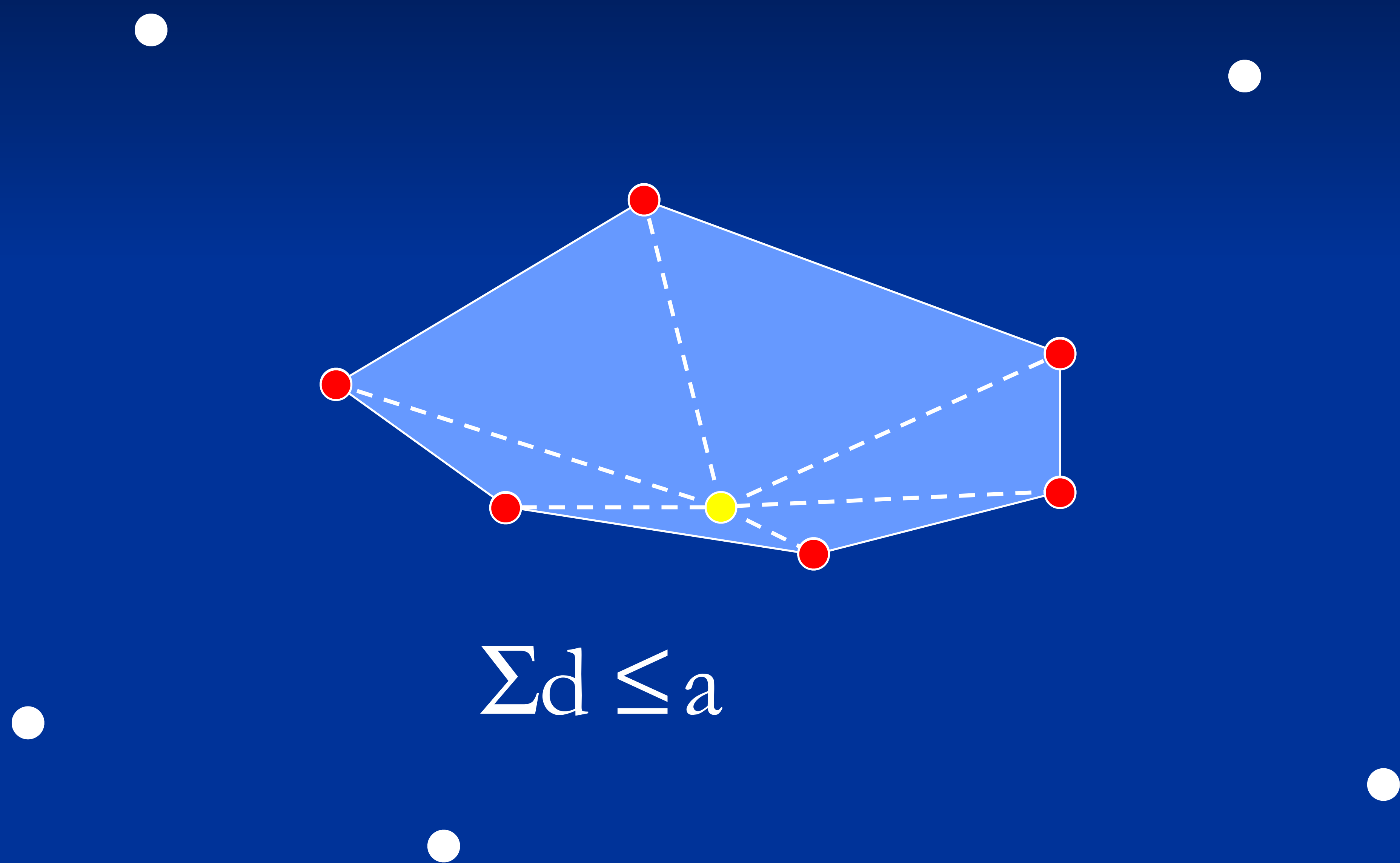
LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze



LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze



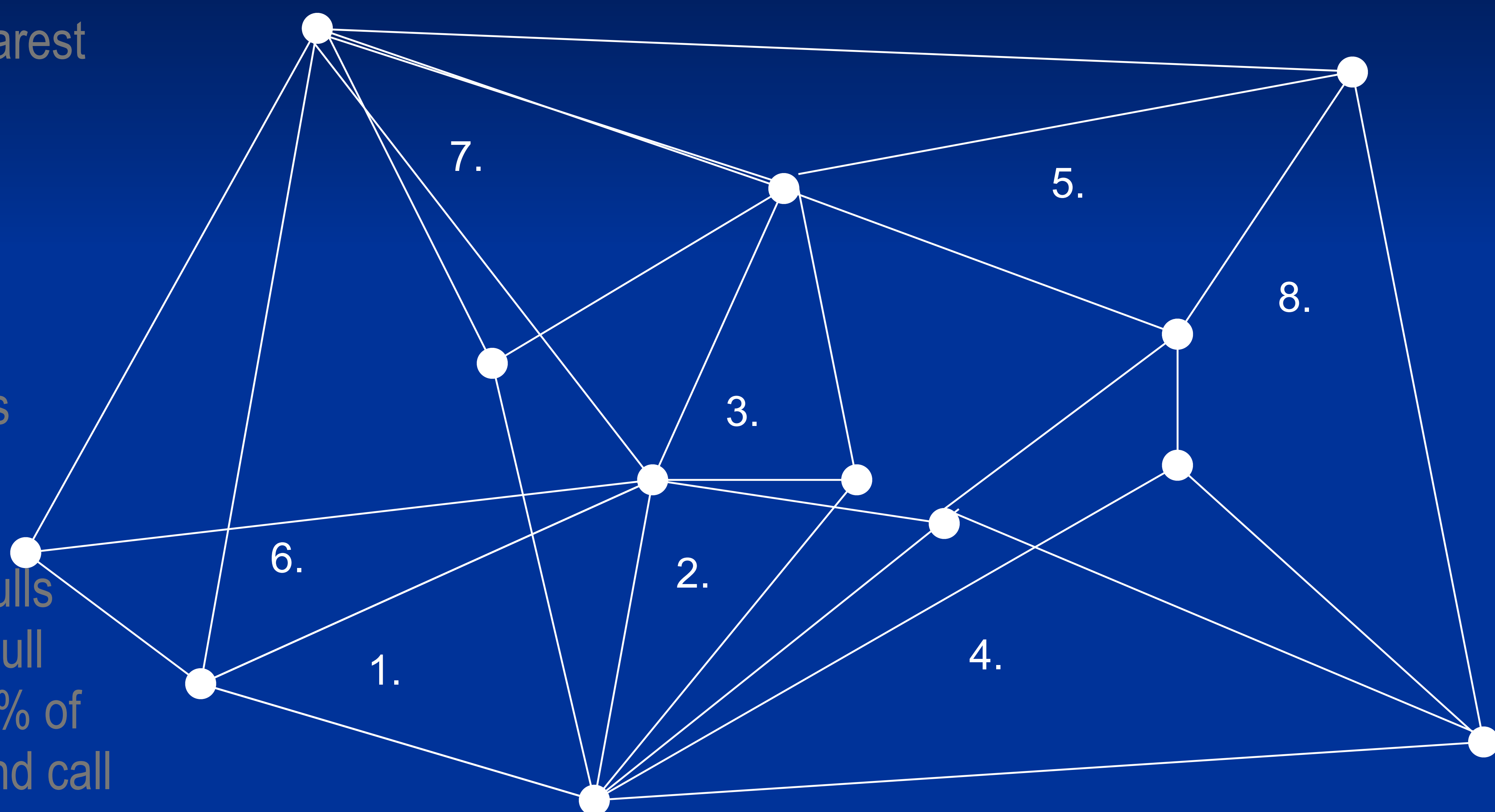
$$\Sigma d \leq a$$

LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze

LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze

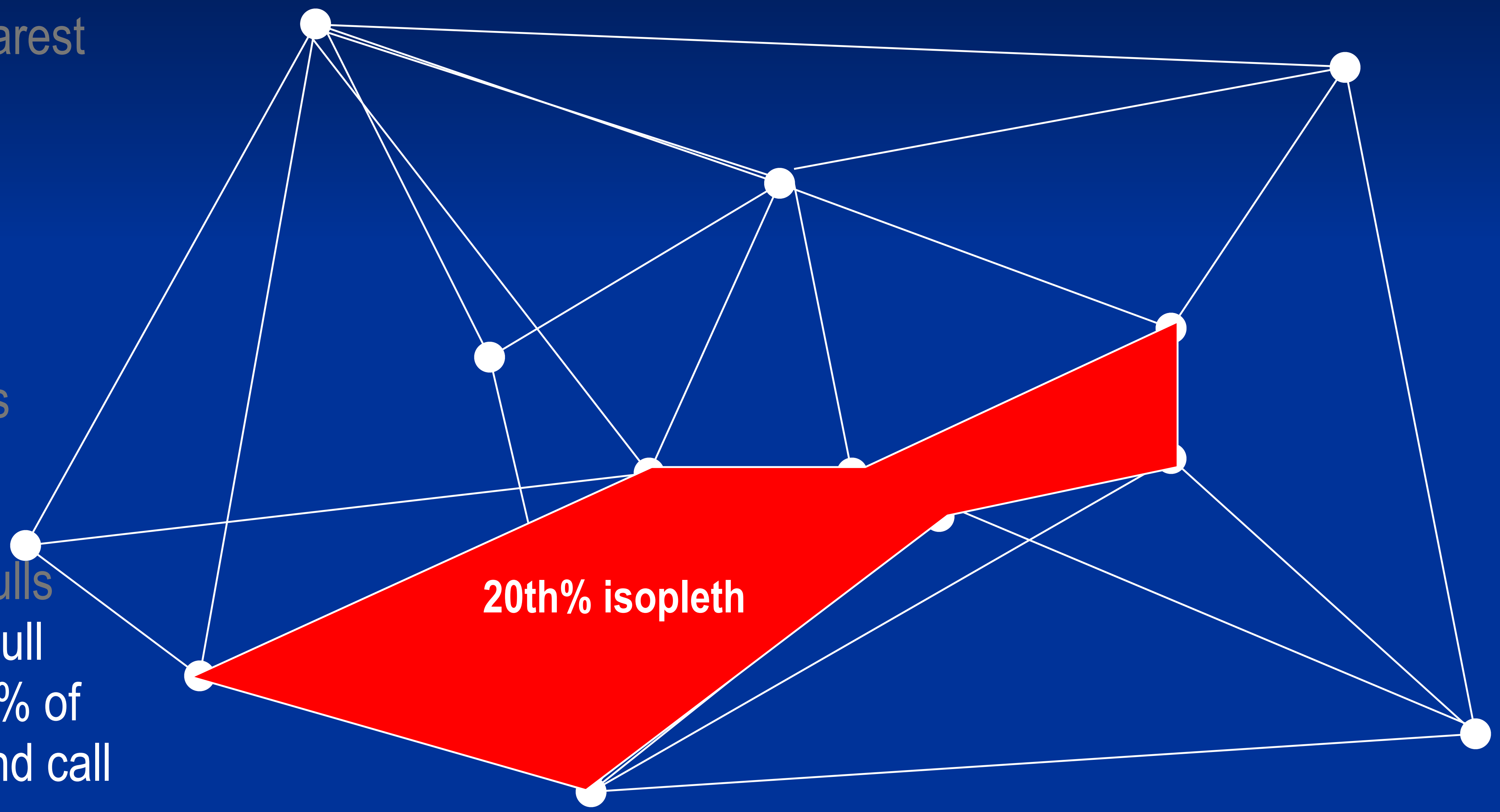


LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze

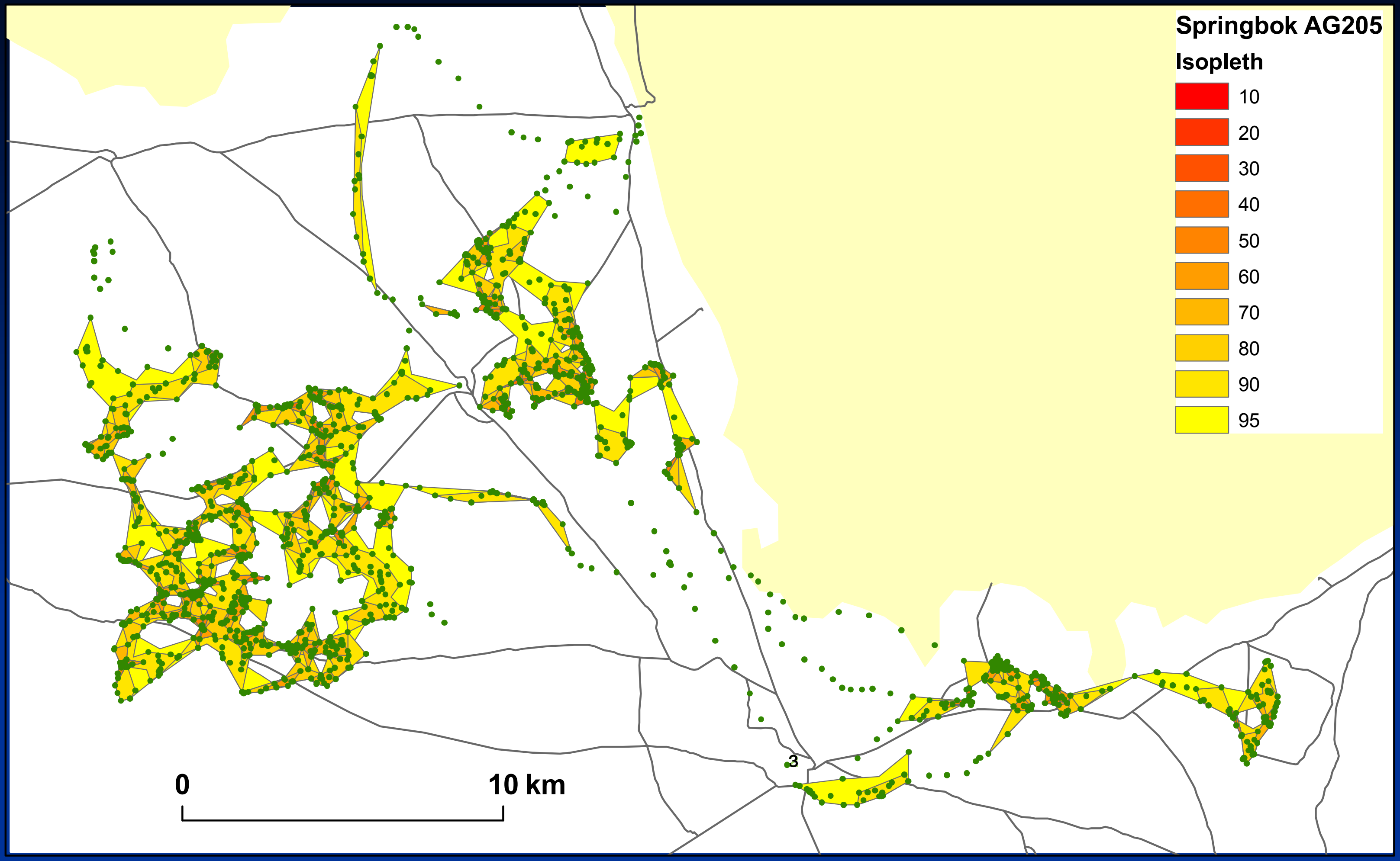
LoCoH Algorithm

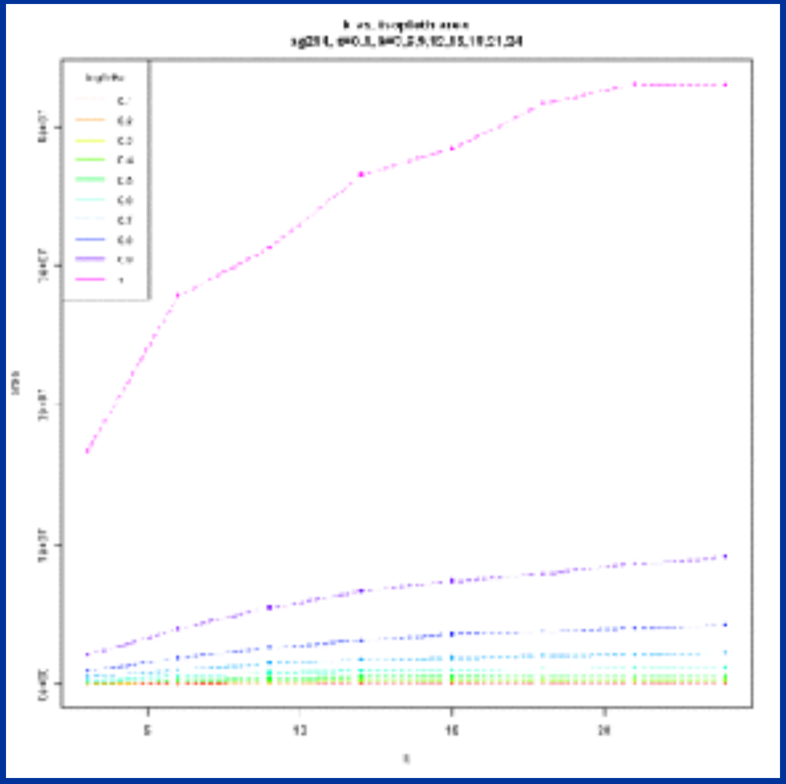
1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze

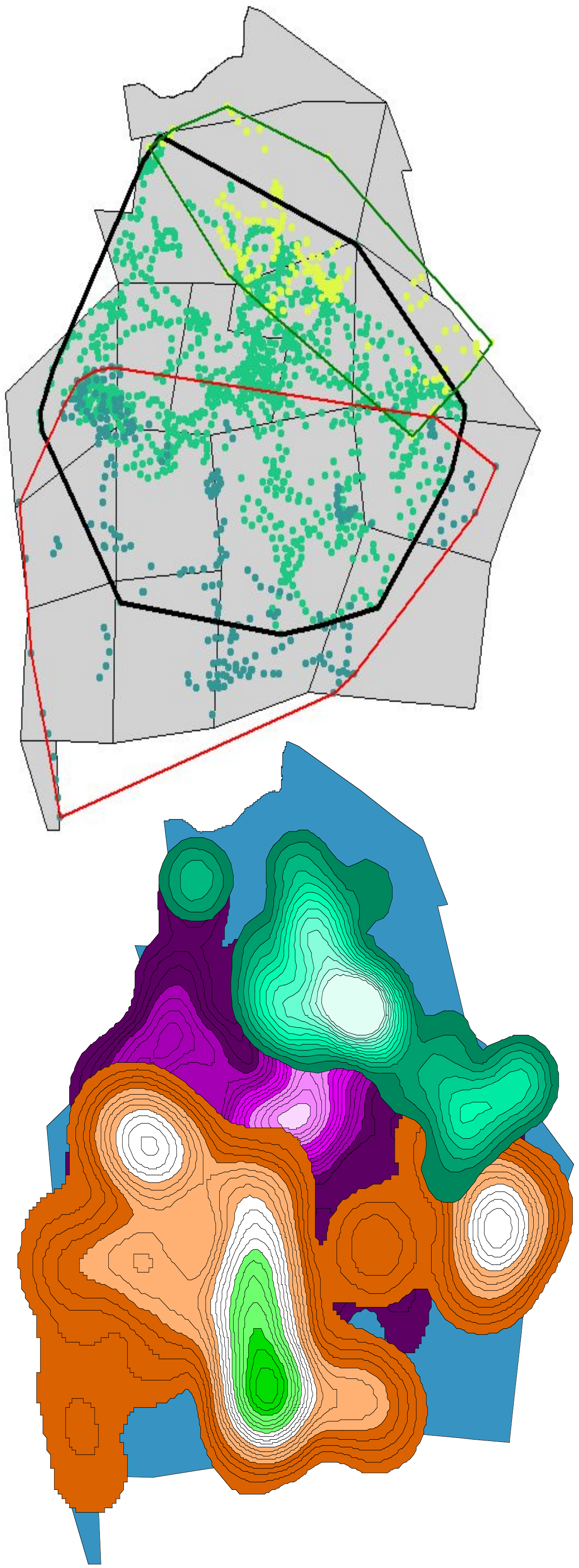


LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze







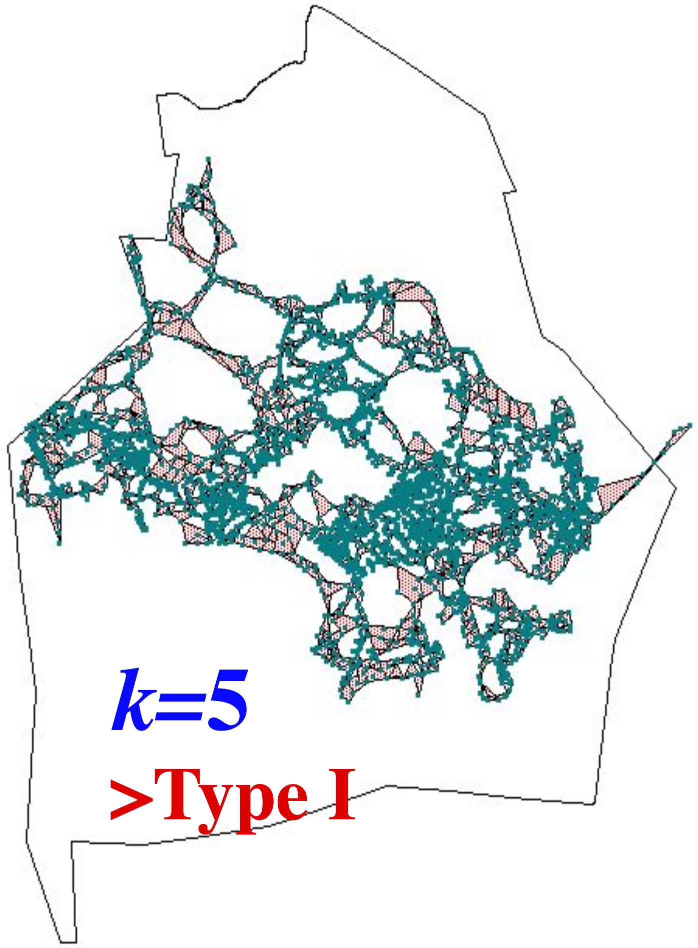
African Buffalo in Klaserie Private Nature Park: data from 3 herds over a season

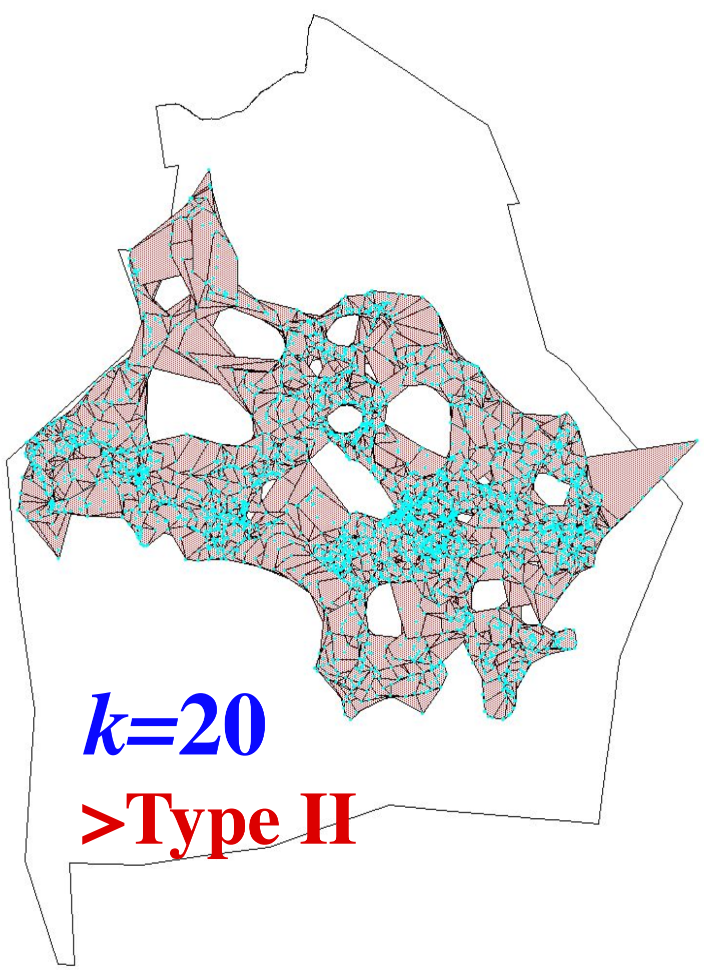
MCP (maximum convex polygon construction)

- simple
- overestimates home range
- ignores densities (no isopleths)

Kernel Methods

- produces isopleths
- smooths irregularities
- ad-hoc boundaries (95 percentile)
- how to choose smoother parameter h





LoCoH Methods

- Local convex hulls: k-1 nearest neighbors of each point
- Take union for home range
- Take progressive unions from smallest to largest *k*-LoCoH to obtain isopleths

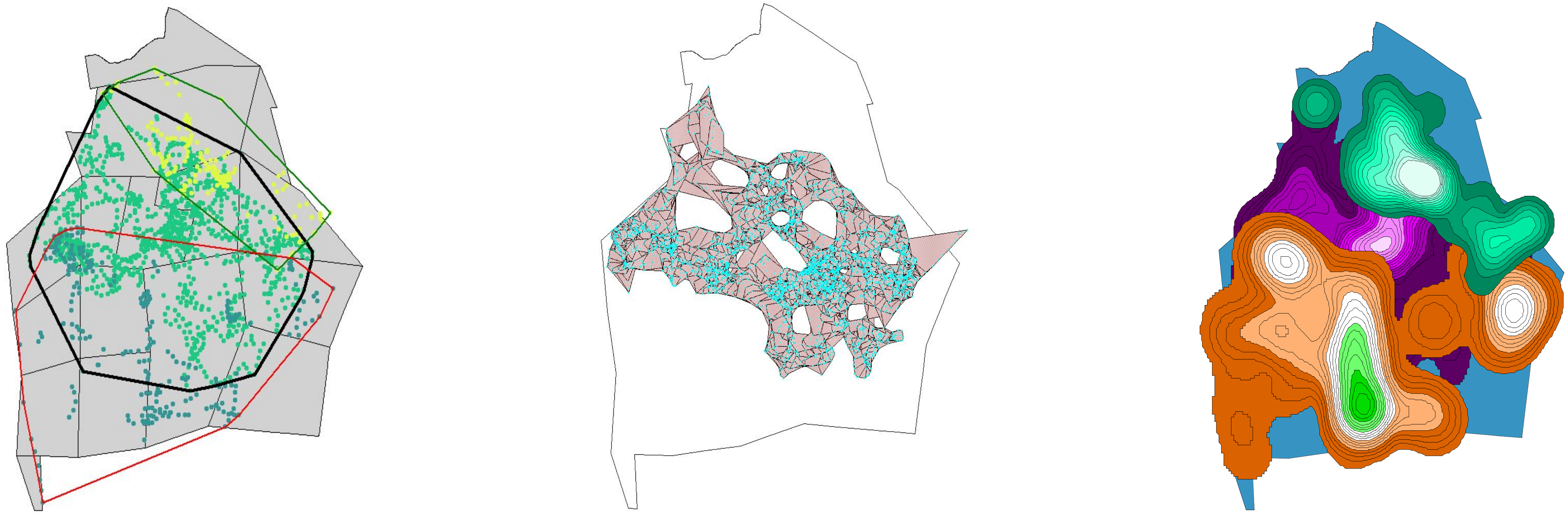
relatively simple

follows irregular data and boundaries

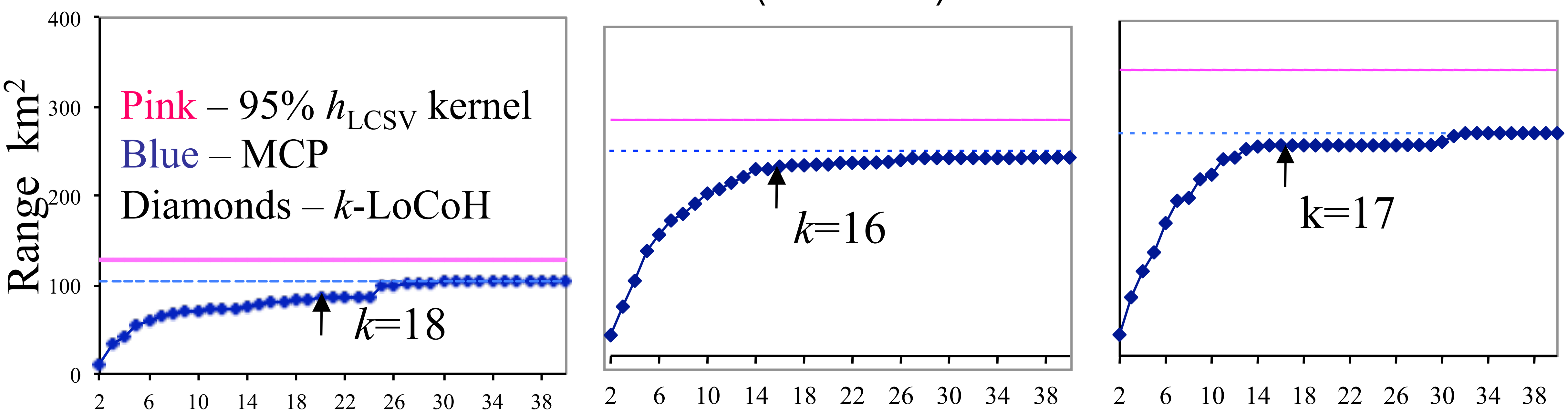
How to choose k?

Type I vs II error trade

Minimum Covering of Spurious Holes



Northern Herd　　Focal (Central) Herd　　Southern Herd

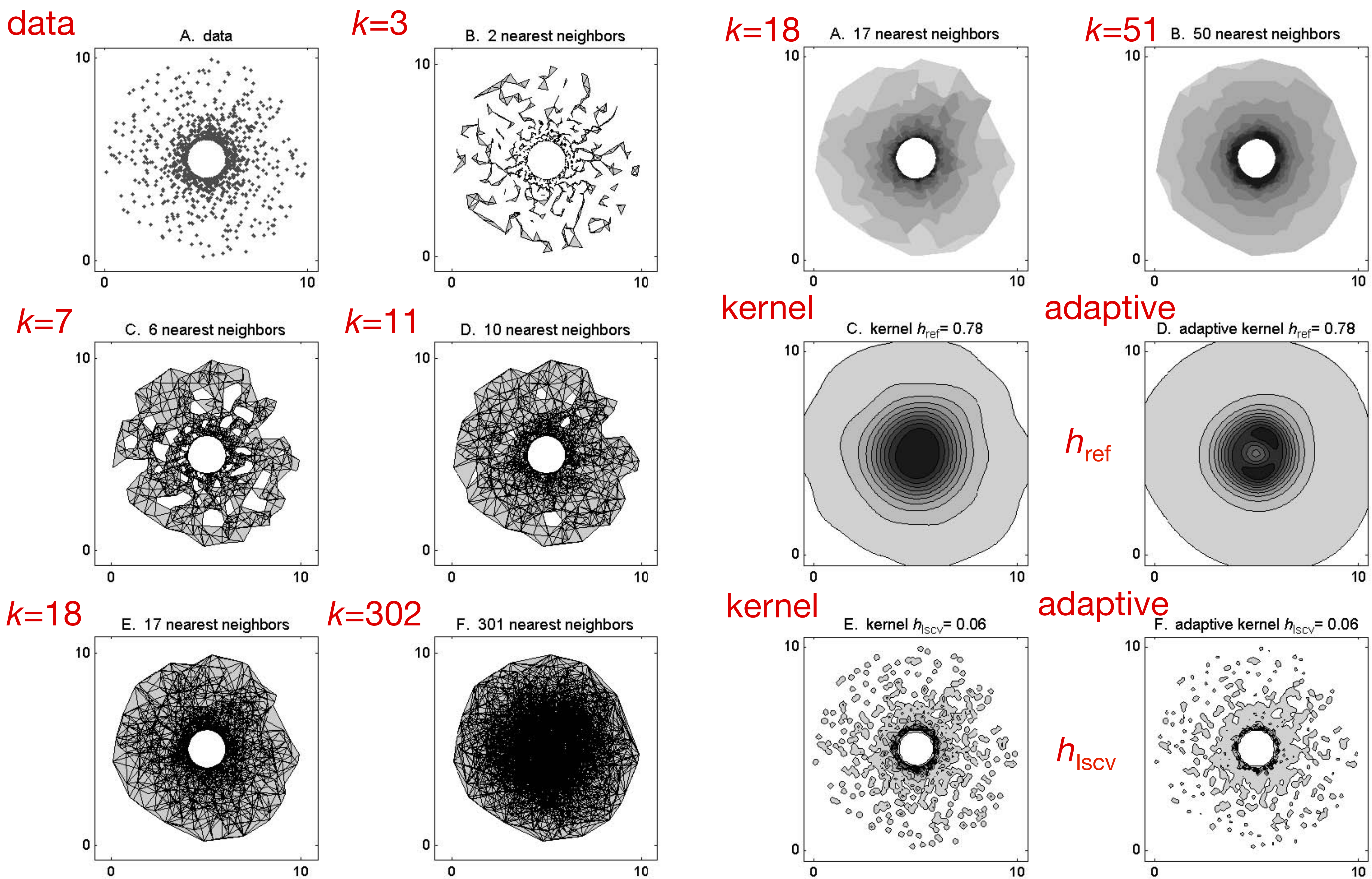


Number of Neighbors in k-LoCoH

Testing our method on simulated data:

- aggregations on boundaries
- holes in the data
- multicore data

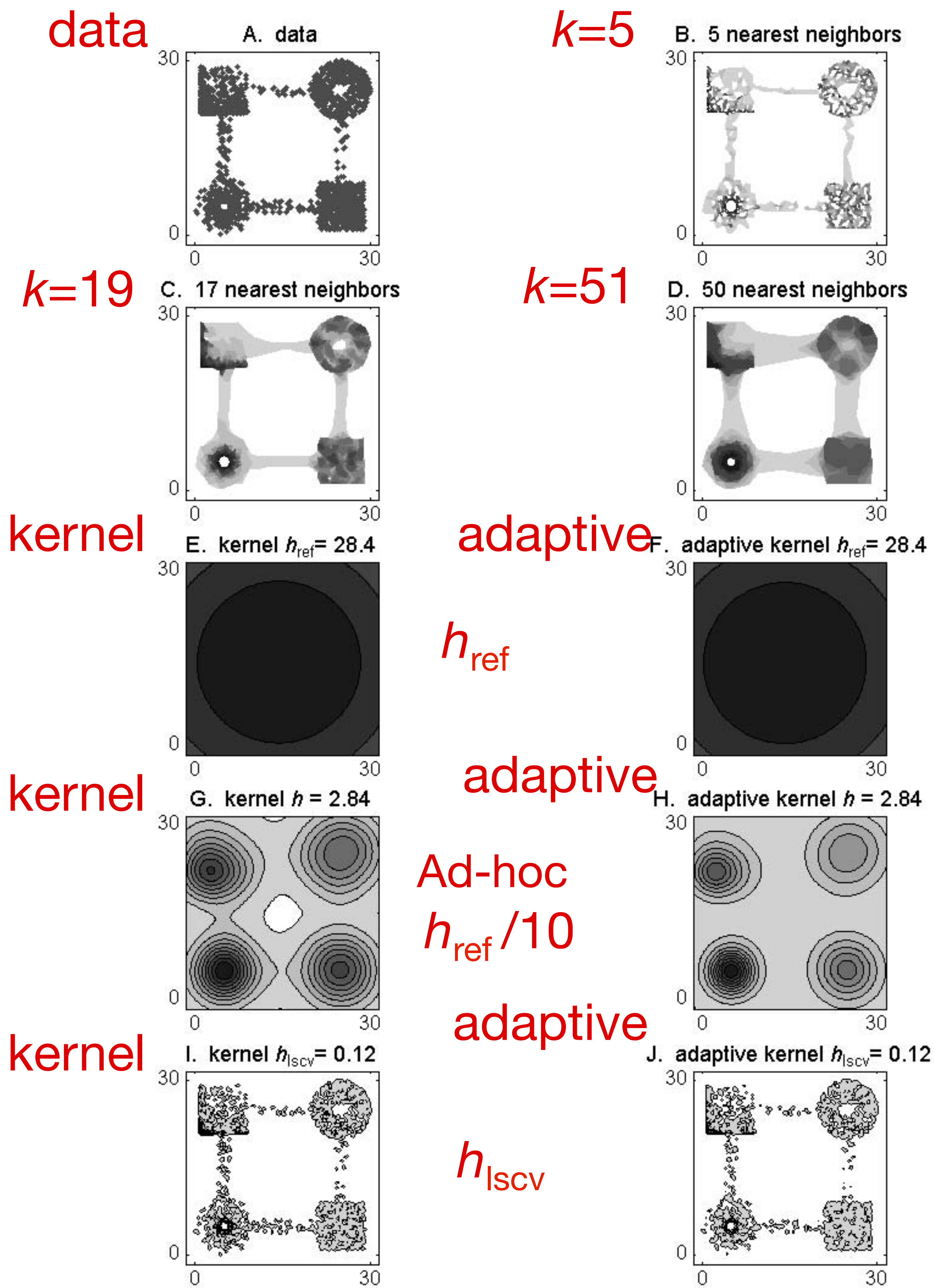
Donut data: constructing UD
k-LoCoH versus Kernel & Adaptive Kernel



Multicore data:

Constructing UD

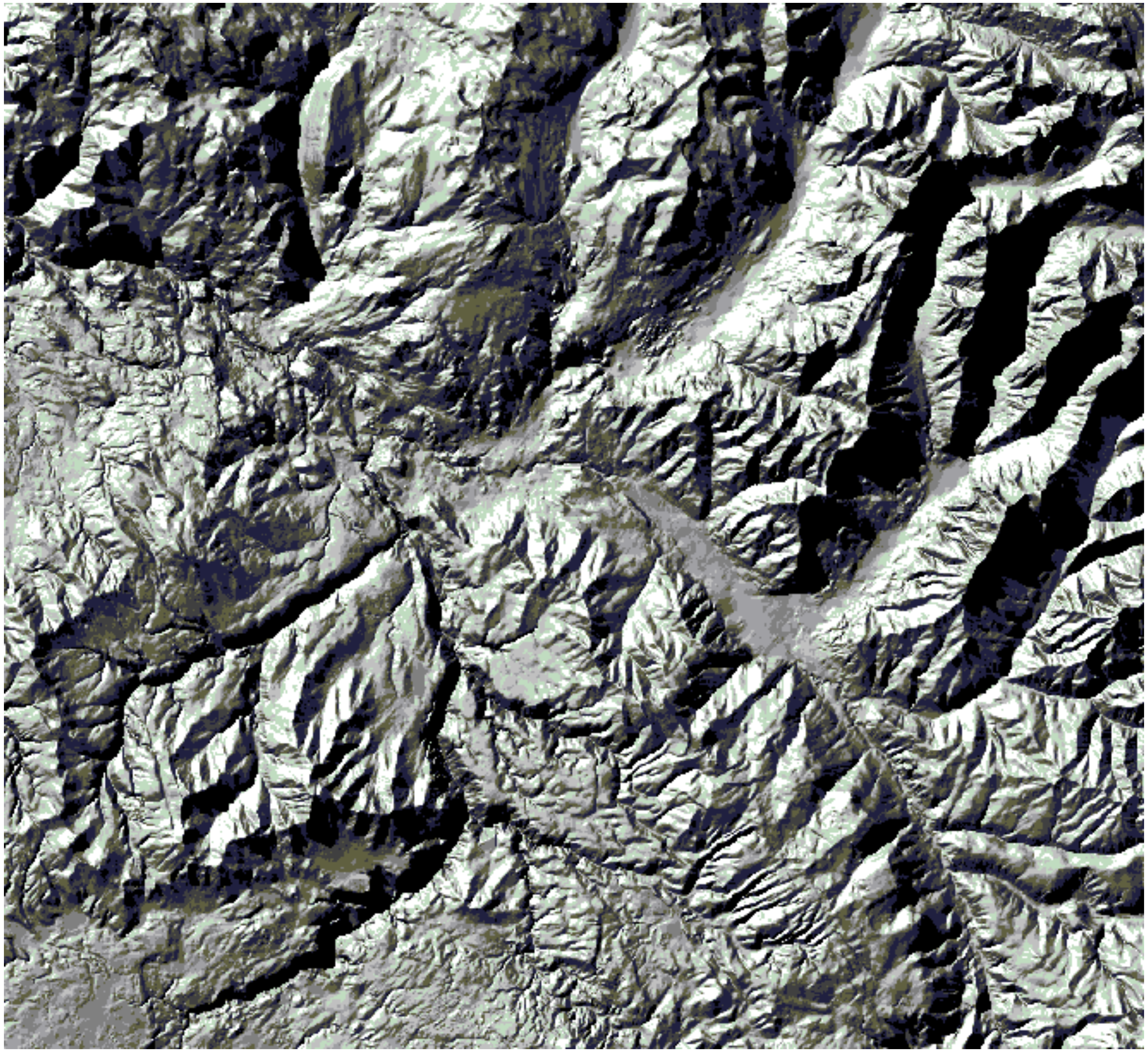
k-LoCoH versus Kernel & Adaptive Kernel



Analysis of Yellowstone wolf data

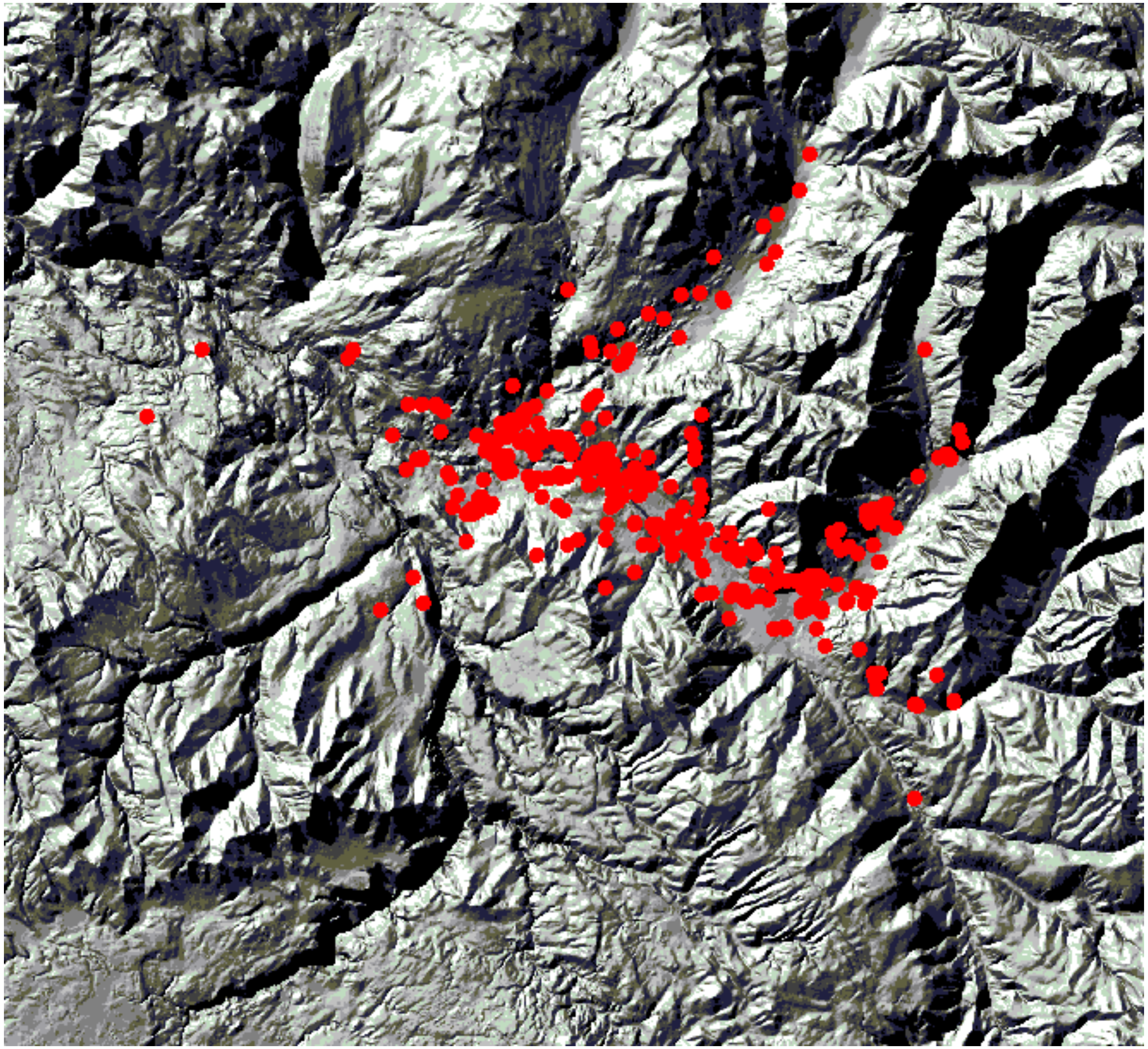
Elk
primary prey

Study area – Yellowstone NP



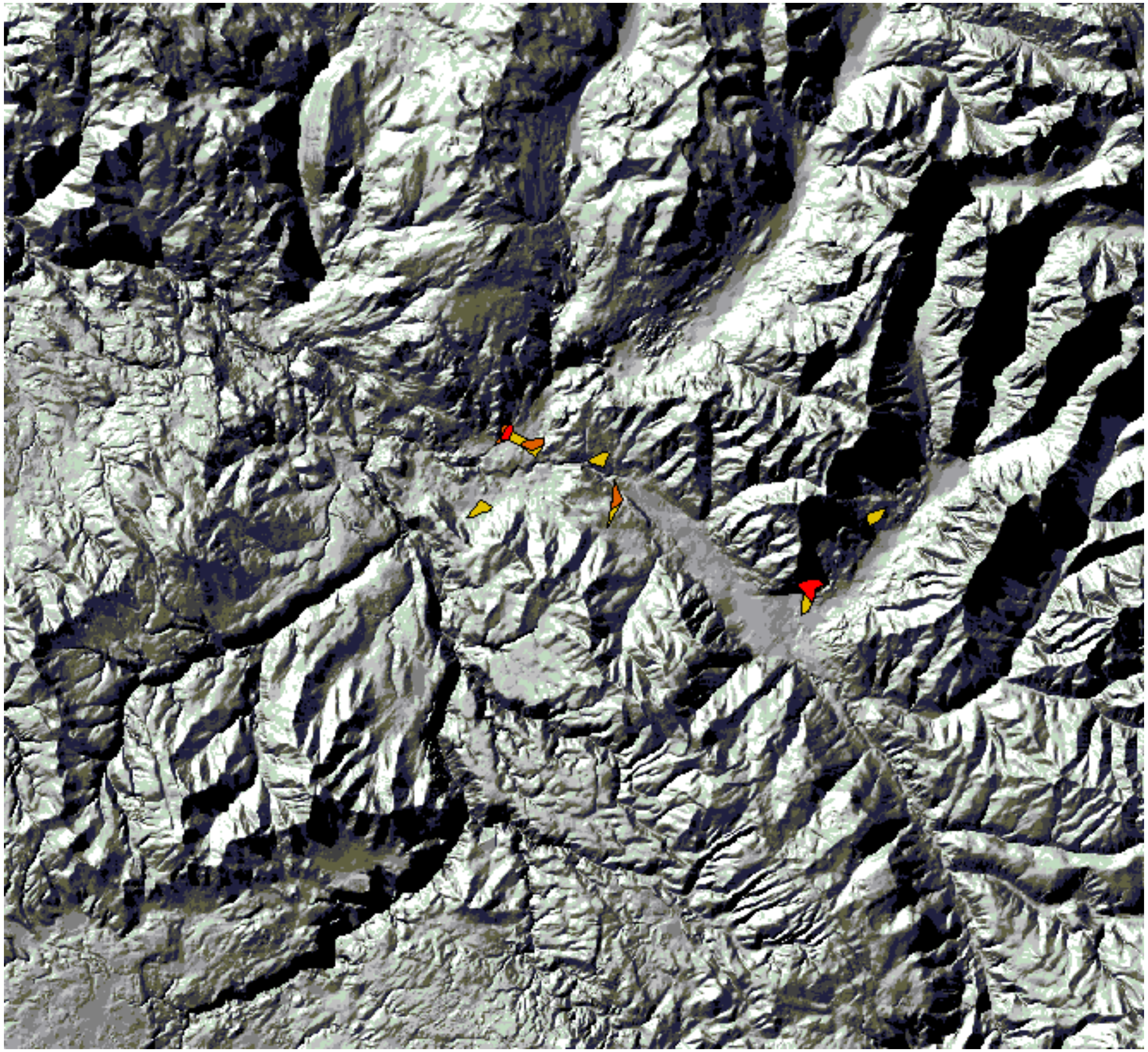
30m Digital Elevation Model with hillshade

Wolf observations

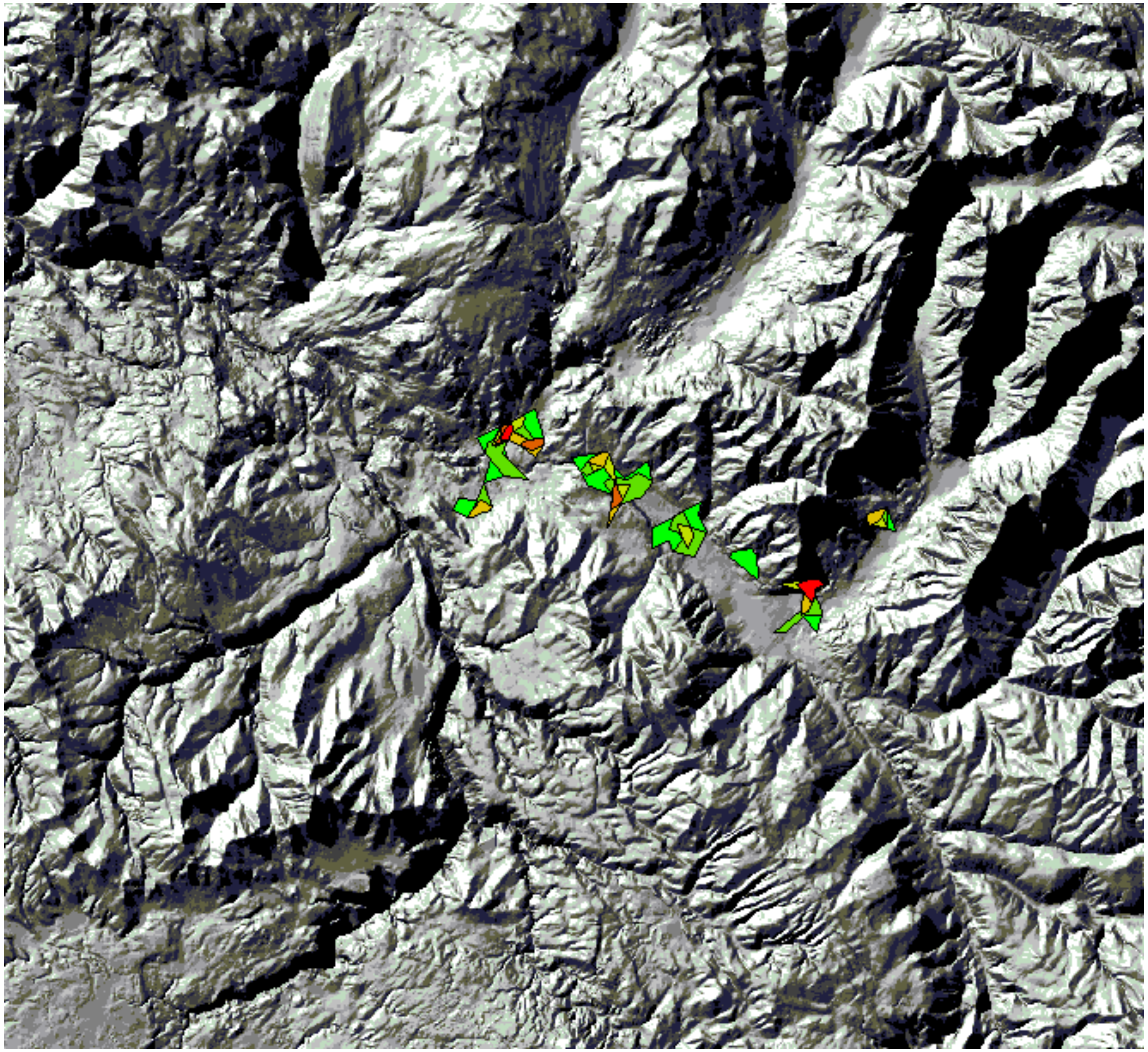


wolves are clearly avoiding step slopes

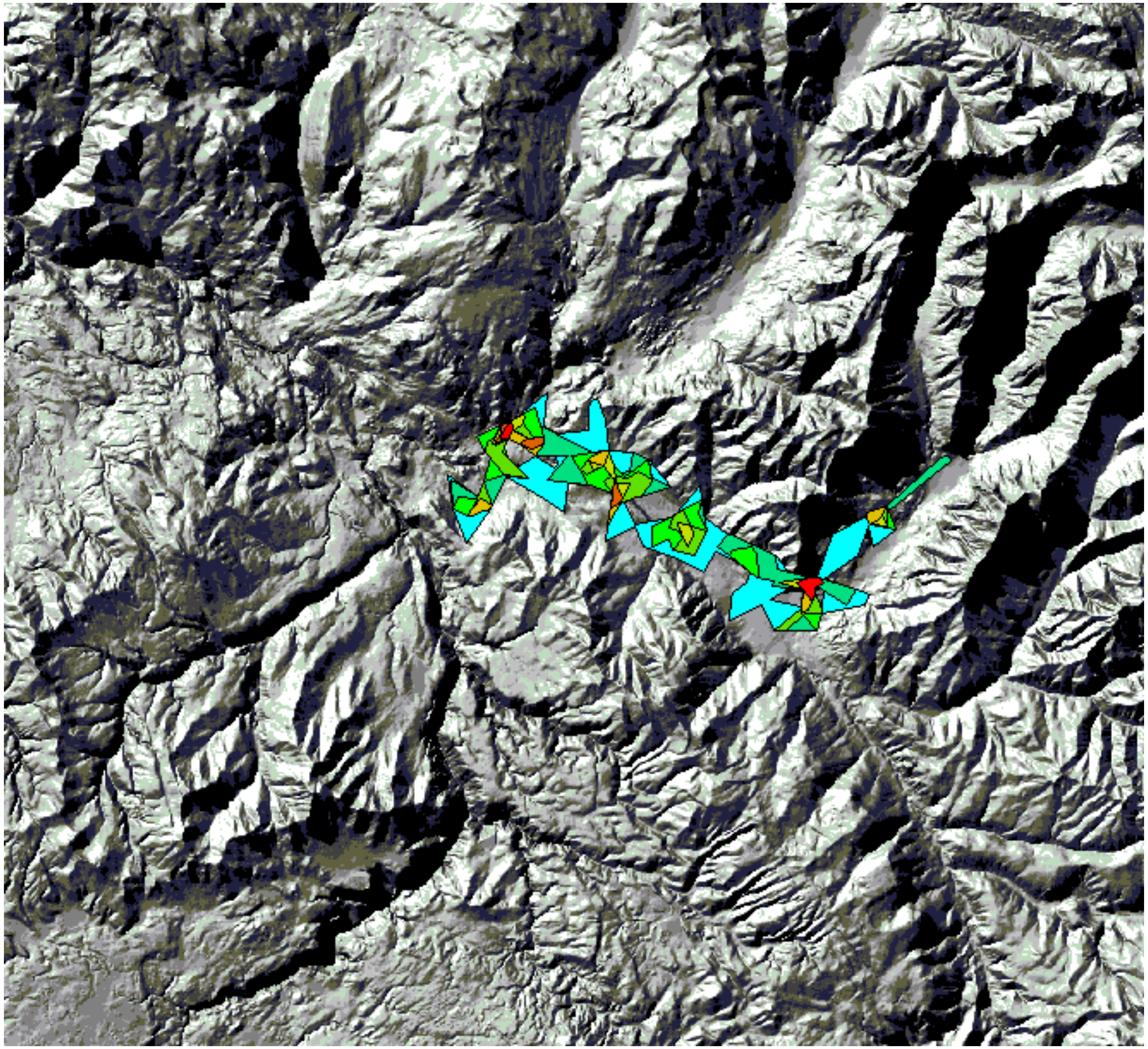
First 3 decile isopleths (30% of observations)



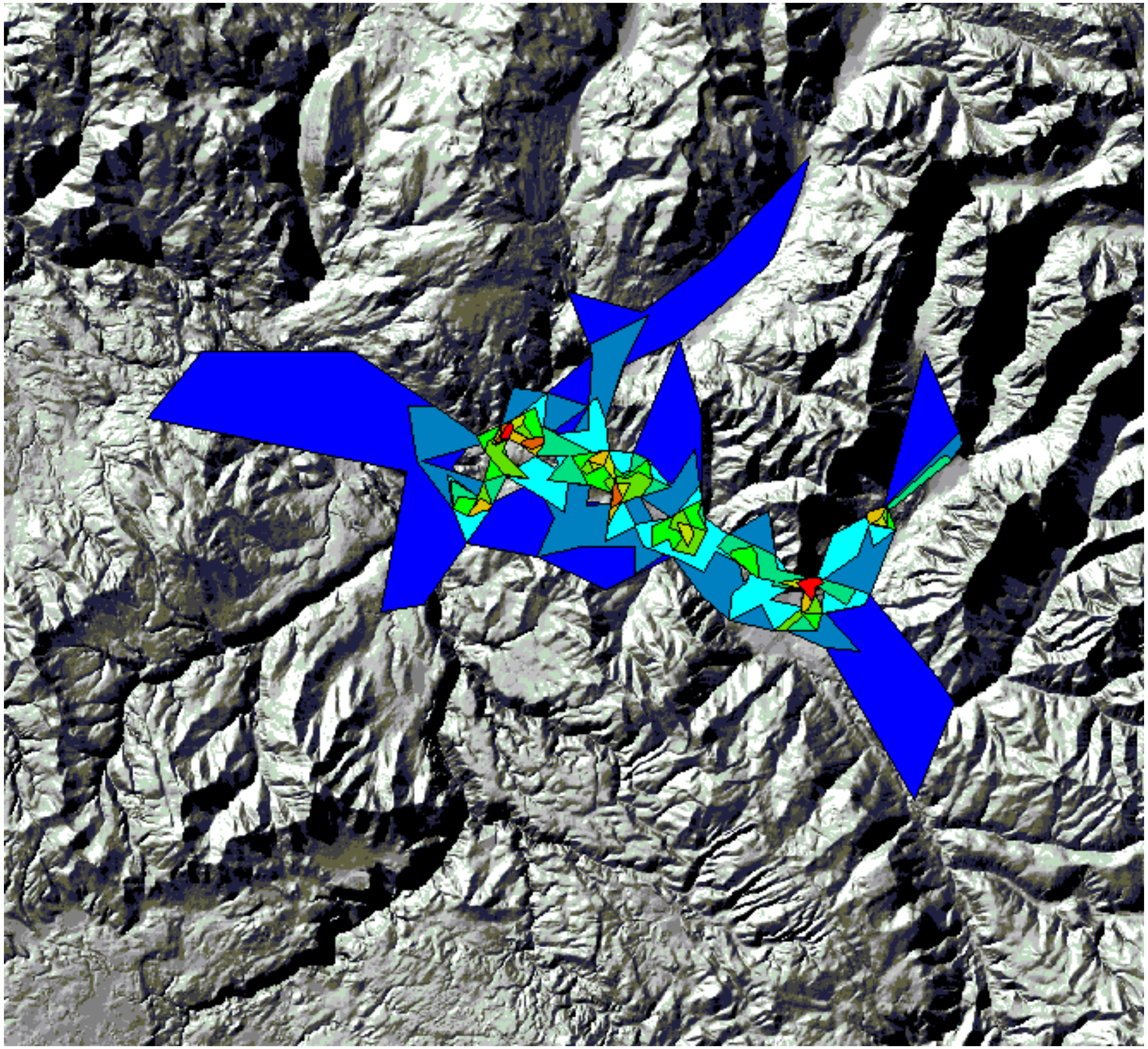
First 6 decile isopleths (60% of observations)



First 9 decile isopleths (90% of observations)



All 10 decile isopleths (100% of observations)



Workshop on R and movement ecology:
Hong Kong University, Jan 2018

Eric Dougherty, Dan Seidel, Wayne Getz

Lecture 3, Part 2

Space-time considerations: T-LoCoH
(Extracted from a talk by Andy Lyons)

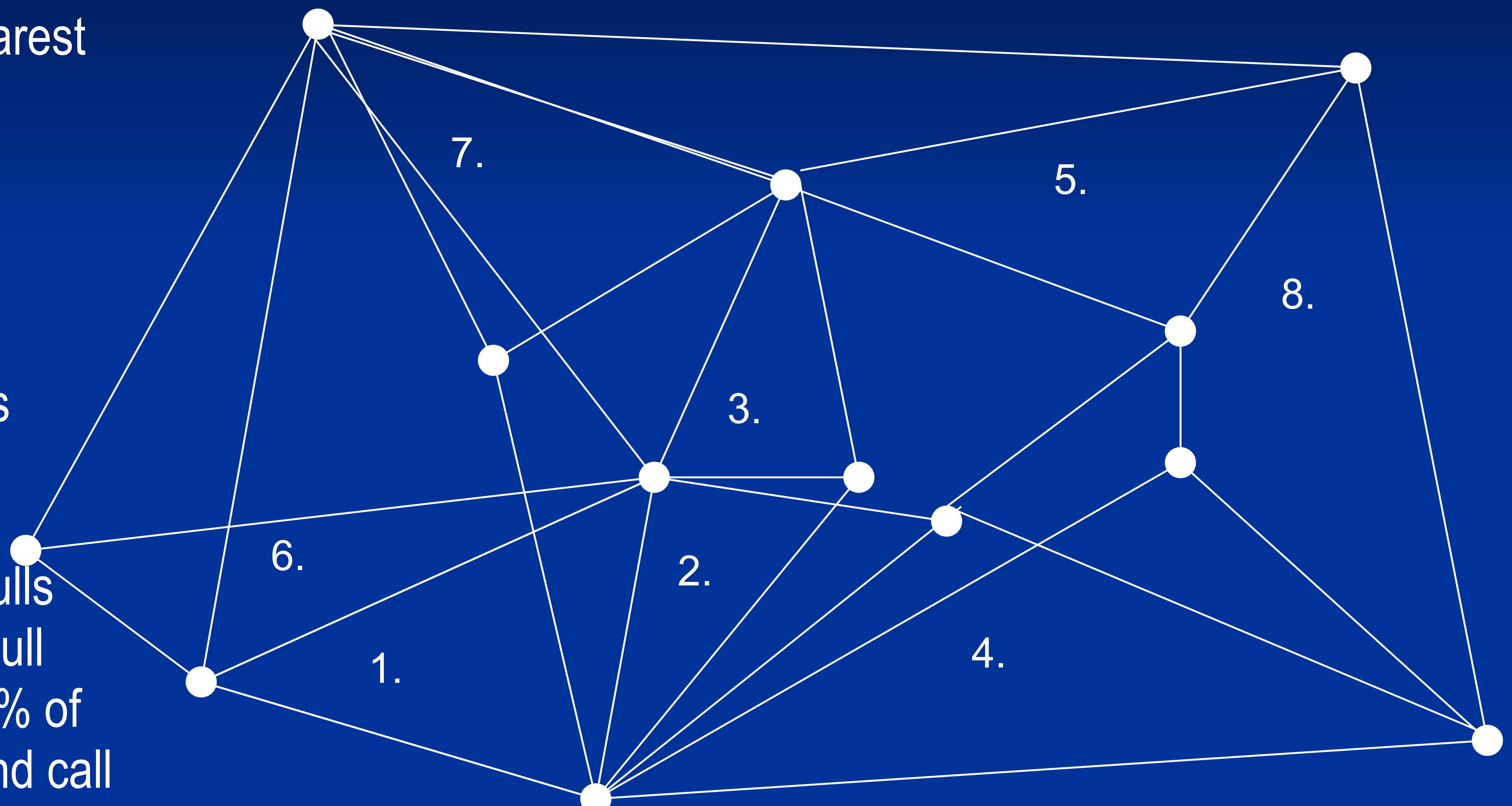
DEPARTMENT *of* ENVIRONMENTAL SCIENCE, POLICY, AND MANAGEMENT

Berkeley UNIVERSITY OF CALIFORNIA College of Natural Resources

LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
    - k-method
    - r-method
    - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze

T-LoCoH Modifications

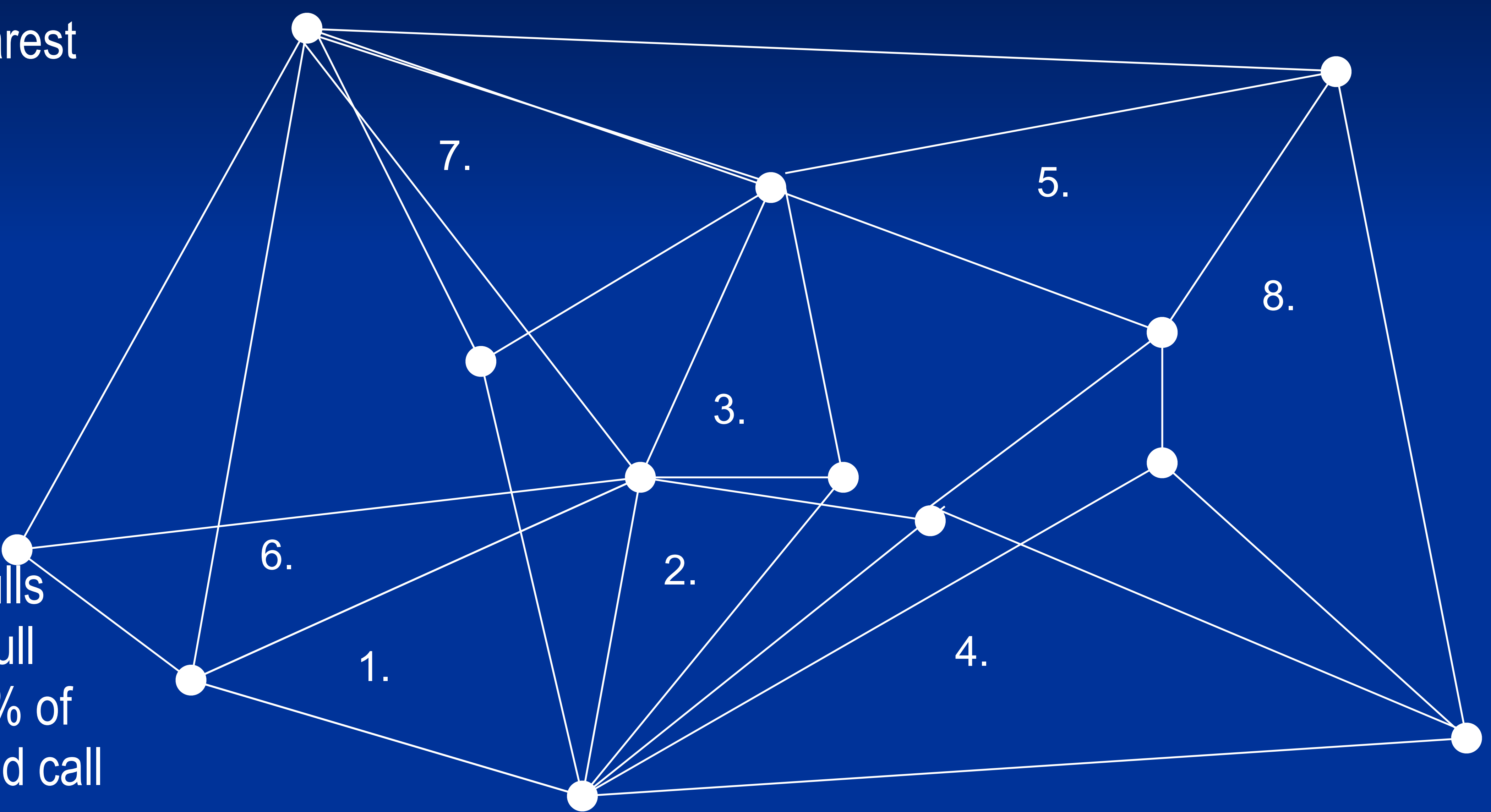


LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze

T-LoCoH Modifications

Euclidean Distance → "Time Scaled Distance"

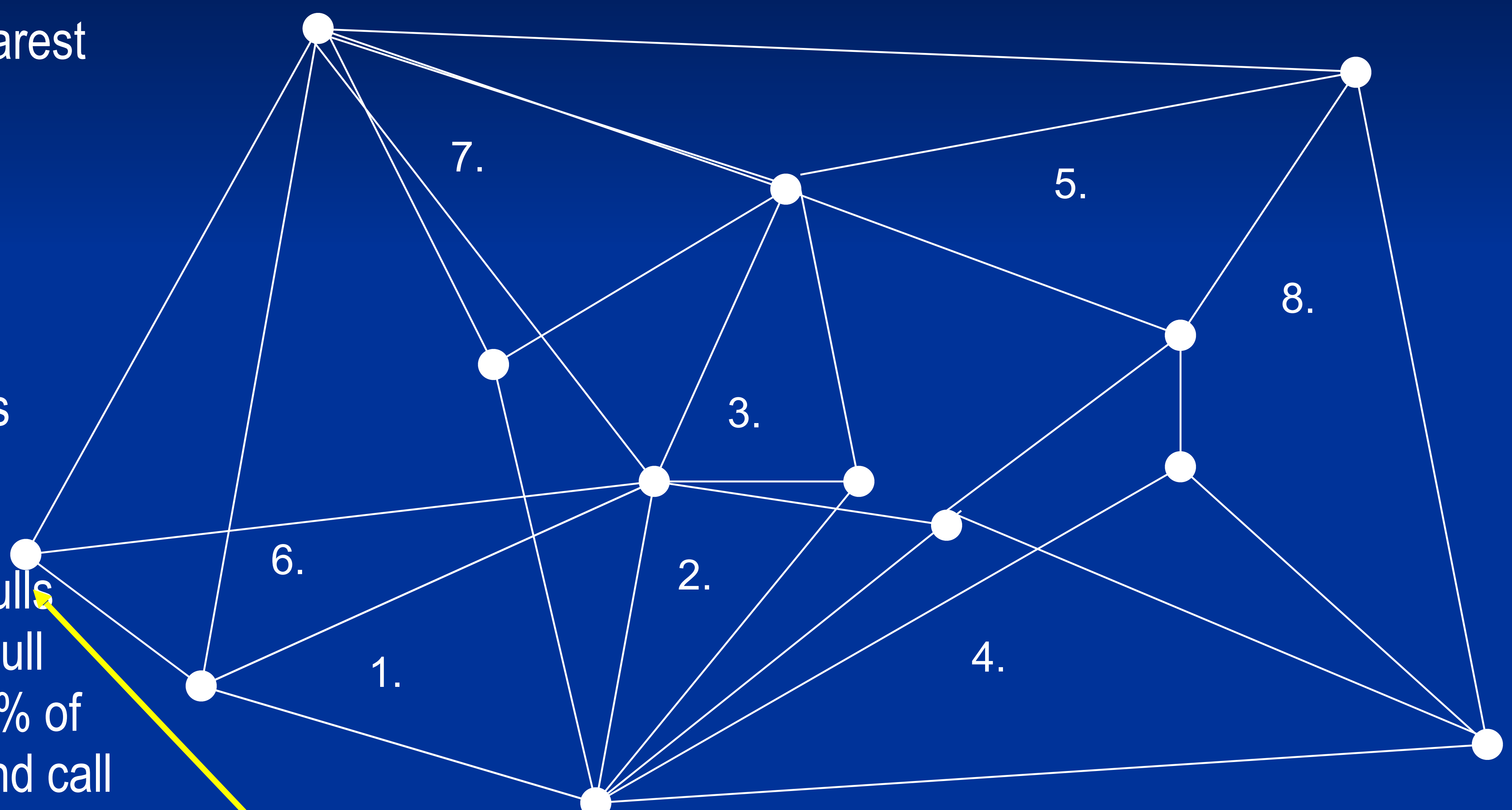


LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze

T-LoCoH Modifications

Euclidean Distance → "Time Scaled Distance"



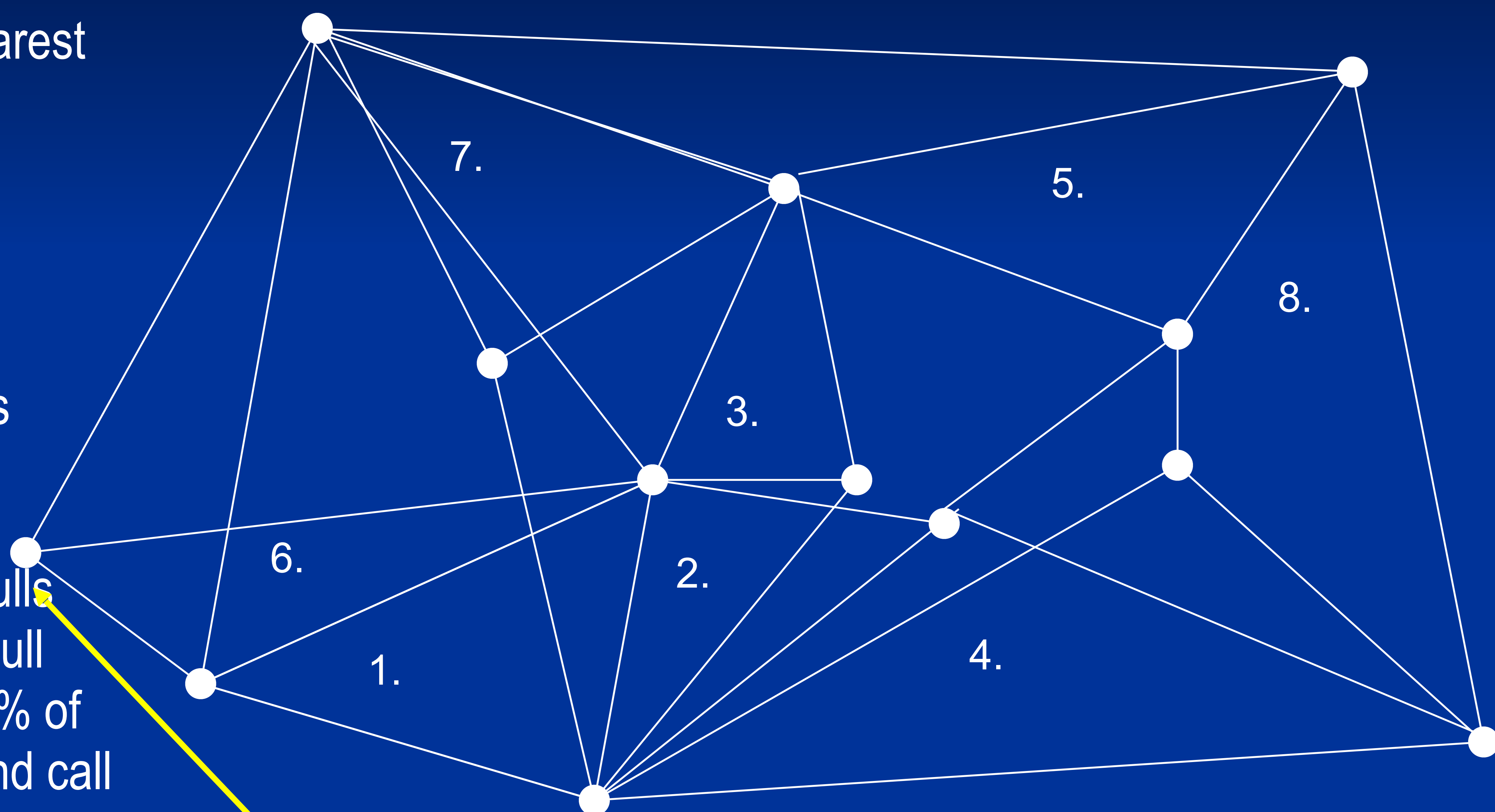
Sort hulls by time-dependent metrics: revisitation, duration, elongation

LoCoH Algorithm

1. Loop through points
2. For each point, calculate distances to nearby points
3. Pick a set of nearest neighbors
   - k-method
   - r-method
   - a-method
4. Draw local hulls around all points
5. Sort hulls in a meaningful way
6. Start merging hulls
7. When merged hull encompasses x% of points, pause and call that an isopleth
8. Visualize & analyze

T-LoCoH Modifications

Euclidean Distance → "Time Scaled Distance"



Sort hulls by time-dependent metrics: revisitation, duration, elongation

New visualization tools

Time Scaled Distance

Time Scaled Distance

- Want the “distance” to reflect both how far apart two points are in space as well as time
  - i.e., “push apart” points based upon how far they are separated in time

Time Scaled Distance

- Want the "distance" to reflect both how far apart two points are in space as well as time
  - i.e., "push apart" points based upon how far they are separated in time
- We transform the time difference between two points to spatial units by asking:

  how far would the animal have traveled in this time interval it had been moving in a random walk?

Time Scaled Distance

- Want the "distance" to reflect both how far apart two points are in space as well as time
  - i.e., "push apart" points based upon how far they are separated in time
- We transform the time difference between two points to spatial units by asking:

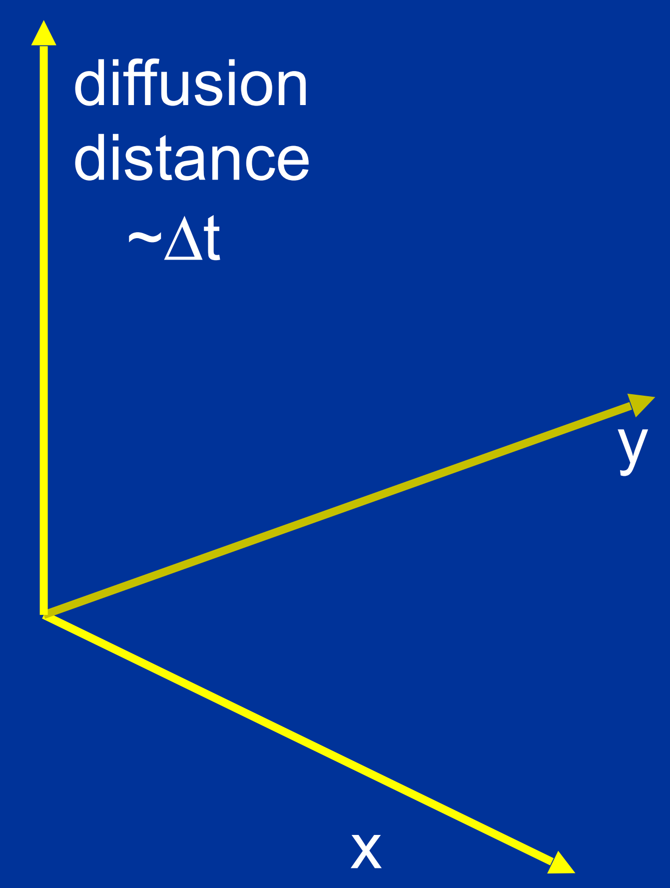
  how far would the animal have traveled in this time interval it had been moving in a random walk?
- This diffusion distance becomes a third axis in "space time" space

Time Scaled Distance

- Want the "distance" to reflect both how far apart two points are in space as well as time
  - i.e., "push apart" points based upon how far they are separated in time
- We transform the time difference between two points to spatial units by asking:

  how far would the animal have traveled in this time interval it had been moving in a random walk?

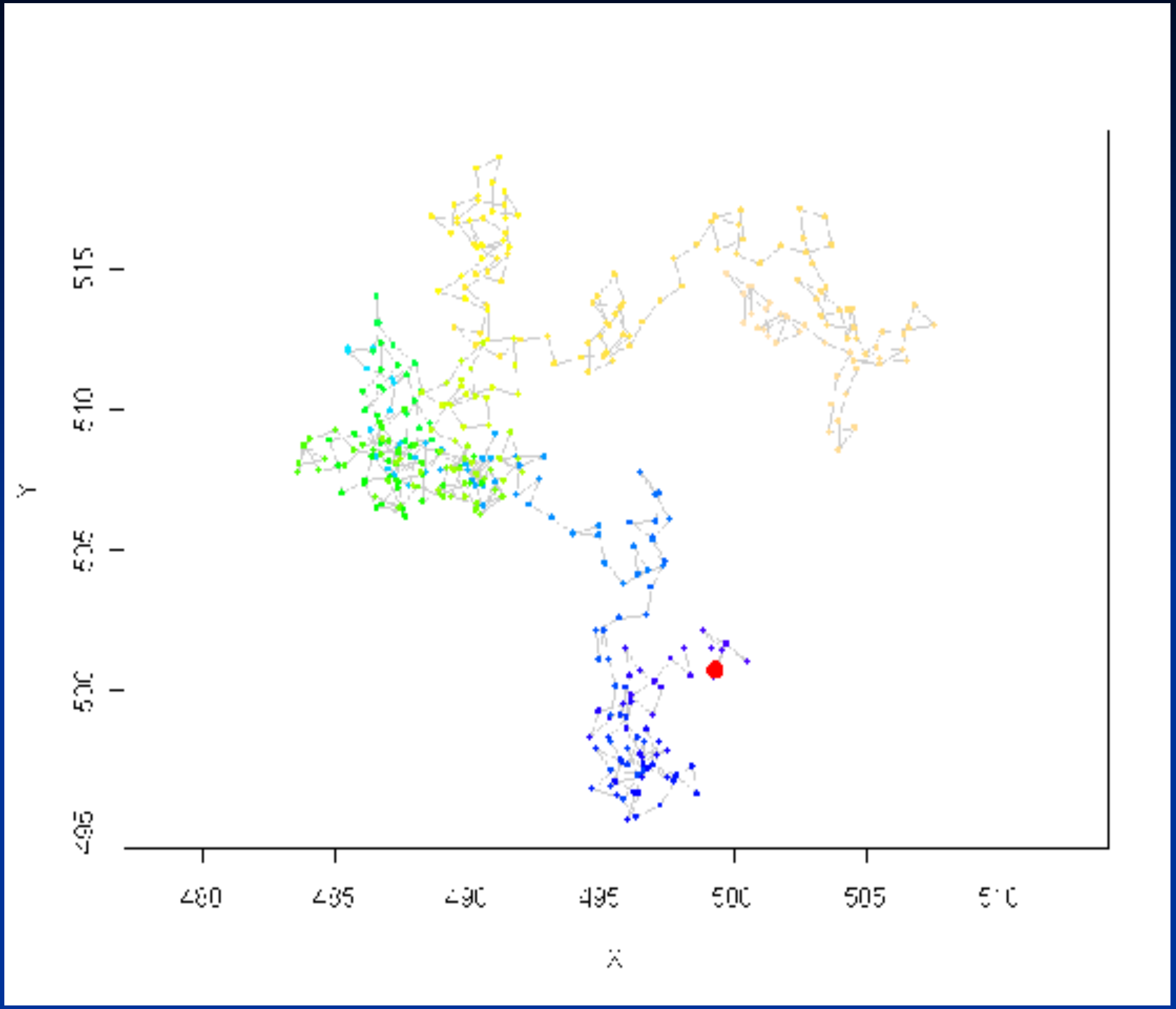
- This diffusion distance becomes a third axis in "space time" space



Diffusion distance δ

For a random walk where:
λ = constant step length
N = number of steps between two selected points

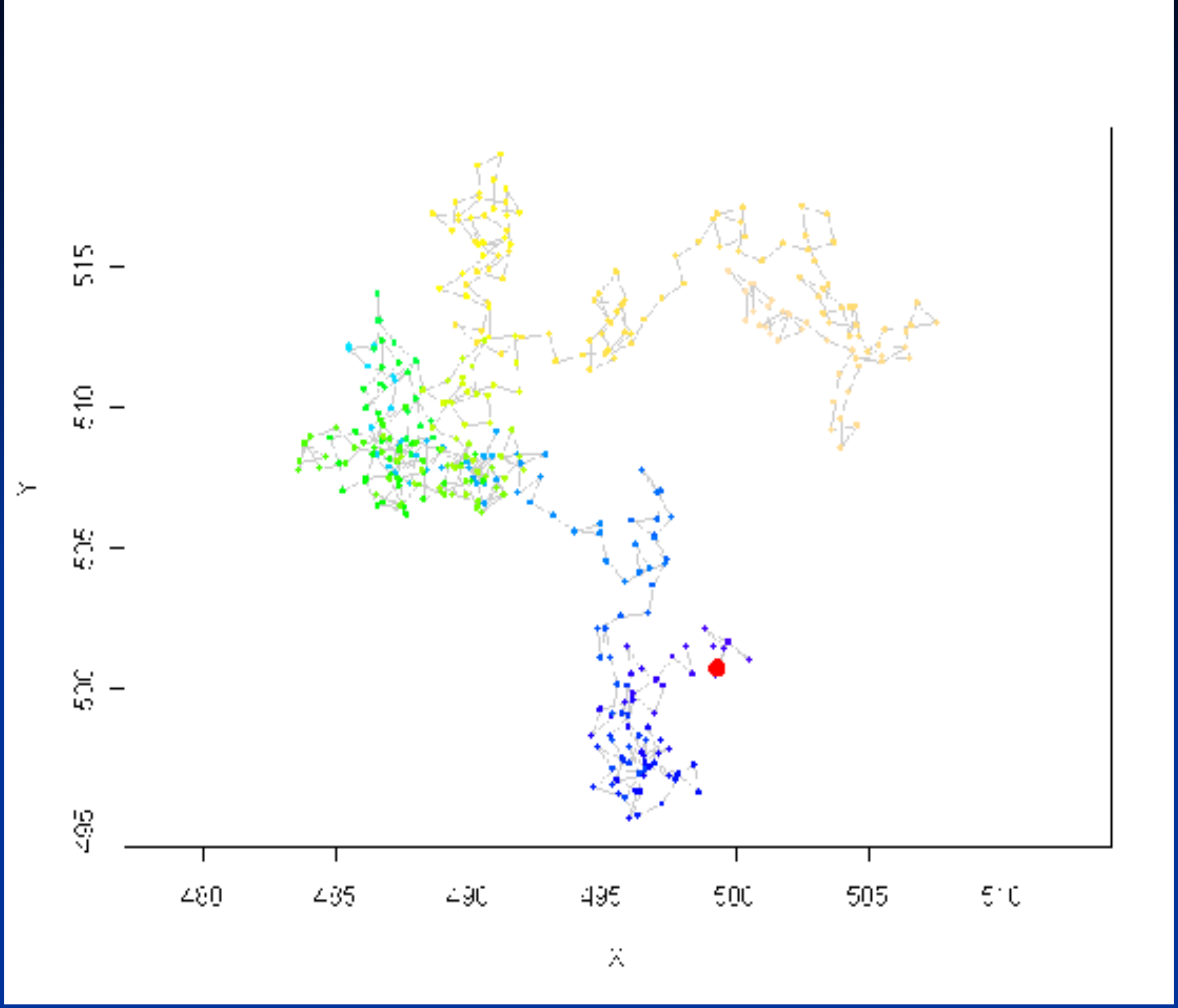
$$\delta(N) = \lambda\sqrt{N}$$



Diffusion distance δ

For a random walk where:
λ = constant step length
N = number of steps between two selected points

$$\delta(N) = \lambda\sqrt{N}$$



For a dataset with variable step length but roughly constant sampling interval:

$$\delta(\Delta t) \approx \bar{d}\sqrt{\frac{\Delta t}{\tau}}$$

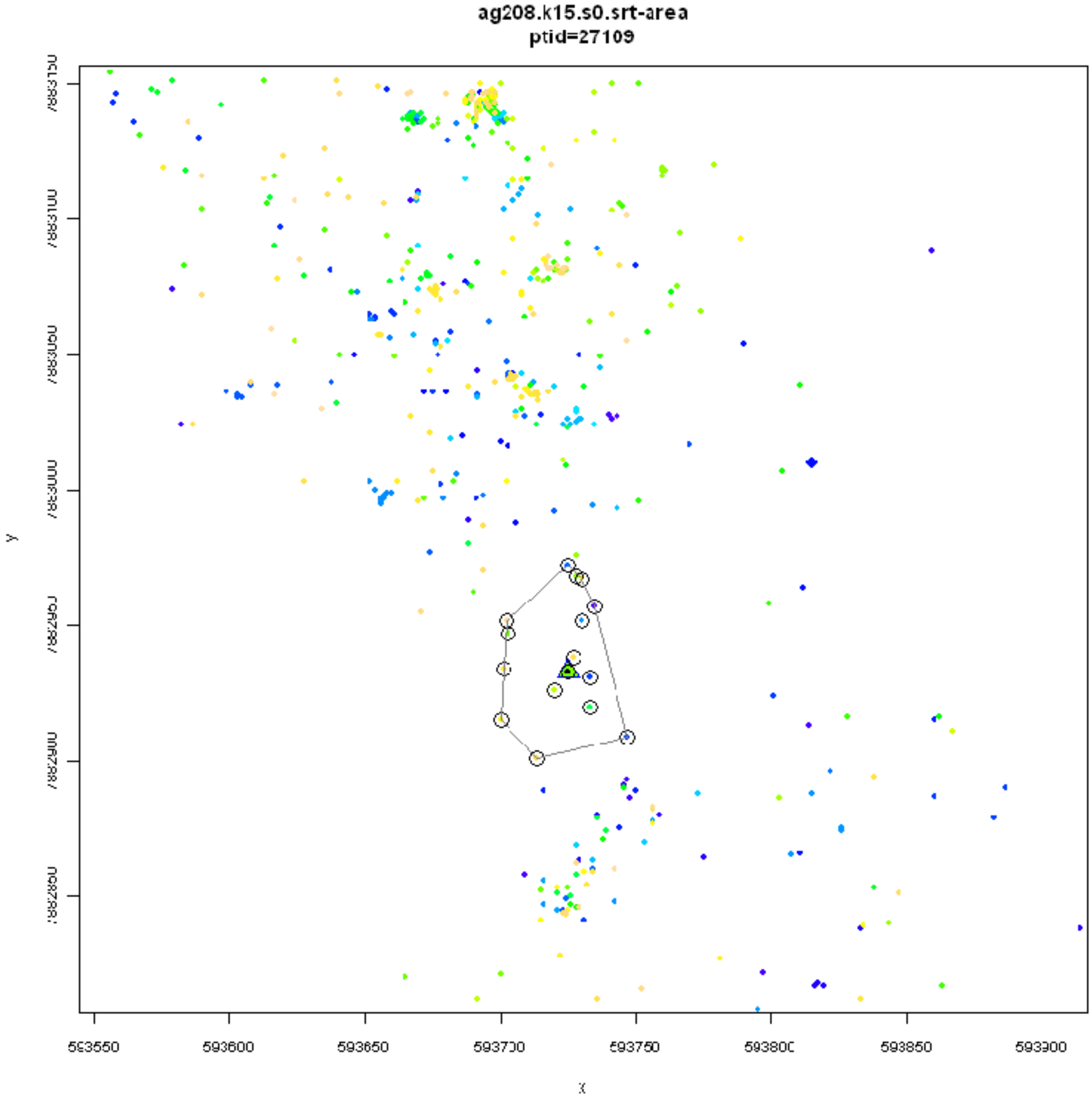
$\bar{d}$ = median step length
τ = median sampling frequency

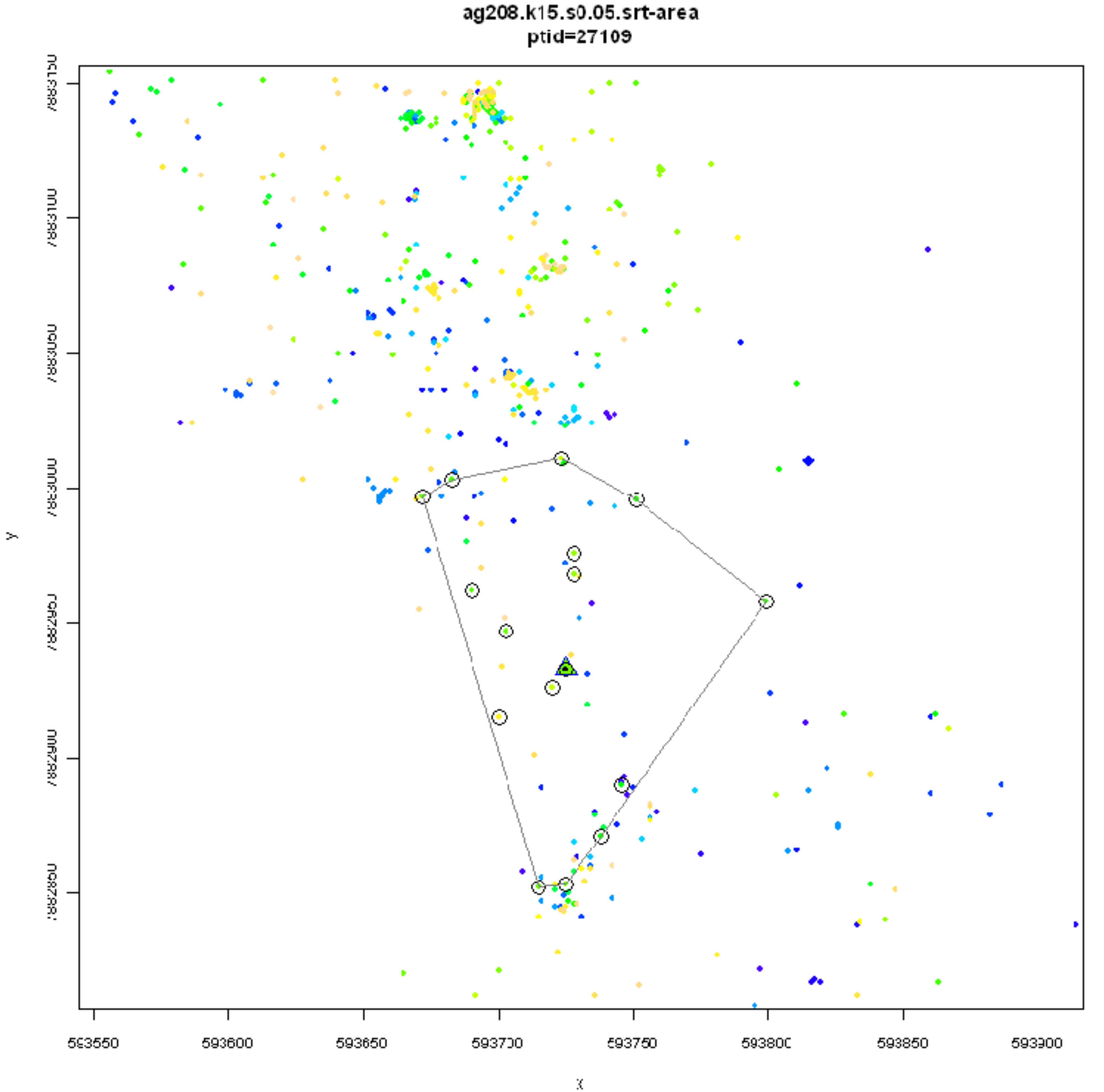
Time-Scaled Distance (TSD)

$$D_{ij} = \sqrt{\Delta x_{ij}^{\;2} + \Delta y_{ij}^{\;2} + s\delta(\Delta t_{ij})^2}$$

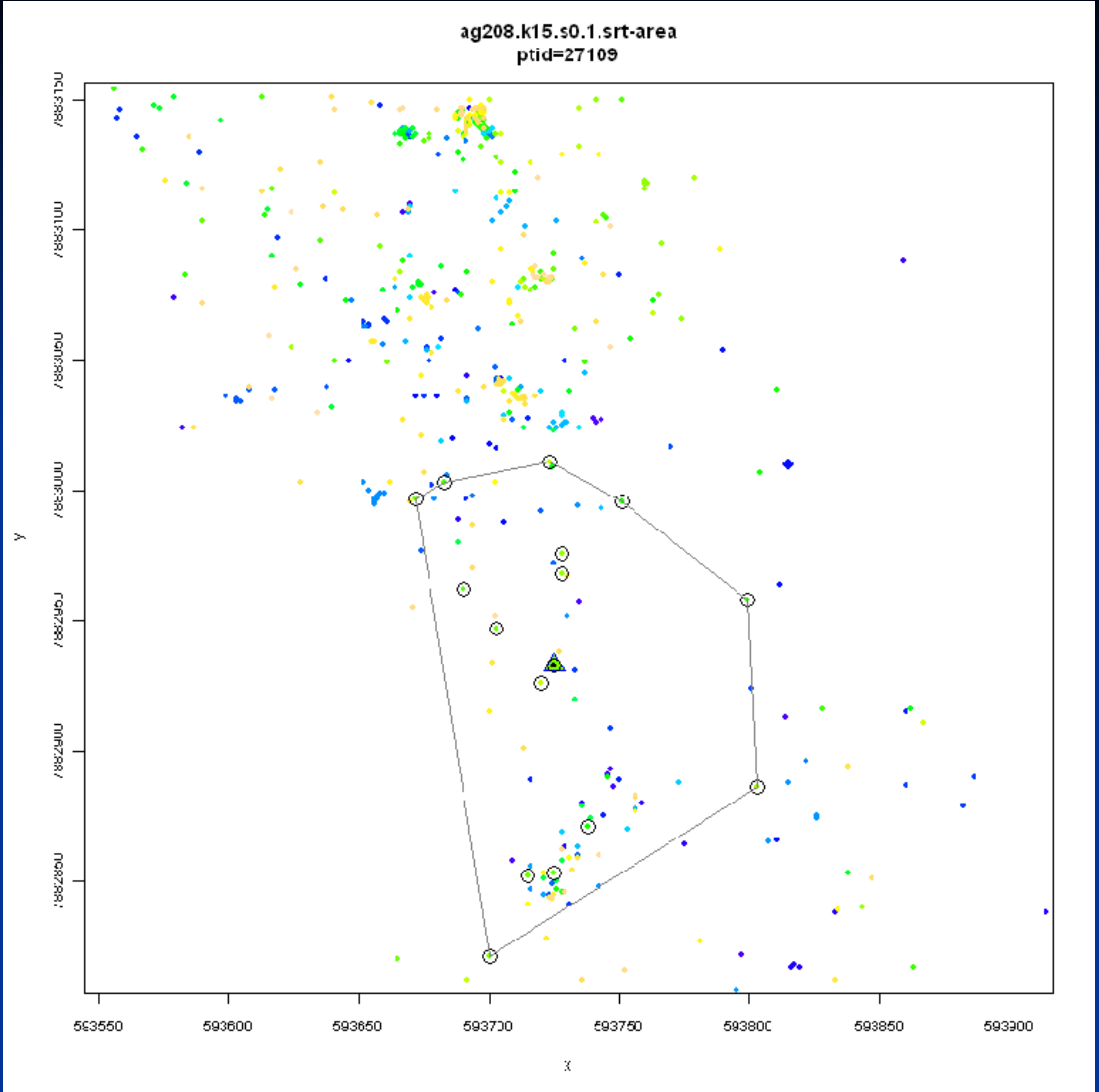
$$= \sqrt{\Delta x_{ij}^{\;2} + \Delta y_{ij}^{\;2} + s\bar{d}^{\,2}\,\frac{\Delta t_{ij}}{\tau}}$$

where *s* is a dimensionless scaling factor that controls the degree to which diffusion distance influences Euclidean distance ($s \geq 0$)

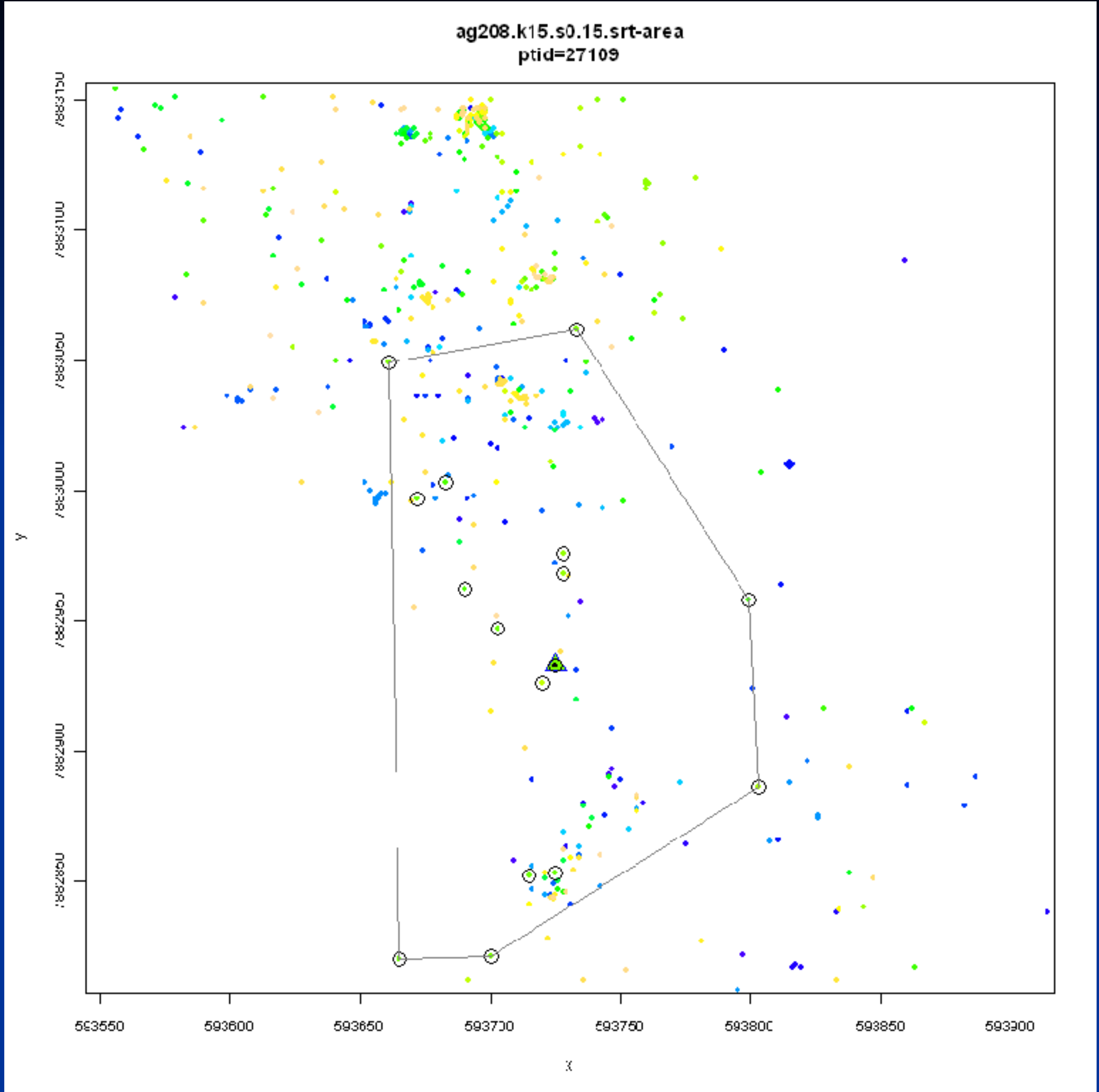

ag208.k15.s0.srt-area
ptid=27109
y
7883150
7883100
7883050
7883000
7882950
7882900
7882850
x
593550
593600
593650
593700
593750
593800
593850
593900

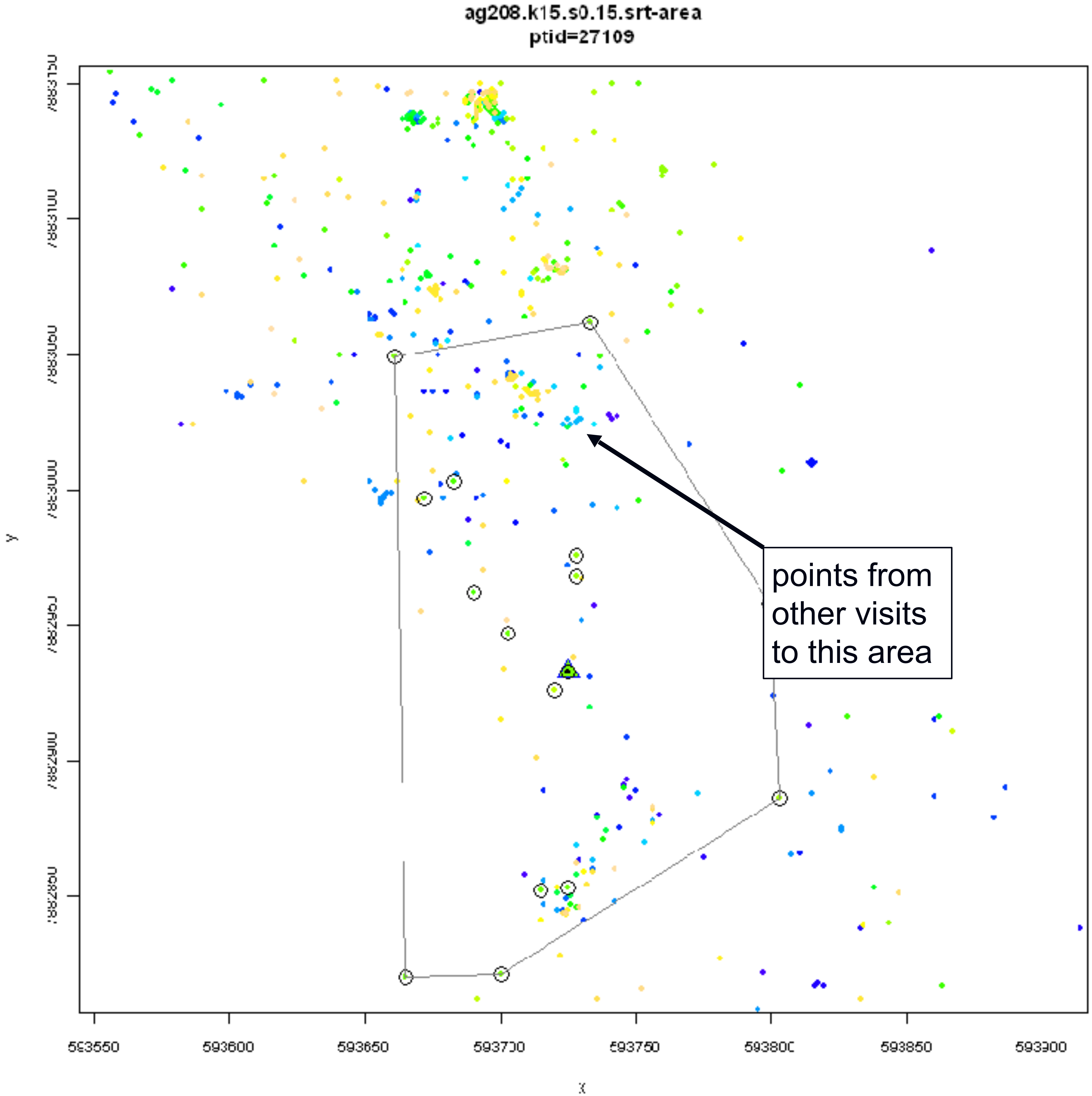


ag208.k15.s0.05.srt-area
ptid=27109
7883150
7883100
7883050
7883000
7882950
7882900
7882850
y
593550
593600
593650
593700
593750
593800
593850
593900
x



ag208.k15.s0.1.srt-area
ptid=27109
y
x
593550
593600
593650
593700
593750
593800
593850
593900


ag208.k15.s0.15.srt-area
ptid=27109
593550
593600
593650
593700
593750
593800
593850
593900
x
y


ag208.k15.s0.15.srt-area
ptid=27109
points from other visits to this area
593550
593600
593650
593700
593750
593800
593850
593900
x
y

Time Use Space

Time Use Space

T-LoCoH Hull Metrics

Density

- **area**
- **number of nearest neighbors** used in hull construction
- **number of enclosed points**

Time Use

- **revisitation rate** (number of separate visits where visits are differentiated by an inter-visit gap period)
- **mean visit duration** (mean number of occurrence per visit)
- revisitation and mean visit duration **normalized by hull area**

Time

- **hour of day** of the parent point
- **month** of the parent point
- **date** of the parent point

Elongation / Movement Phase

- **eccentricity** of a bounding ellipsoid constructed around the hull
- **perimeter / area ratio**
- **average speed of nearest neighbors** used in hull construction (where the speed of a point sampled at time t is measured from t-1 to t+1).
- **average speed of all points** enclosed by the hull
- **standard deviation** of the speed of nearest neighbor points
- **standard deviation** of the speed of enclosed points

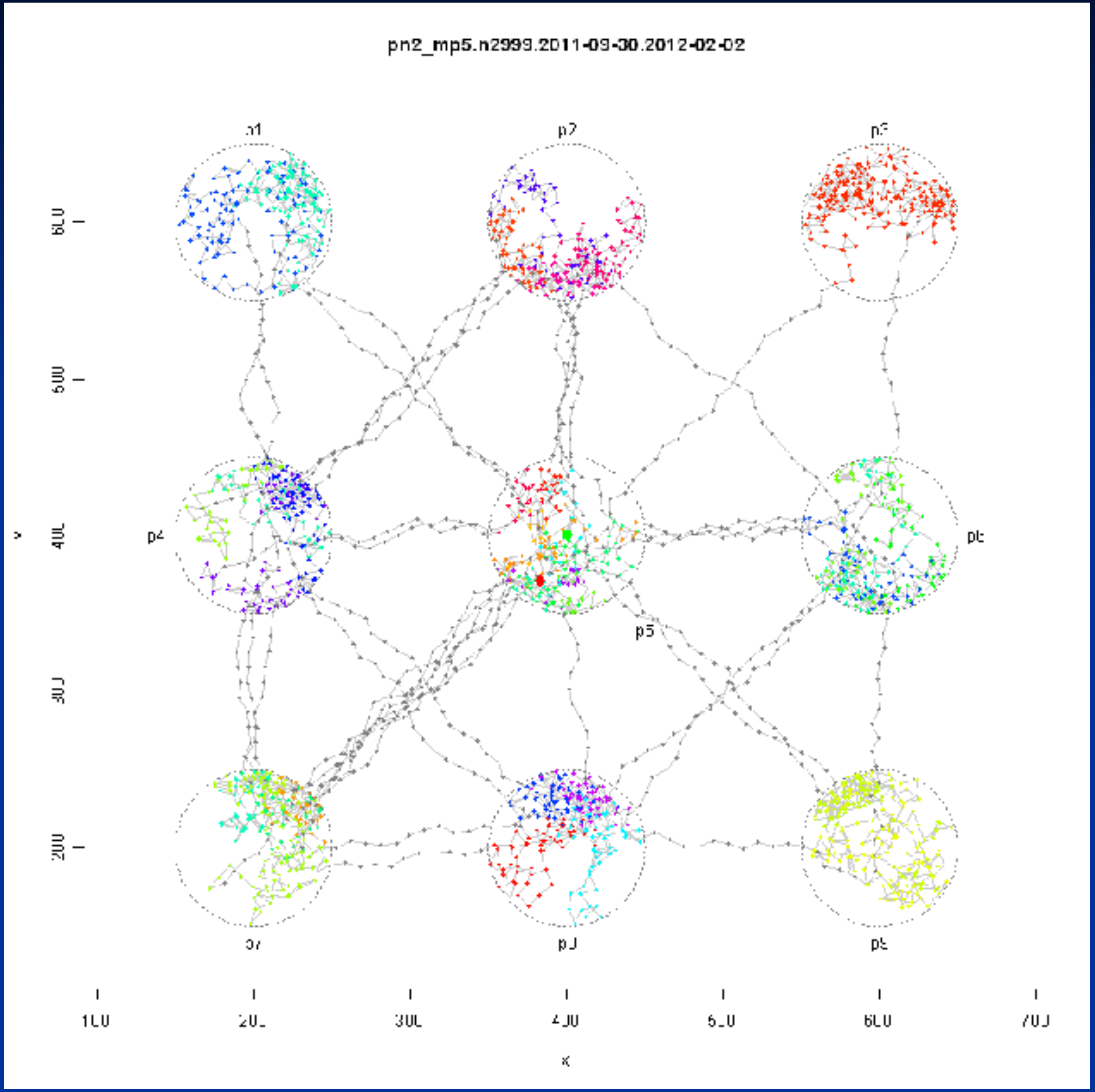
T-LoCoH General Workflow

1. Select a value of *s* based on the time scale of interest
2. Create density isopleths that do a "good job" representing the home range
   e.g., no spurious crossovers
3. Compute hull metrics for elongation and/or time-use
4. Visualize isopleths and/or hull points
5. Interpret and/or plot against environmental variables

Simulated Data

1. Single virtual animal moves between 9 patches
2. constant step size and sampling interval
3. unbounded random walk within each patch for a predetermined # steps
4. directional movement to the next patch
5. duration and frequency of patch use varied

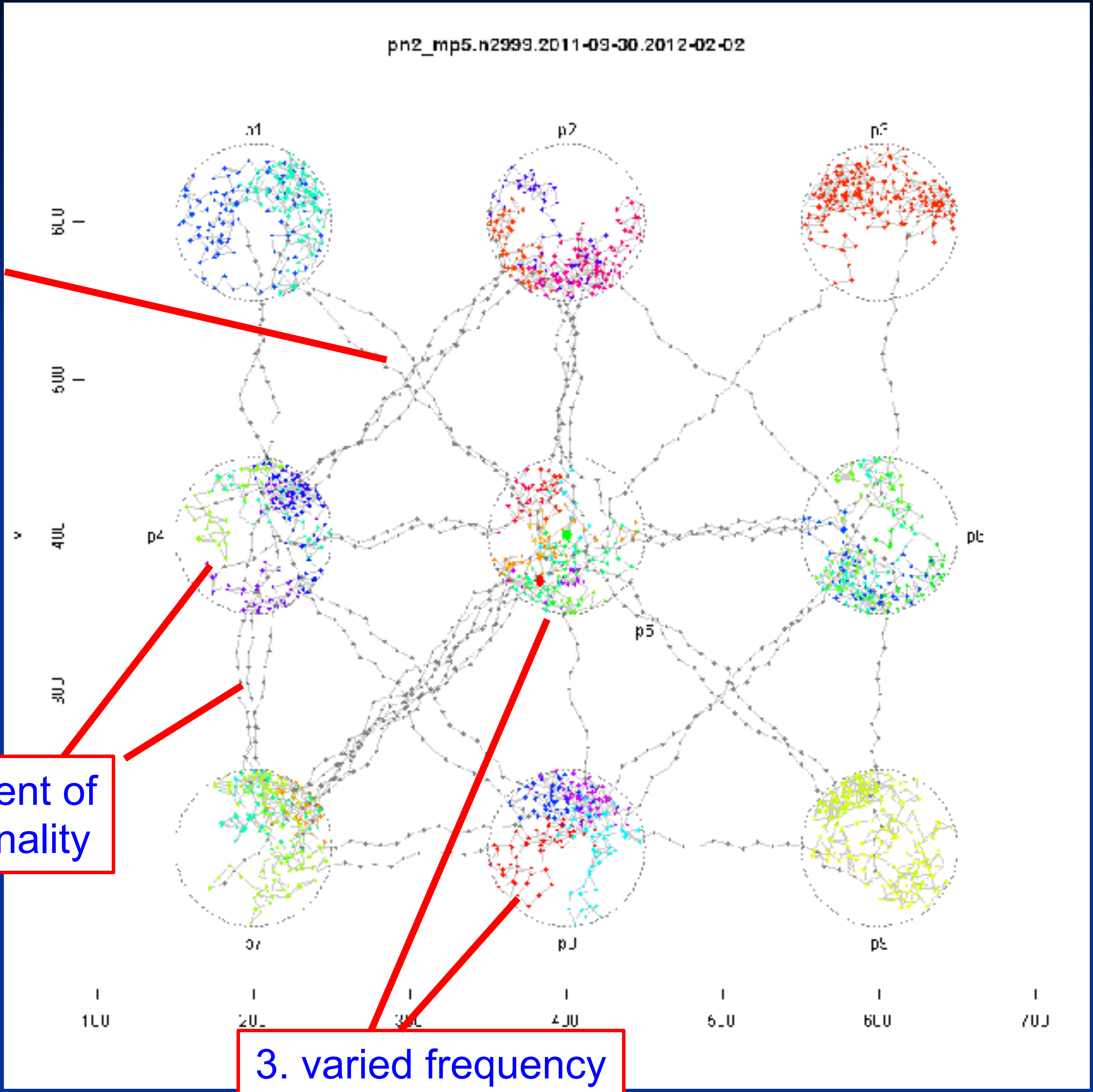
Patch	Visits	Total Pts
p1	2 x 120	240
p2	4 x 60	240
p3	1 x 240	240
p4	6 x 40	240
p5	12 x 20	240
p6	4 x 60	240
p7	6 x 40	240
p8	4 x 60	240
p9	2 x 120	240



Simulated Data

1. Single virtual animal moves between 9 patches
2. constant step size and sampling interval
3. unbounded random walk within each p… # steps…
4. directi… patch
5. duration and frequency of patch use varied

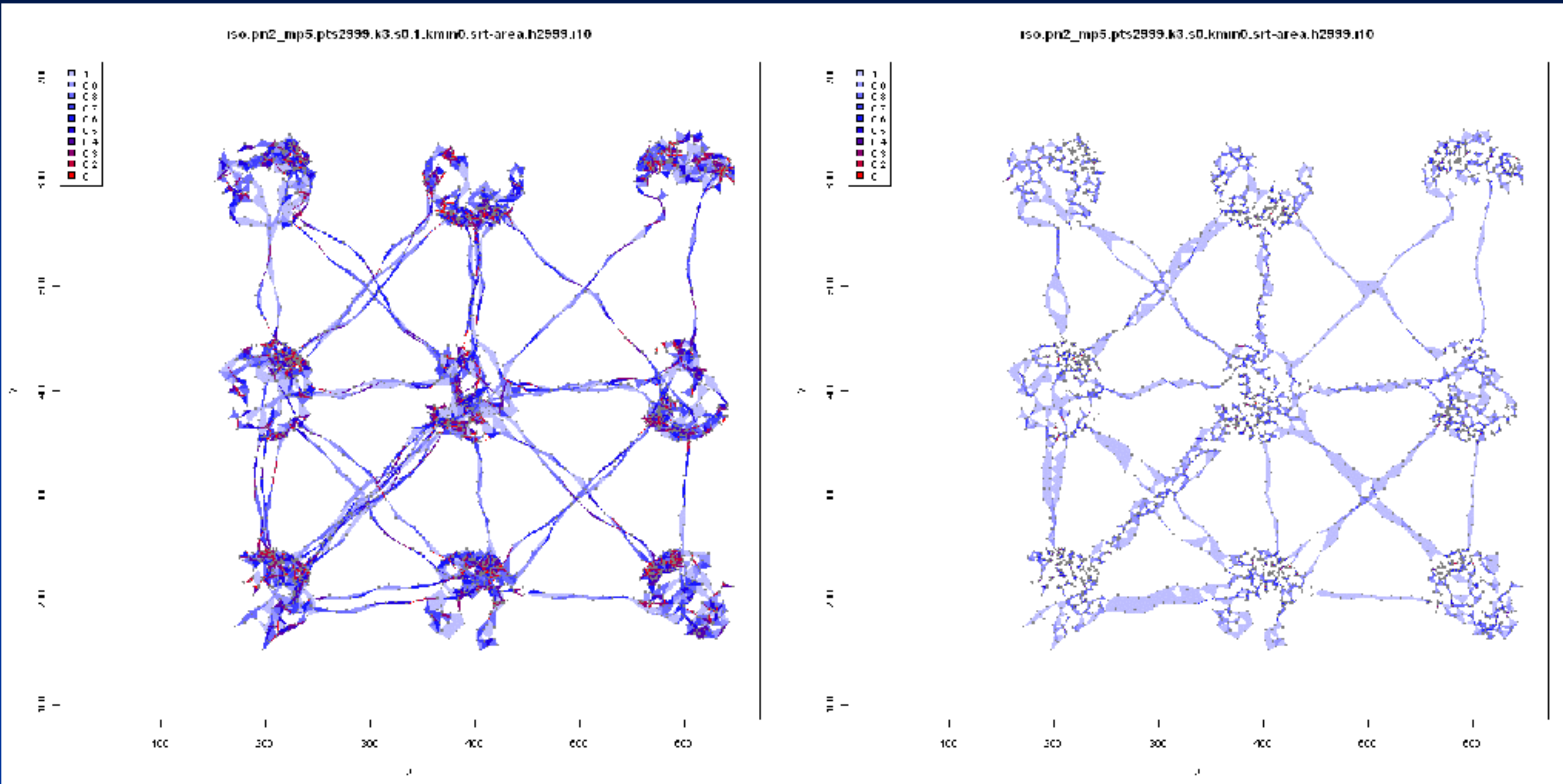
Patch	Visits	Total Pts
p1	2 x 120	240
p2	4 x 60	240
p3	1 x 240	240
p4	6 x 40	240
p5	12 x 20	240
p6	4 x 60	240
p7	6 x 40	240
p8	4 x 60	240
p9	2 x 120	240



With Time

k = 3

Without Time



s = 0.1

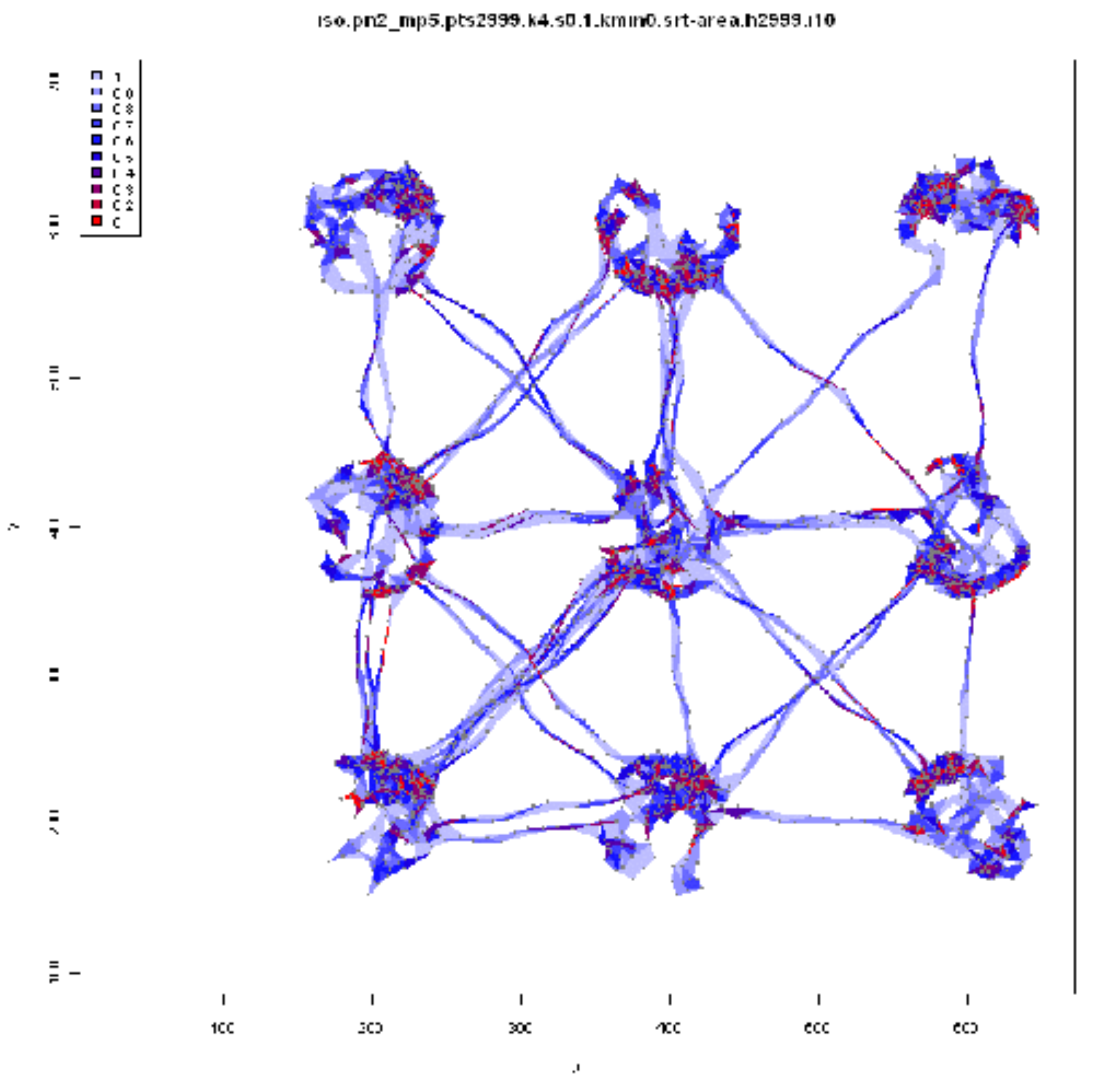
s = 0

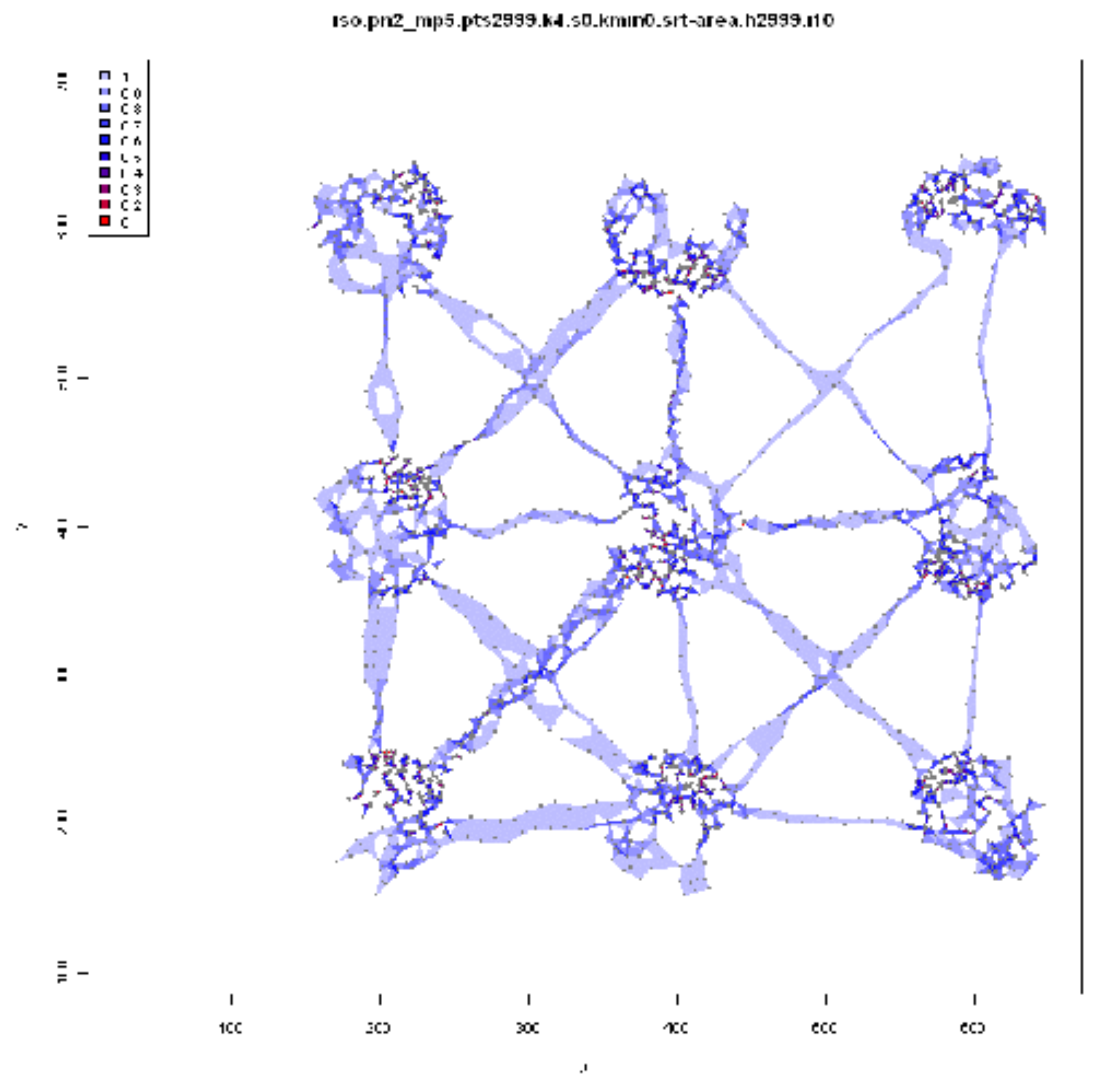
Isopleth level indicates the proportion of total points enclosed along a gradient of point density (red highest density, light blue lowest).

With Time

k = 4

Without Time



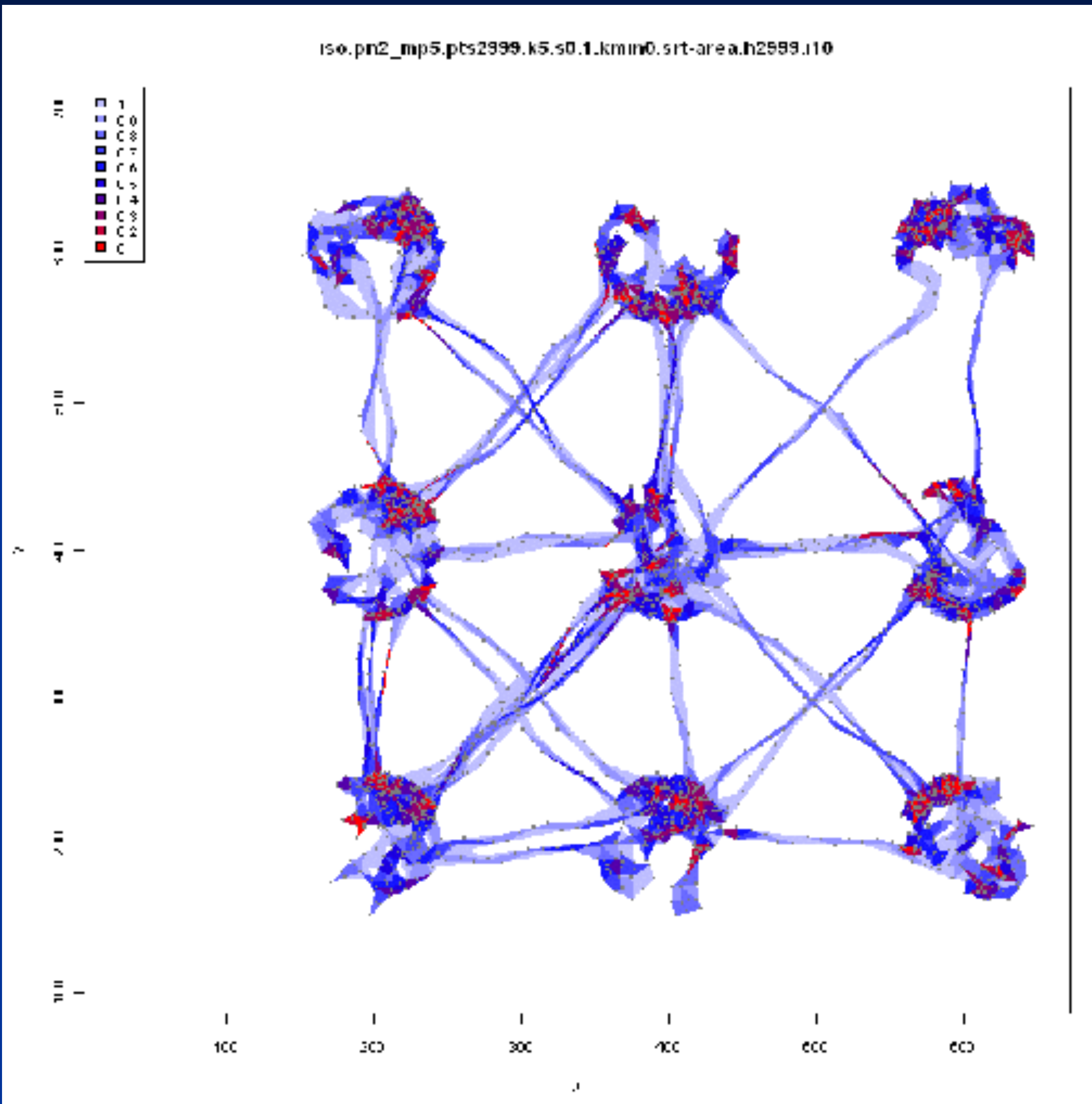


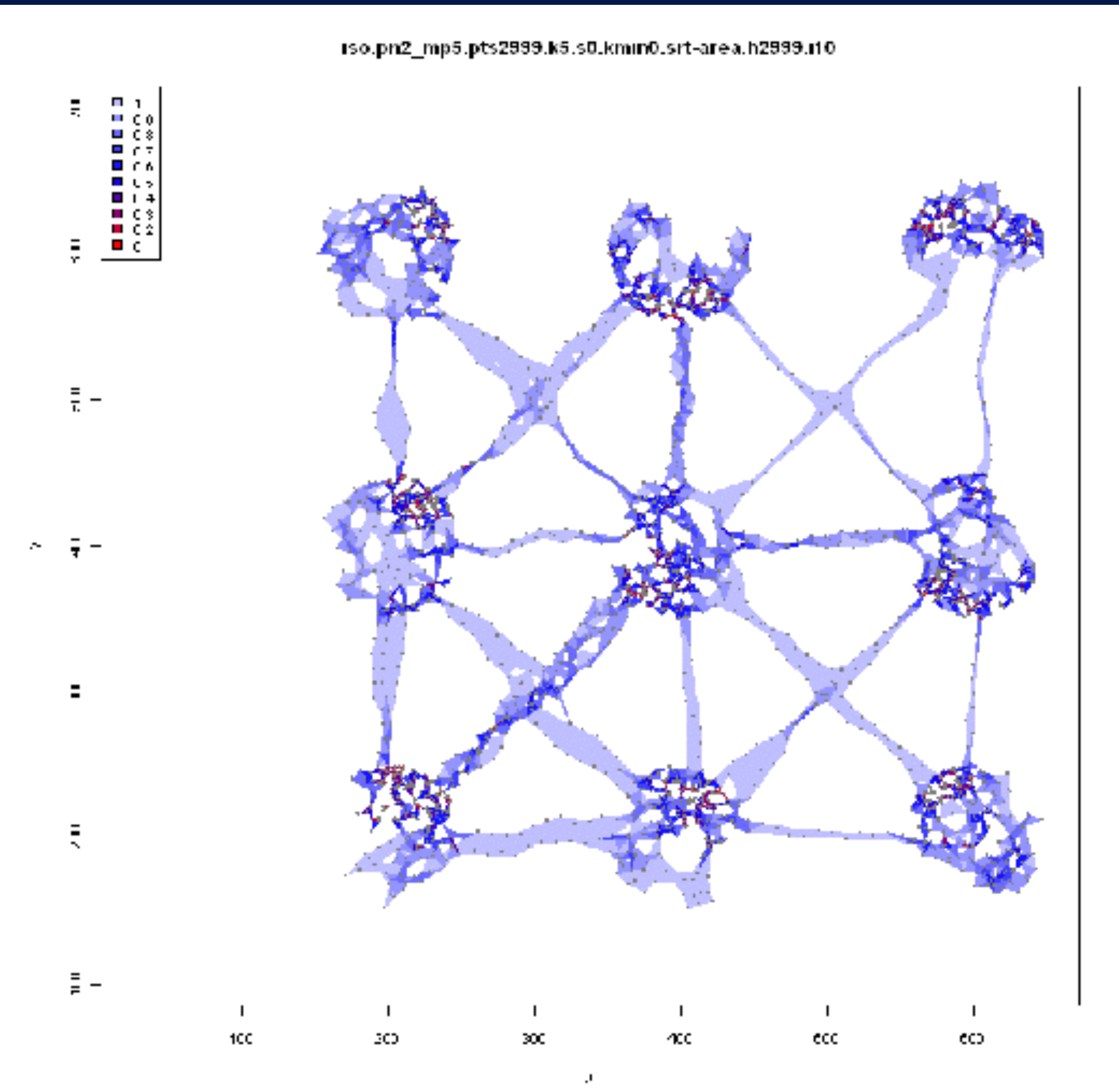
s = 0.1

s = 0

Isopleth level indicates the proportion of total points enclosed along a gradient of point density (red highest density, light blue lowest).

With Time **k = 5** **Without Time**





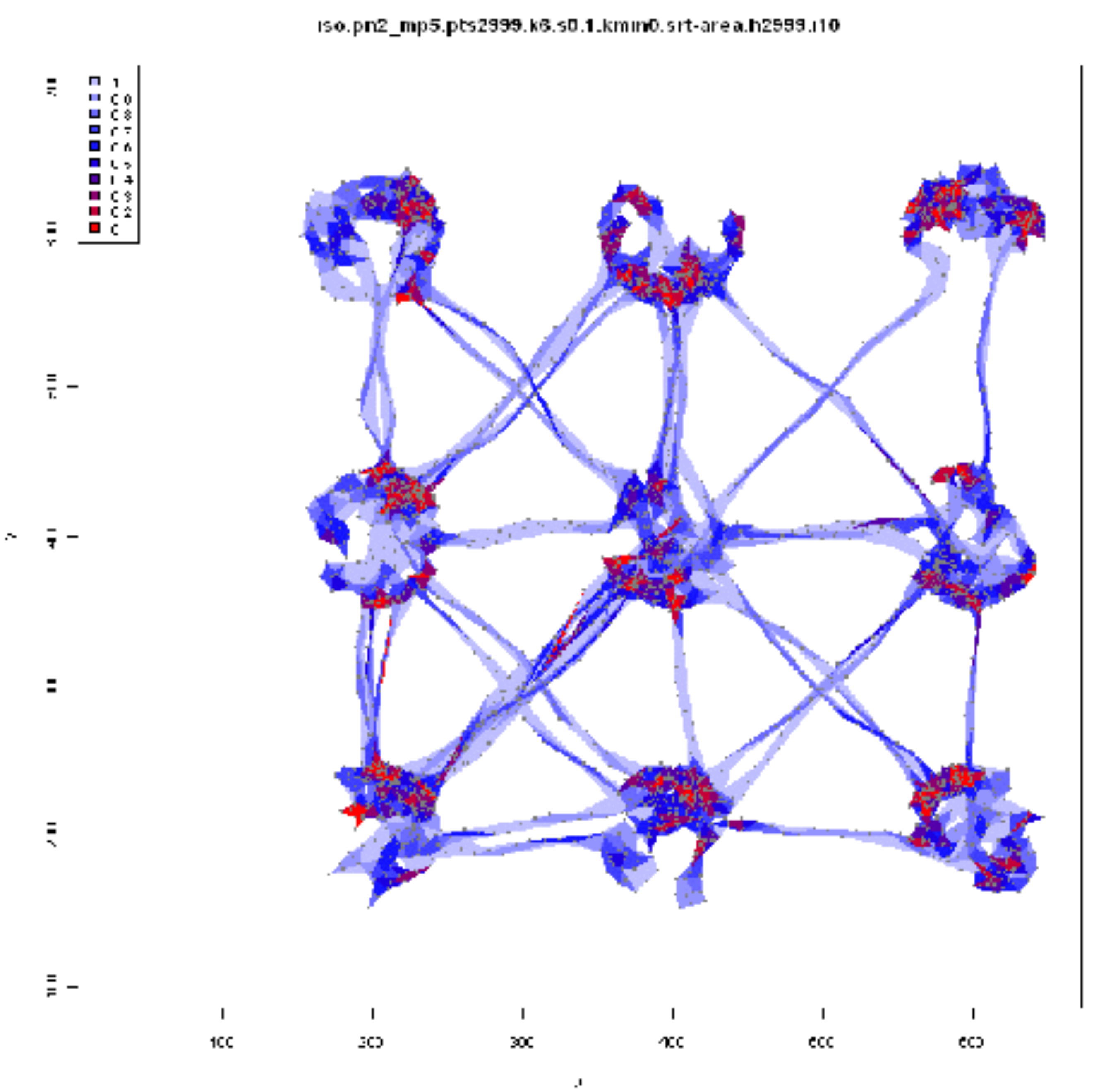
s = 0.1 **s = 0**

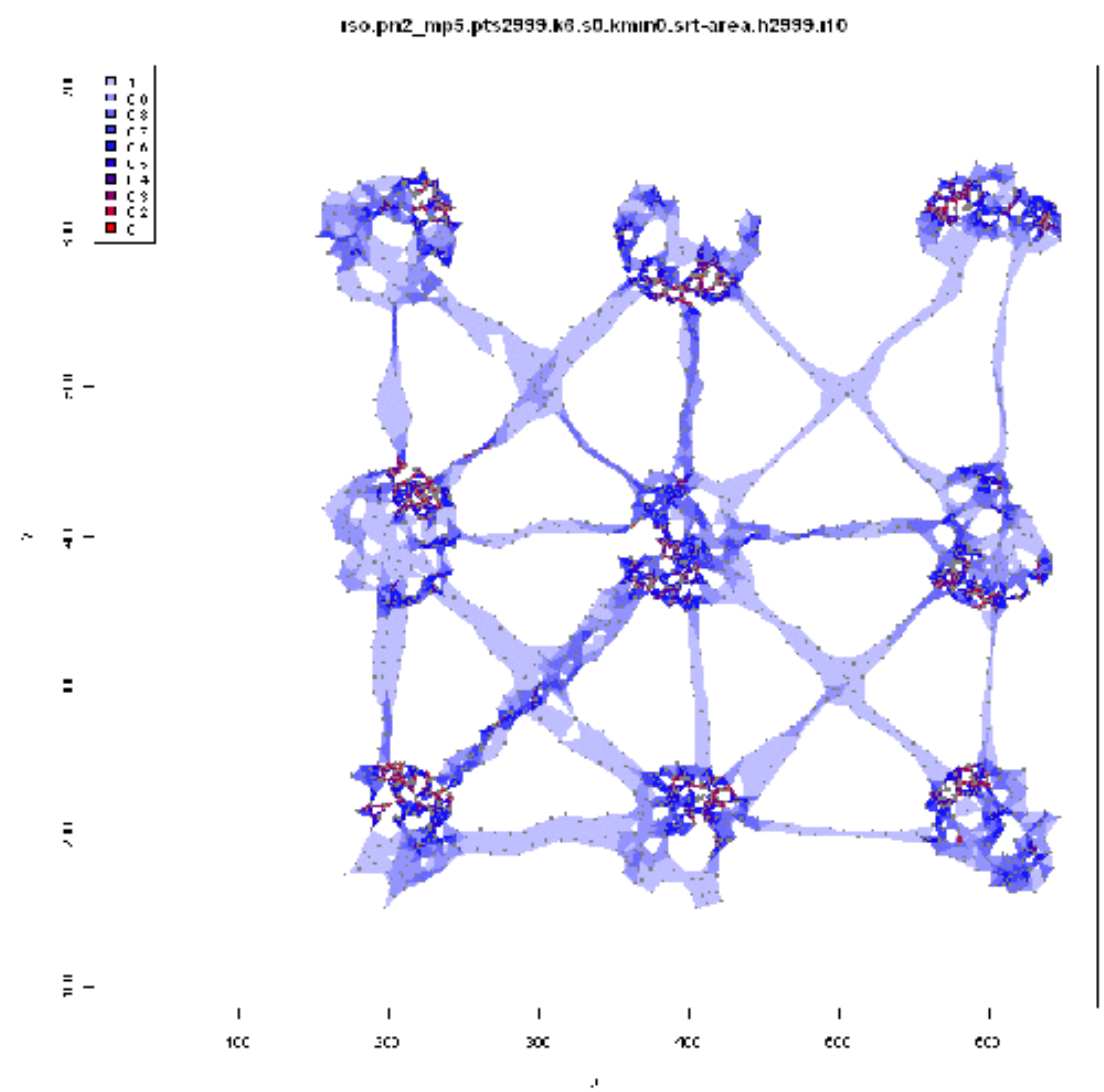
Isopleth level indicates the proportion of total points enclosed along a gradient of point density (red highest density, light blue lowest).

With Time

k = 6

Without Time



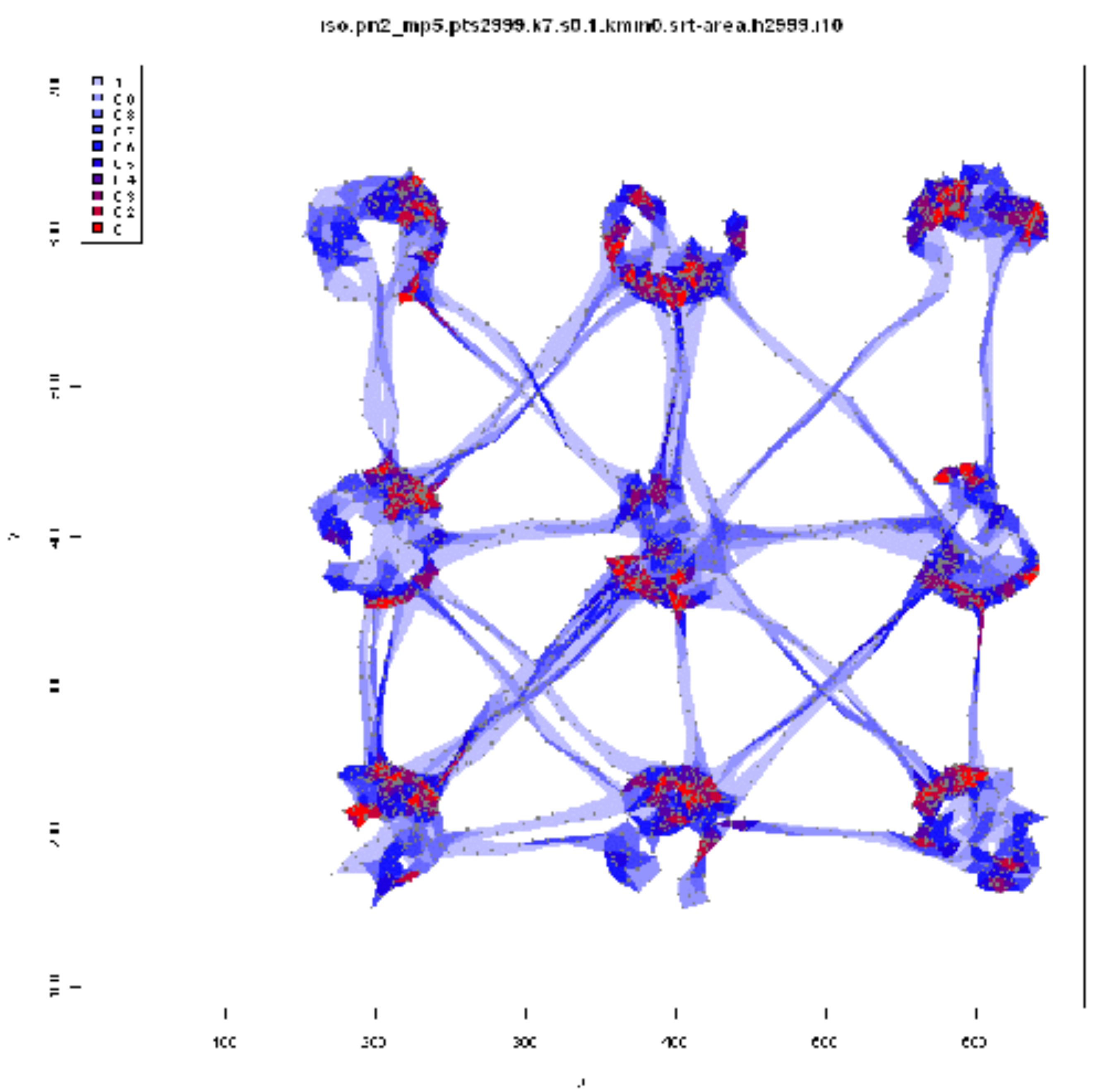


s = 0.1

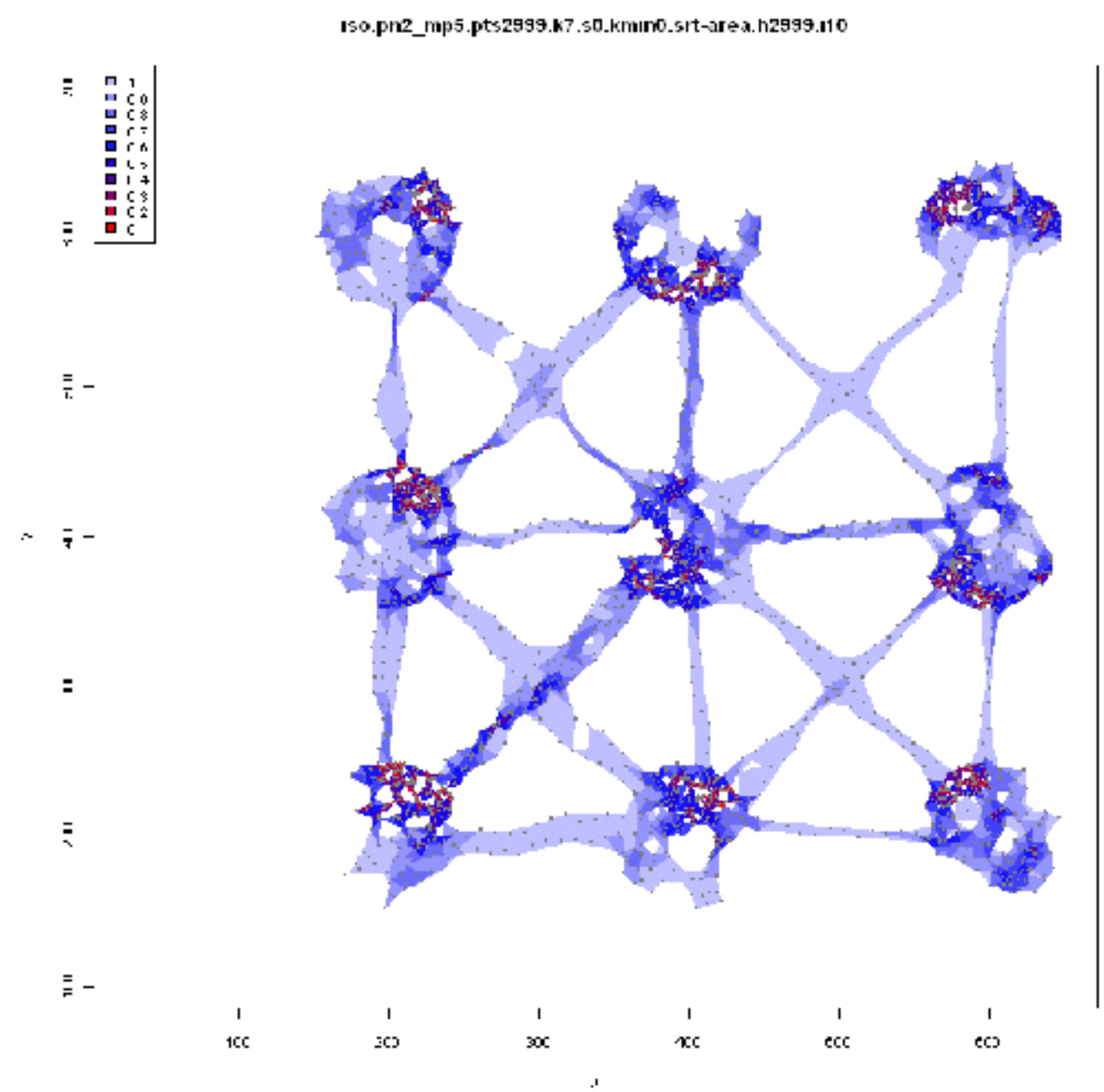
s = 0

Isopleth level indicates the proportion of total points enclosed along a gradient of point density (red highest density, light blue lowest).

With Time k = 7 Without Time



s = 0.1



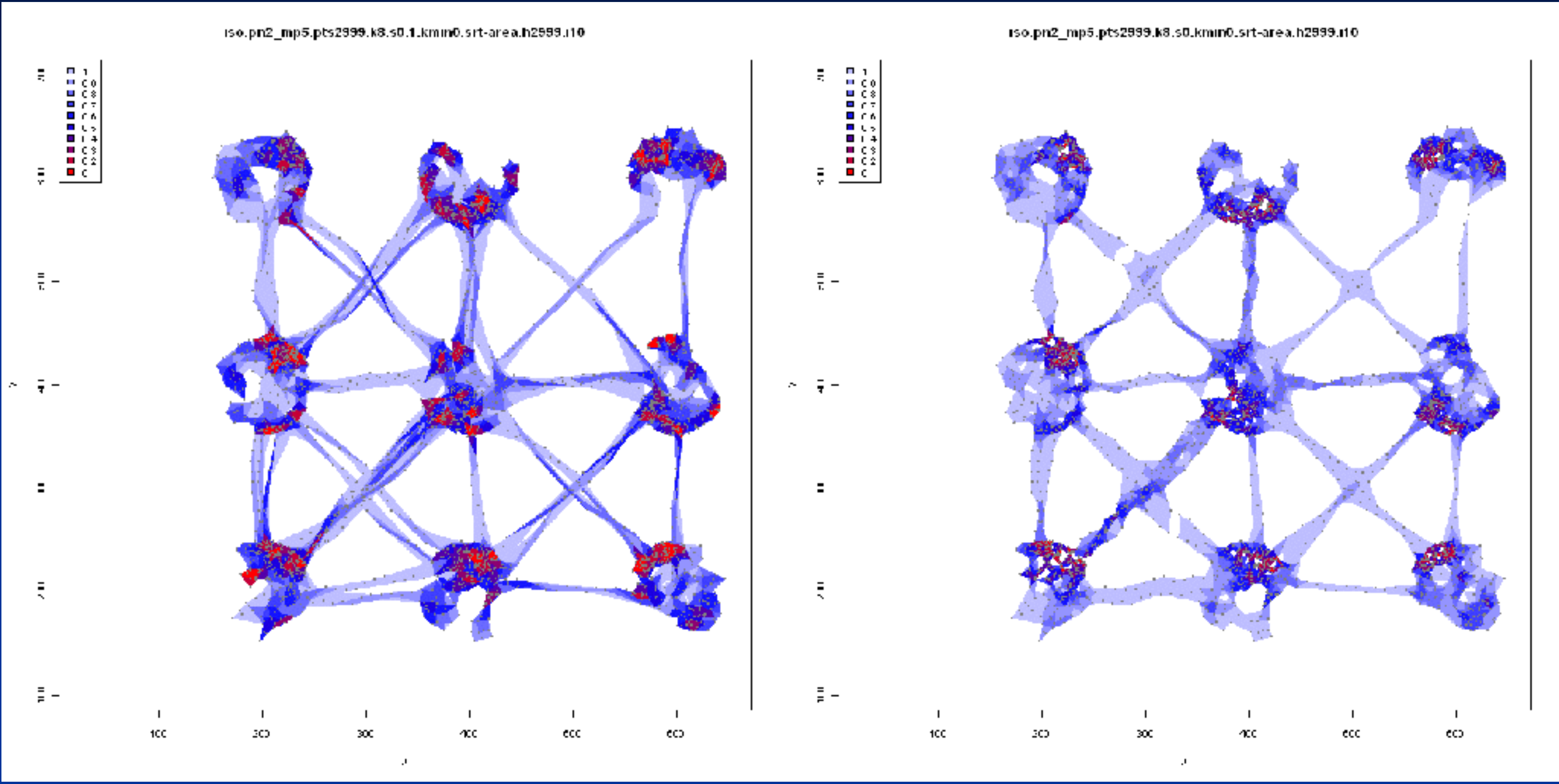
s = 0

Isopleth level indicates the proportion of total points enclosed along a gradient of point density (red highest density, light blue lowest).

With Time

k = 8

Without Time



s = 0.1

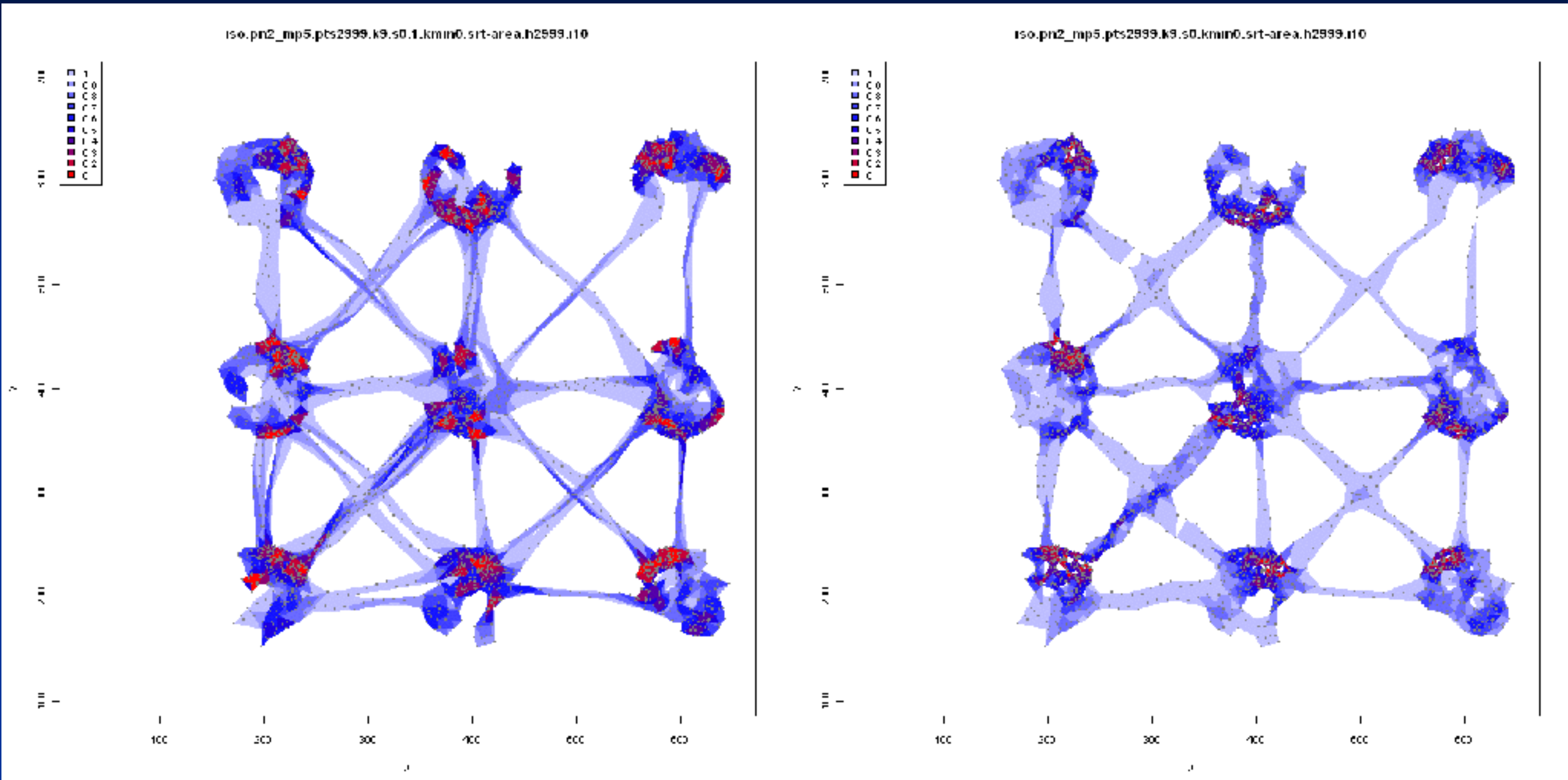
s = 0

Isopleth level indicates the proportion of total points enclosed along a gradient of point density (red highest density, light blue lowest).

With Time

k = 9

Without Time



s = 0.1

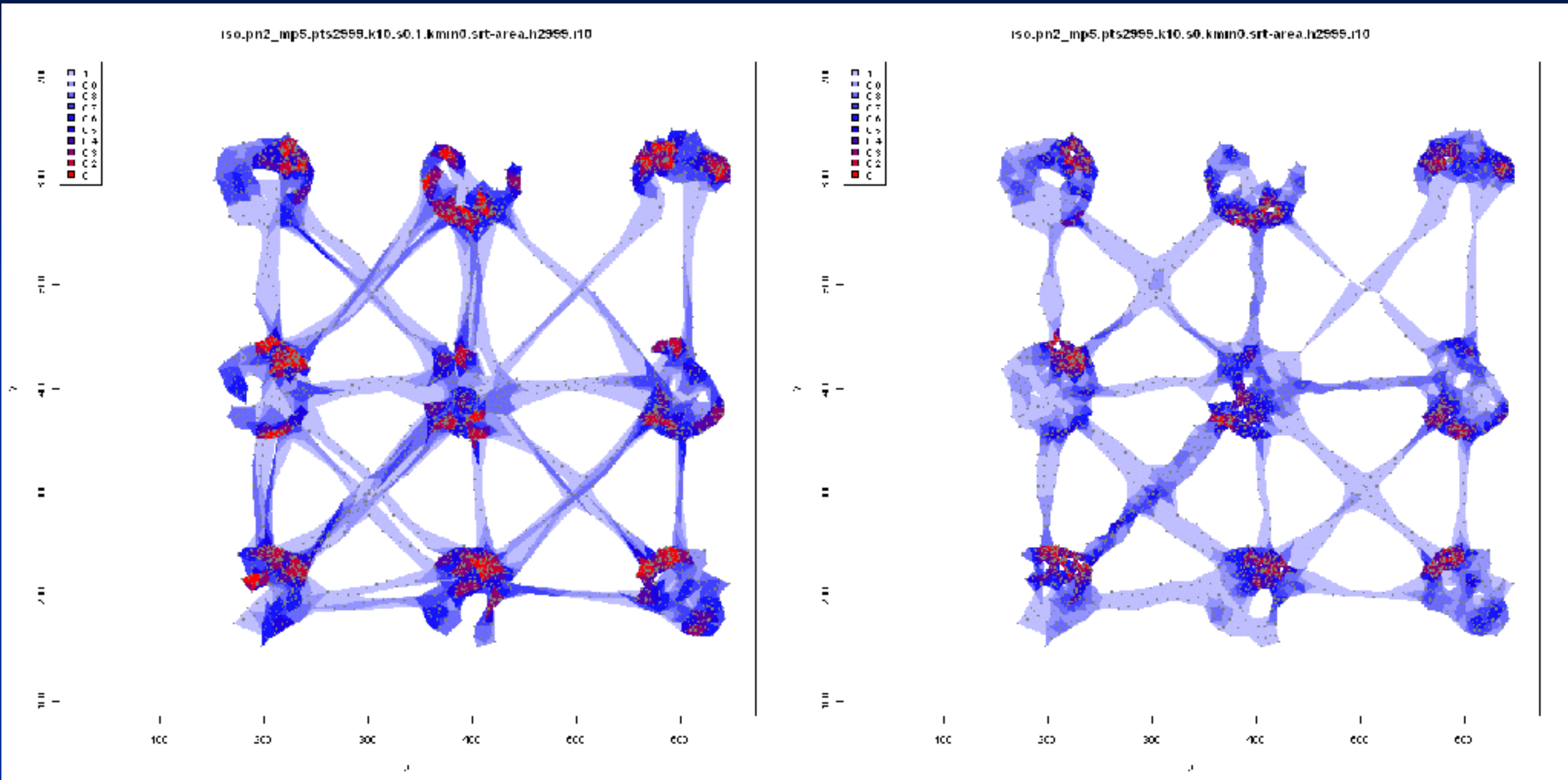
s = 0

Isopleth level indicates the proportion of total points enclosed along a gradient of point density (red highest density, light blue lowest).

With Time

k = 10

Without Time

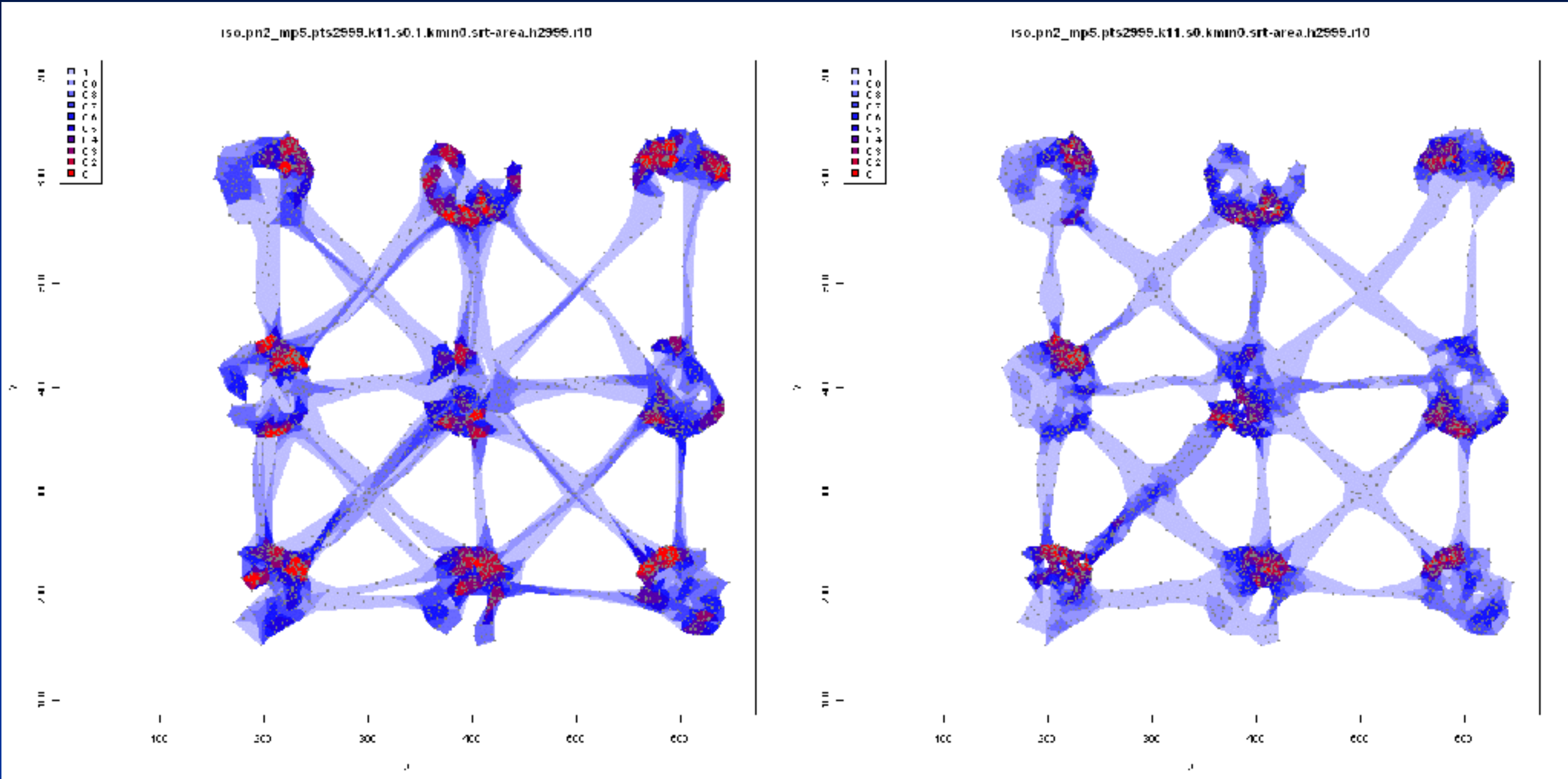


s = 0.1

s = 0

Isopleth level indicates the proportion of total points enclosed along a gradient of point density (red highest density, light blue lowest).

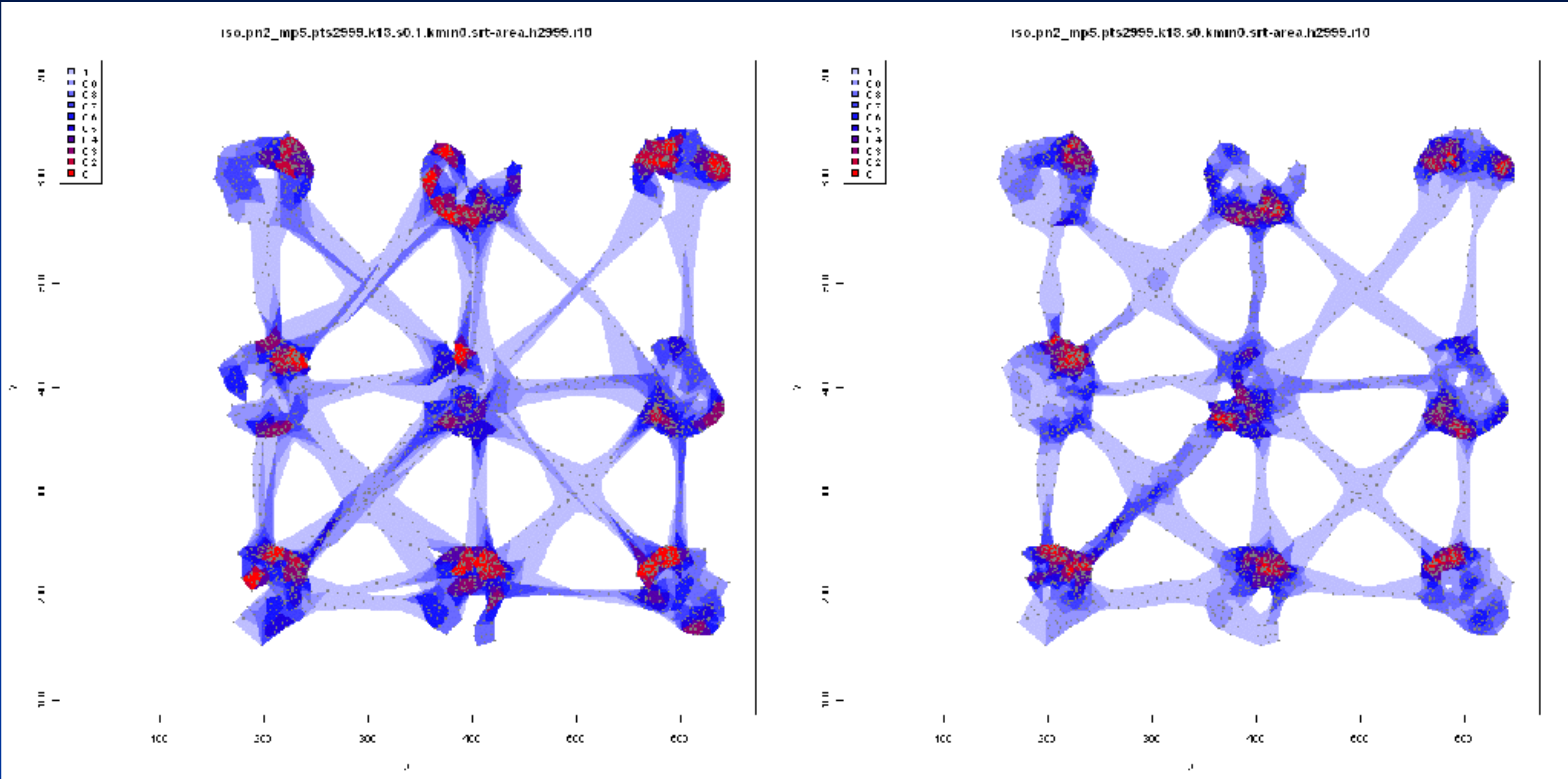
With Time — k = 11 — Without Time



s = 0.1 — **s = 0**

Isopleth level indicates the proportion of total points enclosed along a gradient of point density (red highest density, light blue lowest).

With Time | k = 13 | Without Time



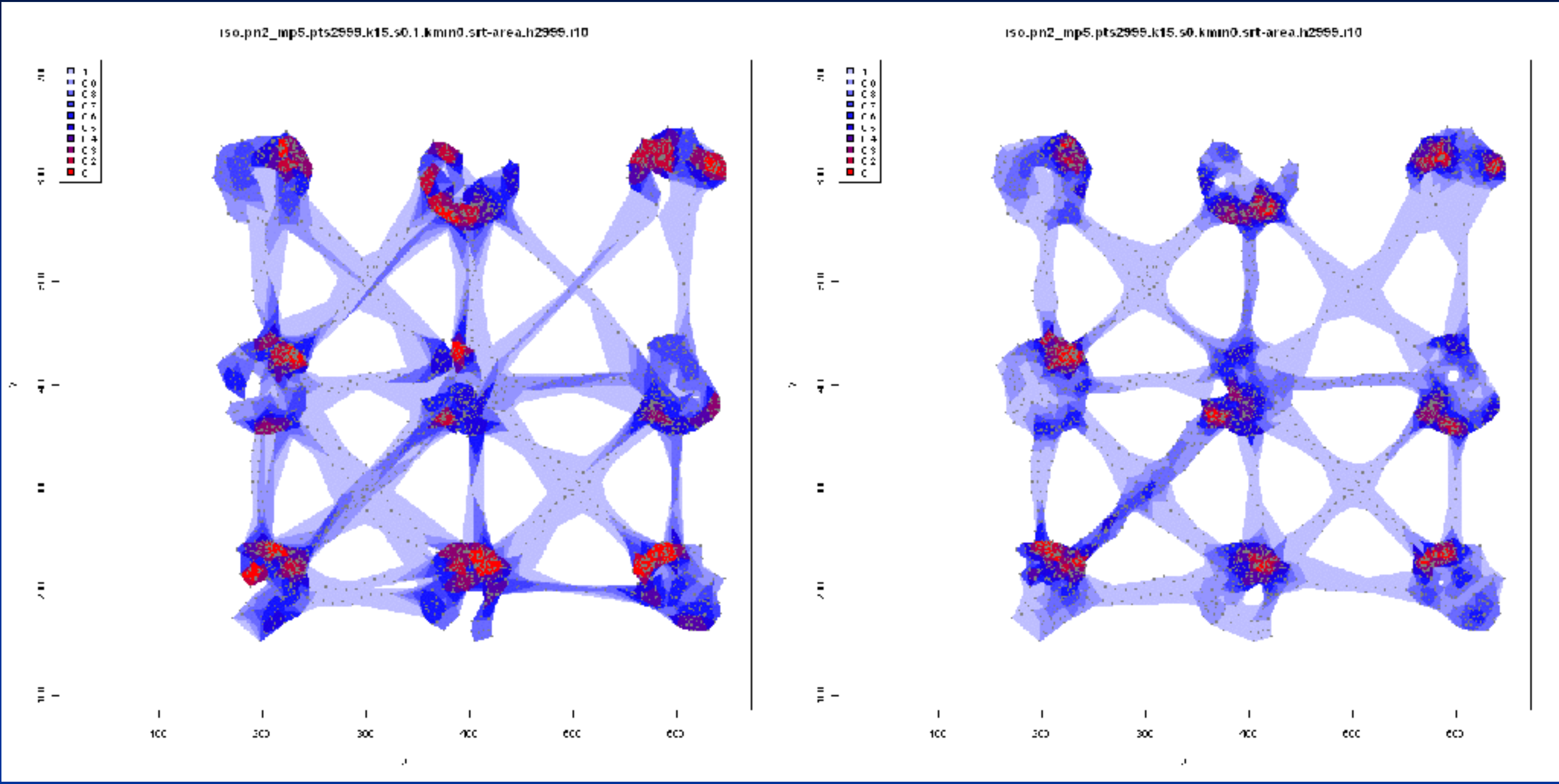
s = 0.1 | s = 0

Isopleth level indicates the proportion of total points enclosed along a gradient of point density (red highest density, light blue lowest).

With Time

k = 15

Without Time

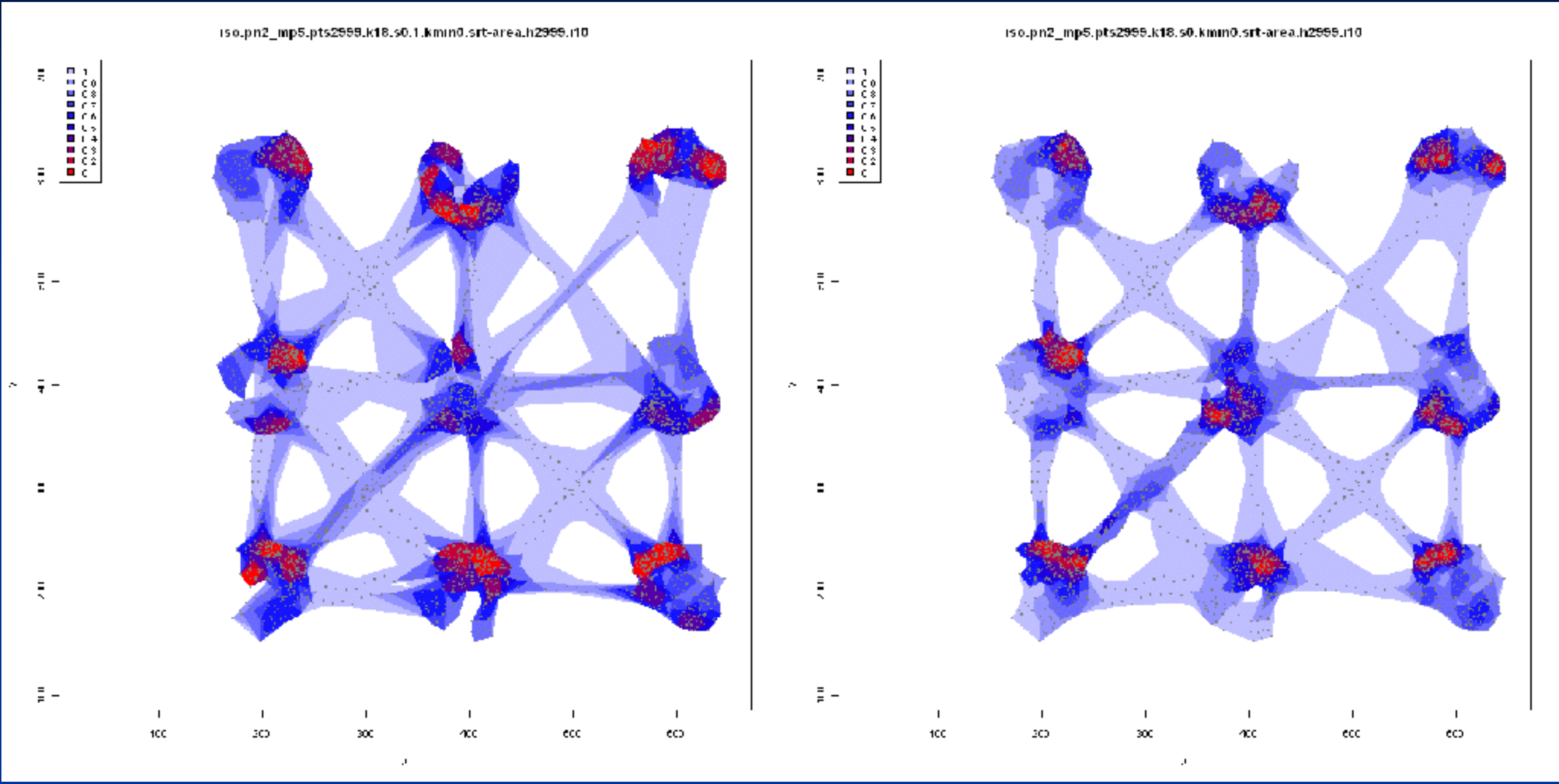


s = 0.1

s = 0

Isopleth level indicates the proportion of total points enclosed along a gradient of point density (red highest density, light blue lowest).

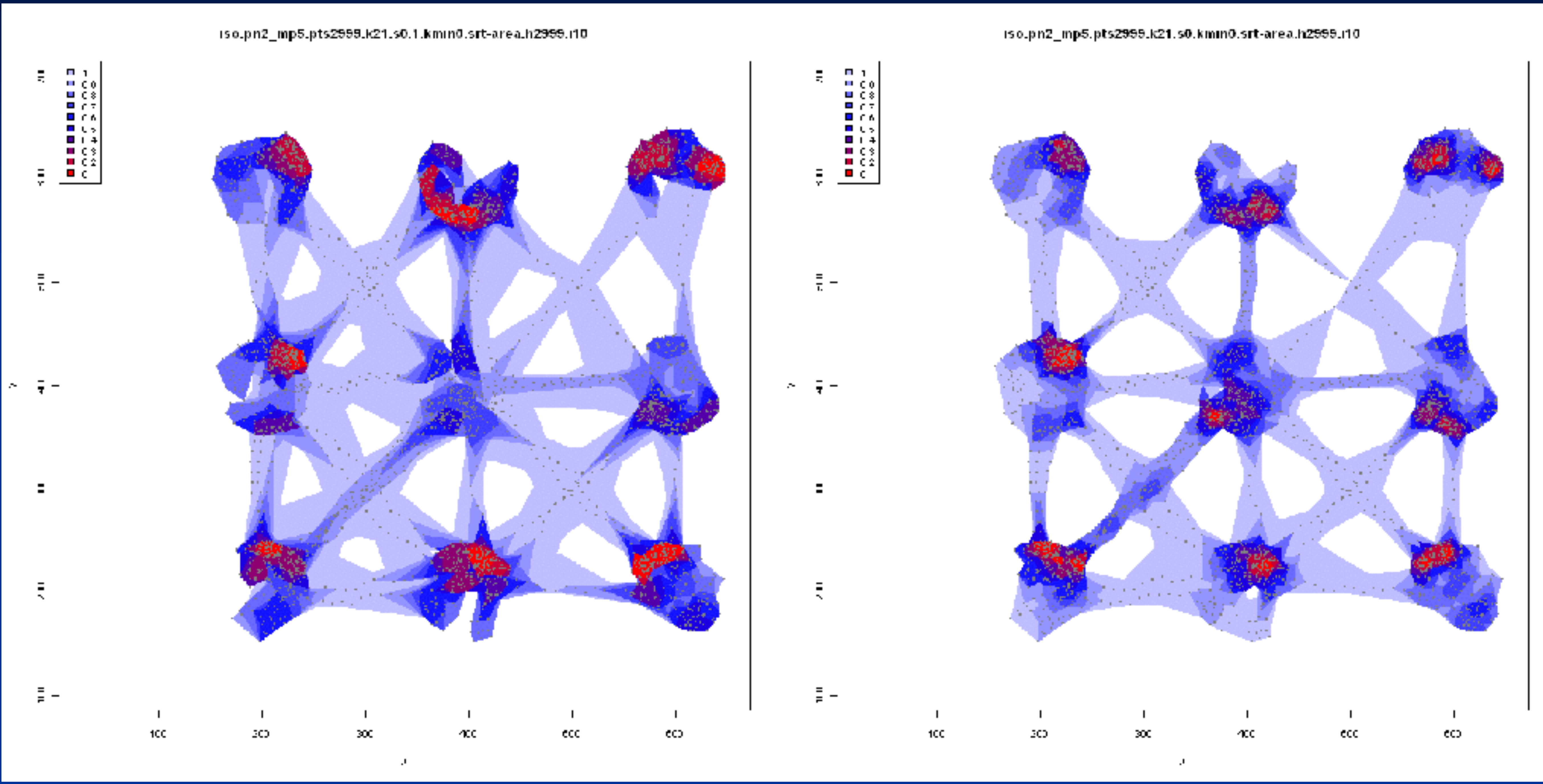
With Time | k = 18 | Without Time



s = 0.1 | s = 0

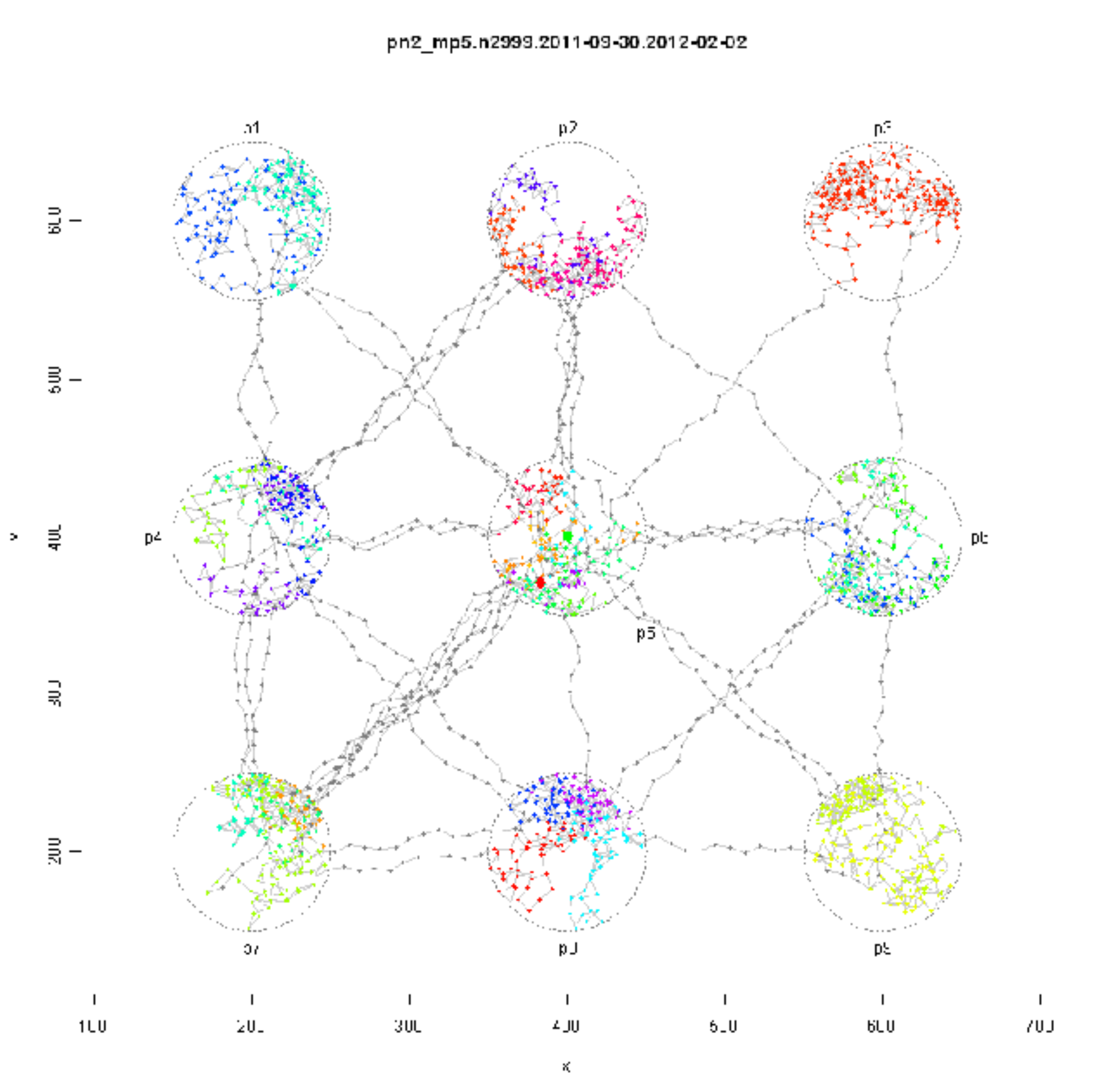
Isopleth level indicates the proportion of total points enclosed along a gradient of point density (red highest density, light blue lowest).

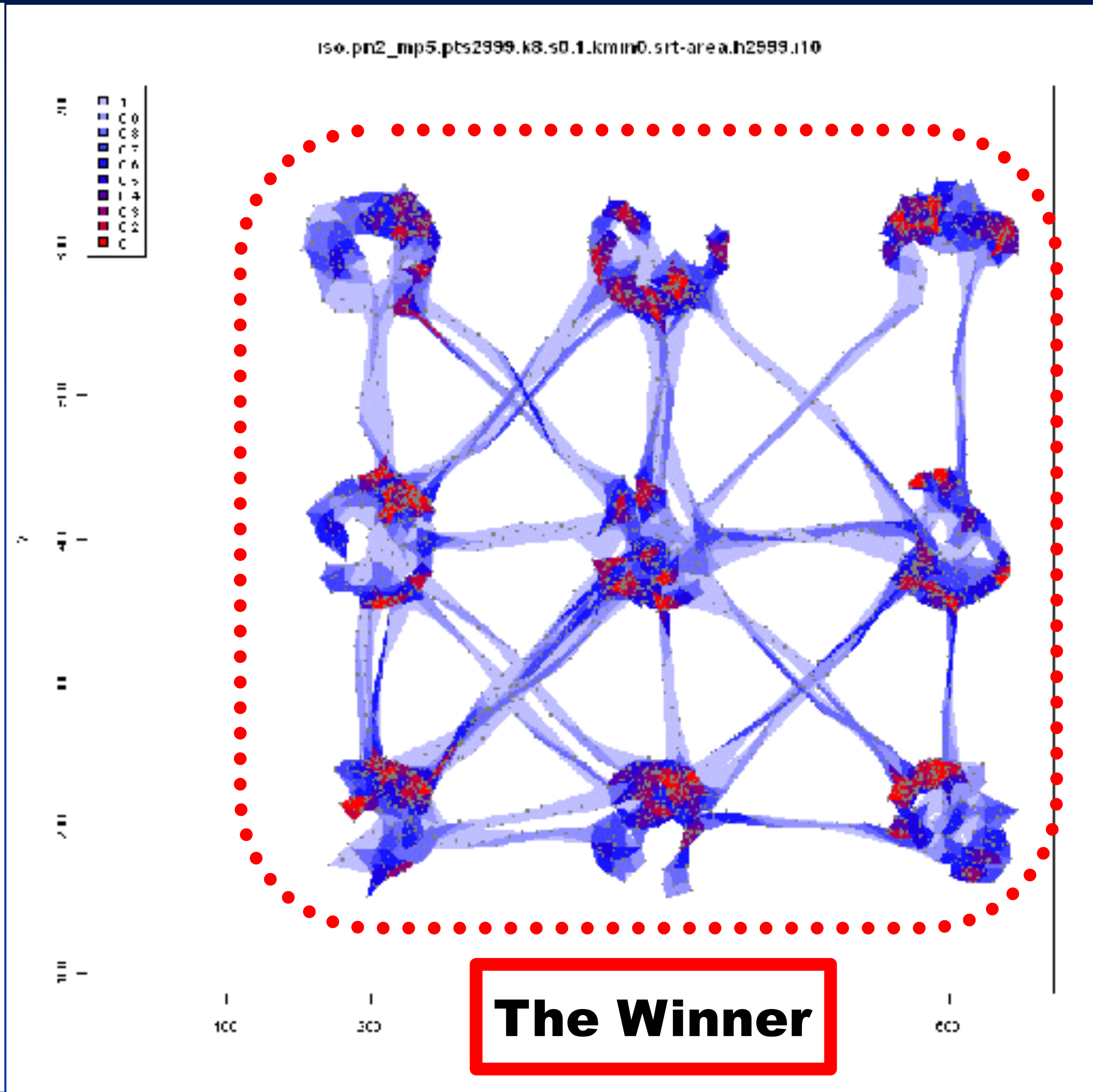
With Time | k = 21 | Without Time



s = 0.1 | s = 0

Isopleth level indicates the proportion of total points enclosed along a gradient of point density (red highest density, light blue lowest).

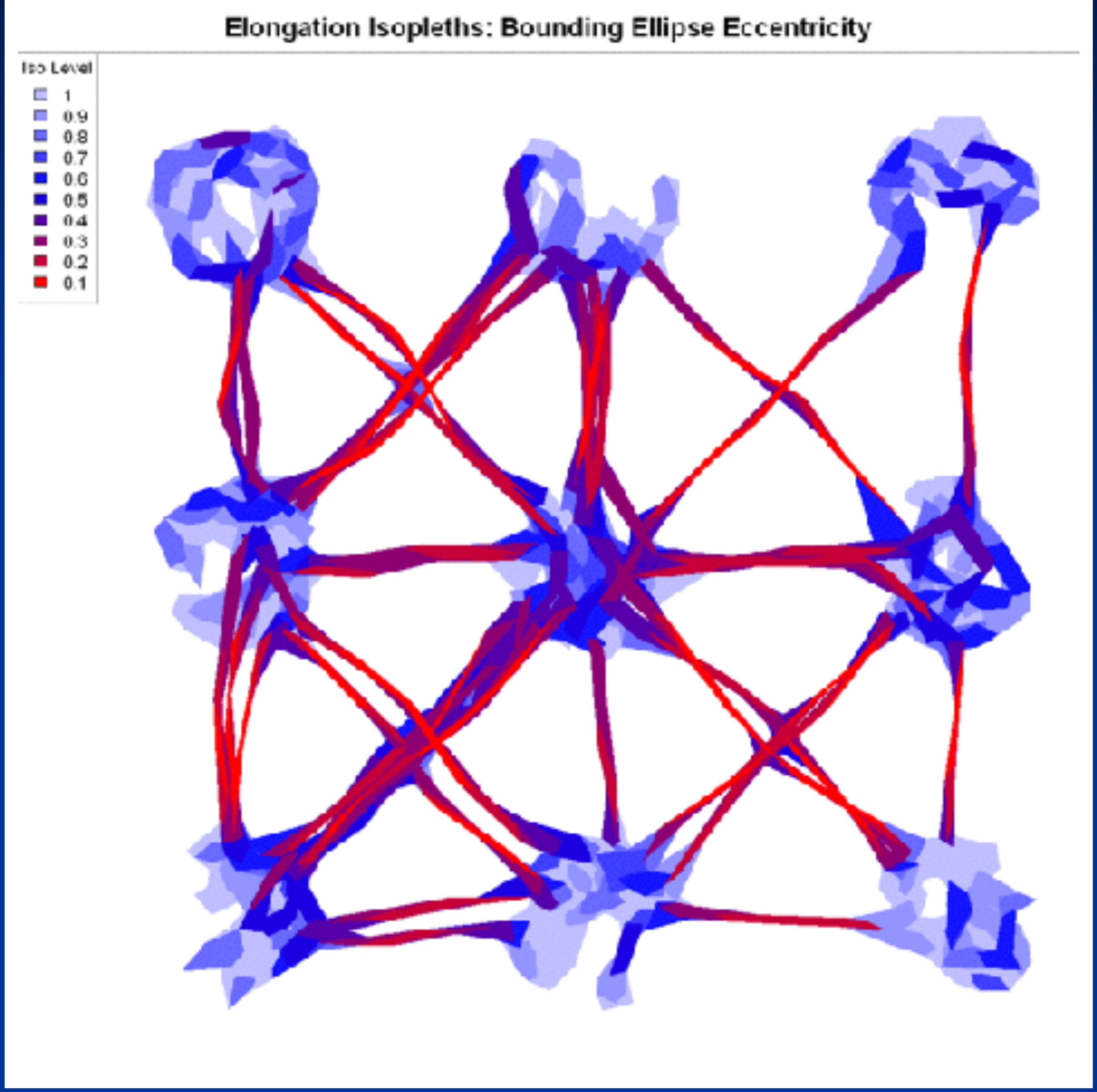


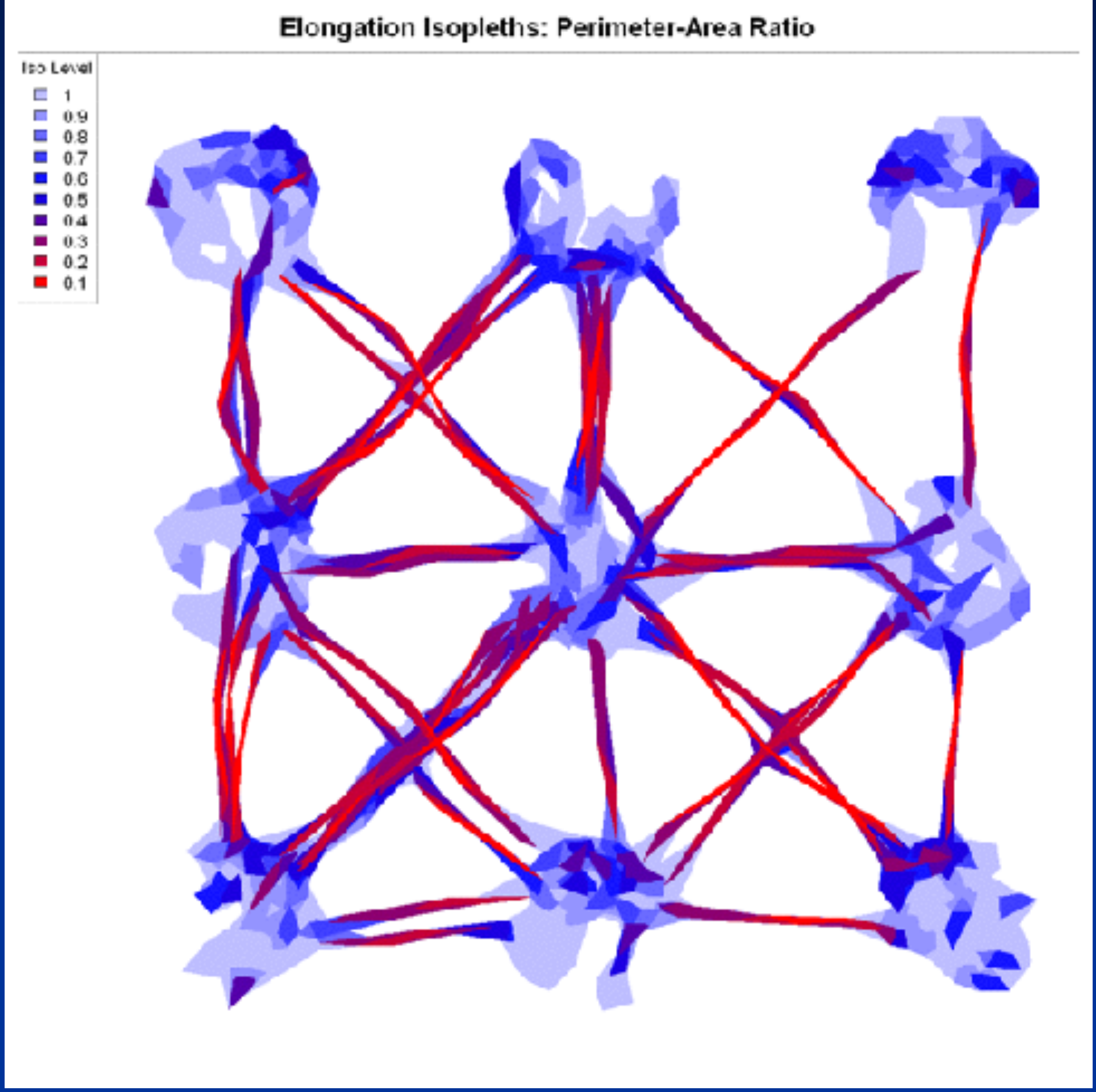


s = 0.1

k = 8

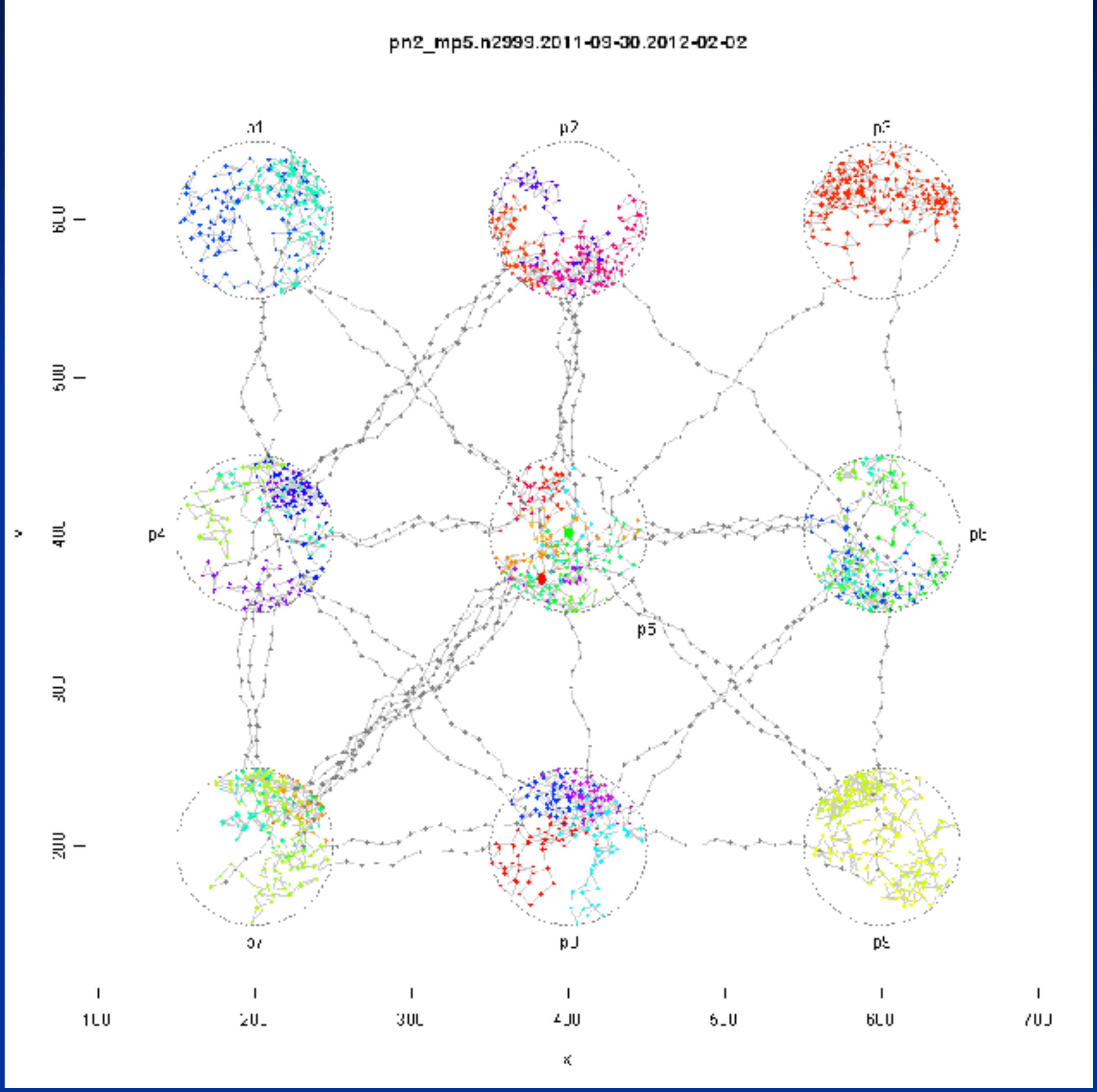
Simulated Data:
Elongation Isopleths

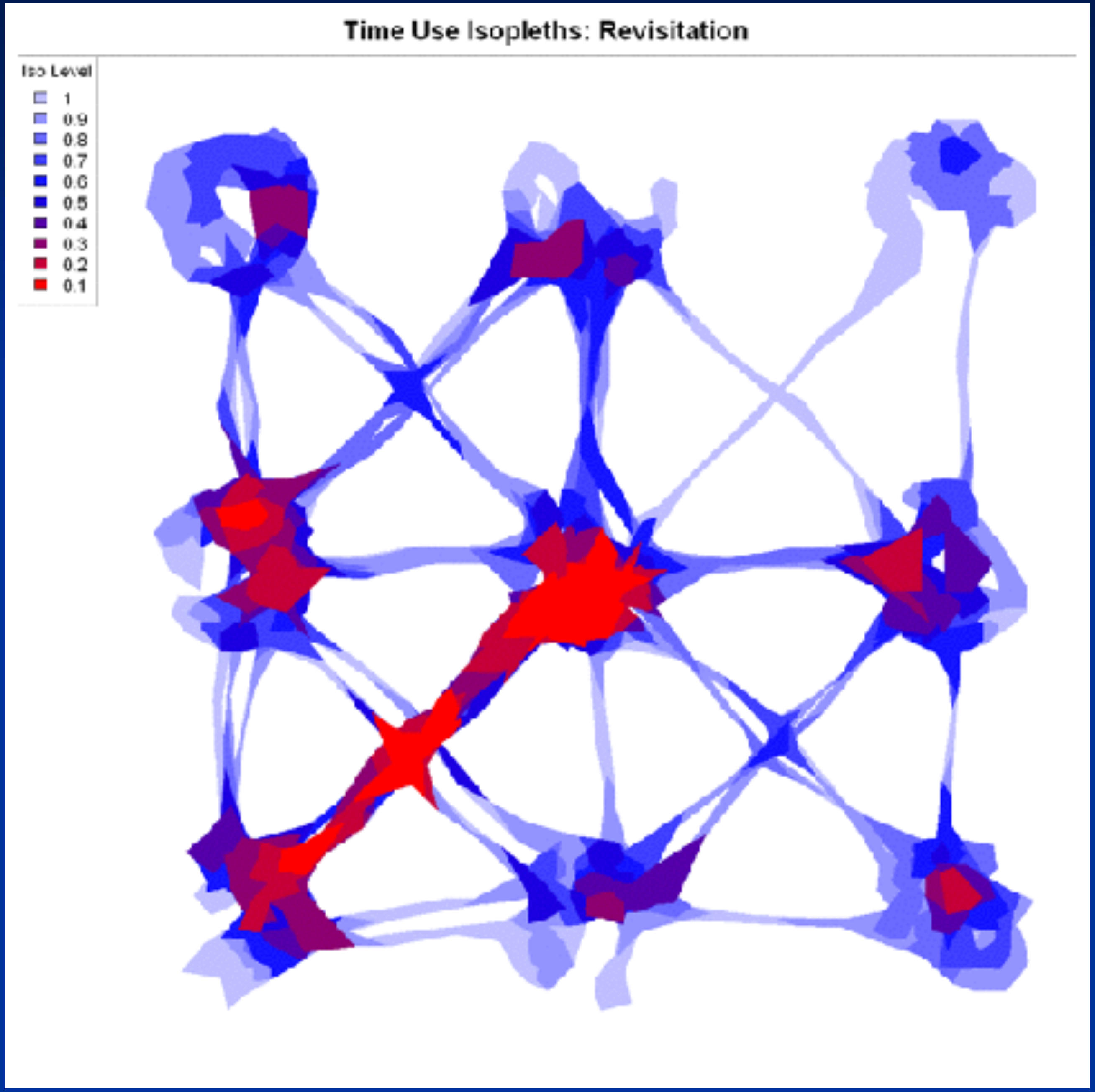




Isopleth levels indicate the proportion of total points enclosed along a gradient of elongation (red most elongated, light-blue least). Hulls sorted by eccentricity of bounding ellipse (left) and perimeter / area ratio (right). Both did a good job identifying the areas of directional movement. One can even see trails within patches when the individual was told ‘it’s time to go’.

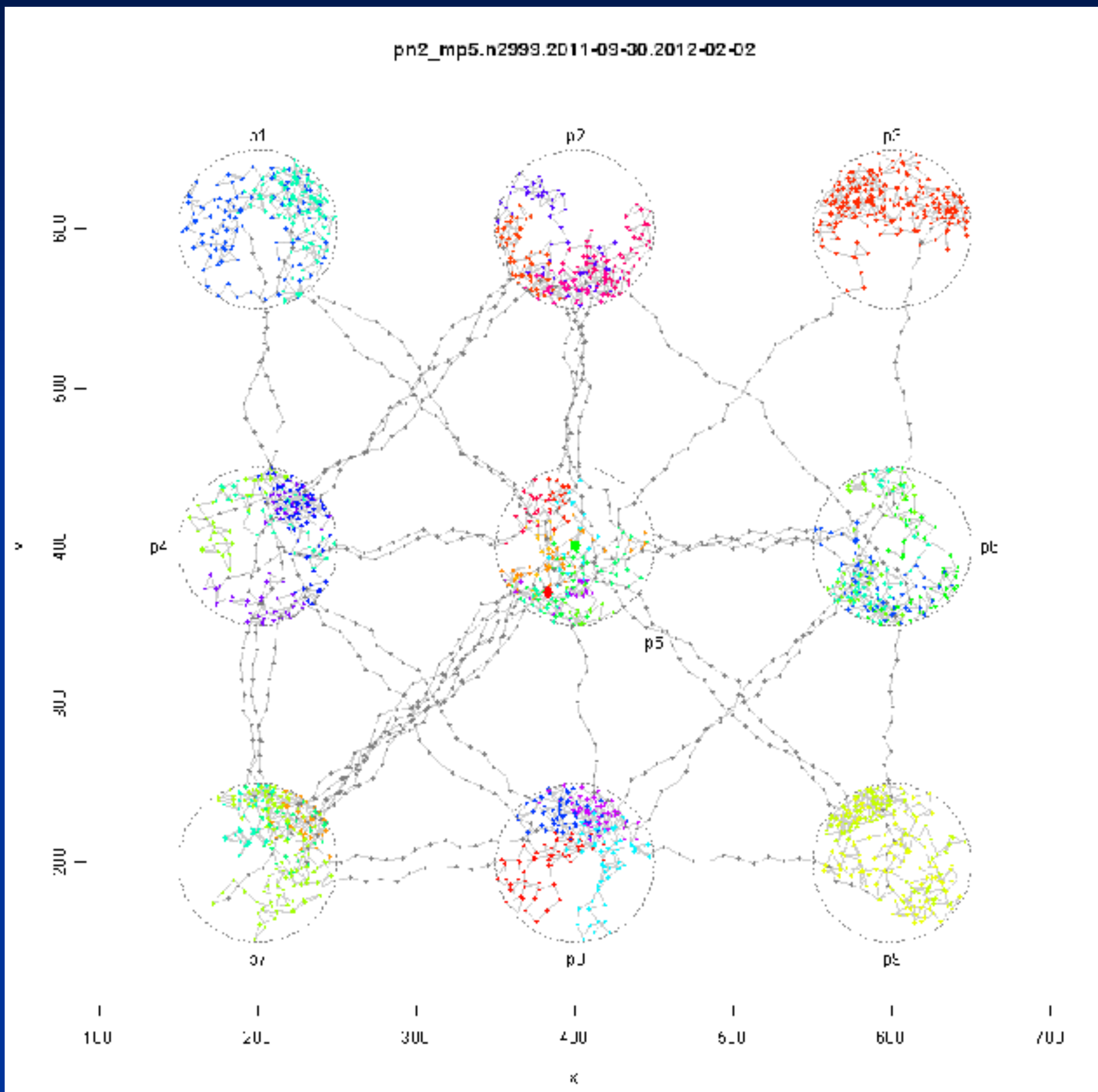
Simulated Data:
Revisitation Isopleths

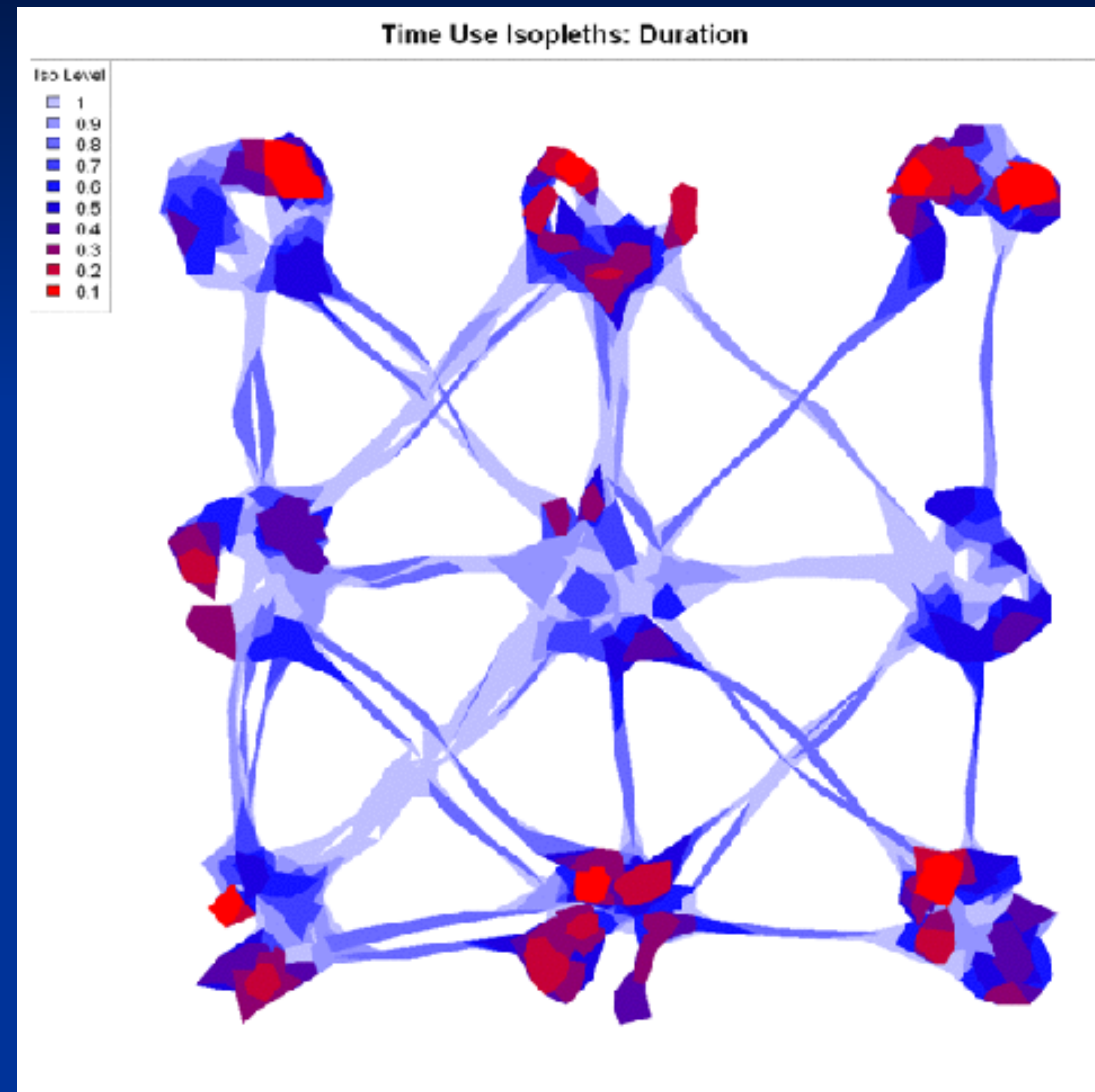




Hulls sorted by number of separate visits (inter-visit gap = 24 time steps). Hulls most revisited were found in the center patch (revisited more than any other patch) and the “superhighway” between patch five and seven. Also the ‘foyer’ area of patches.

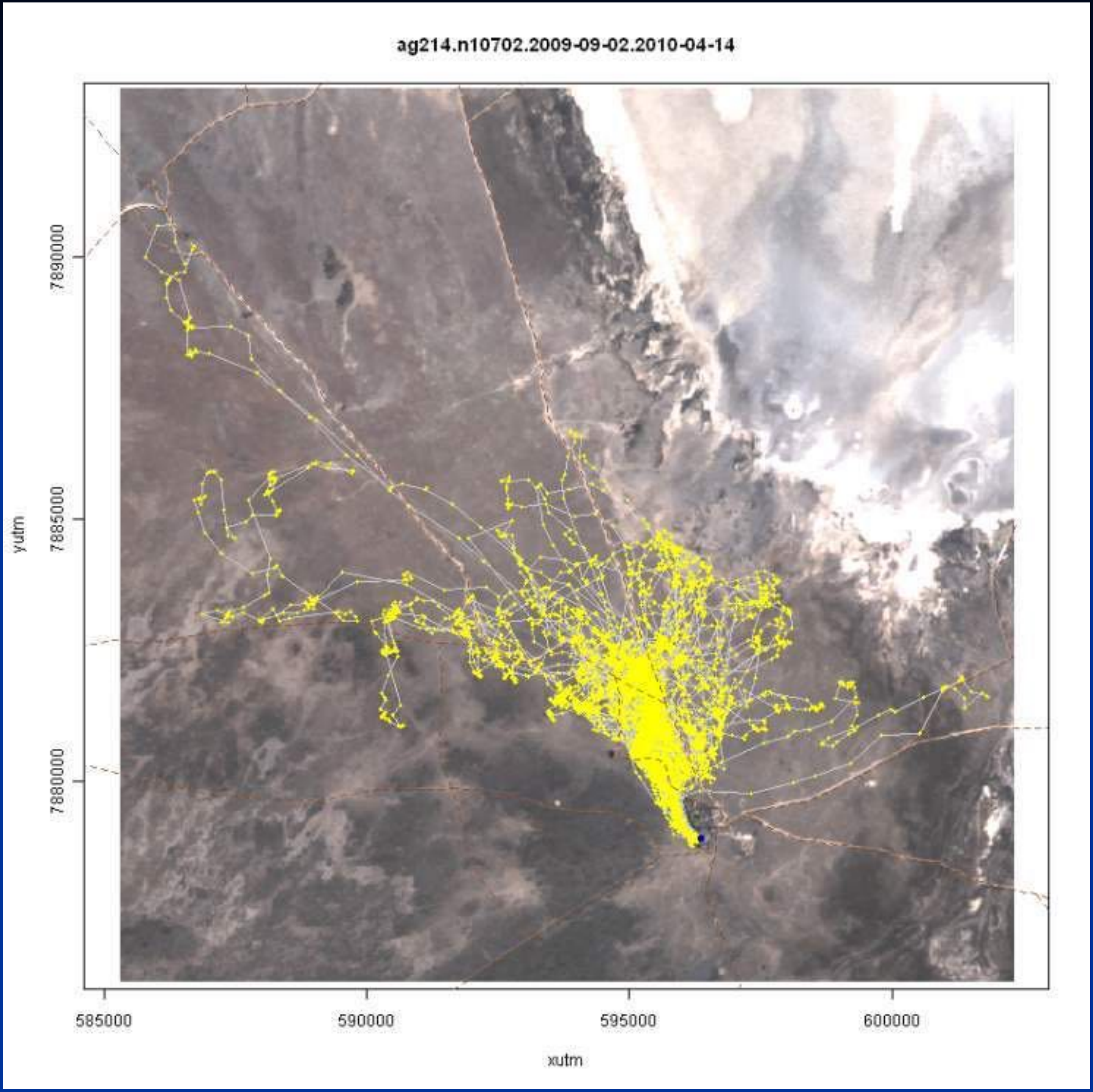
Simulated Data:
Duration Isopleths



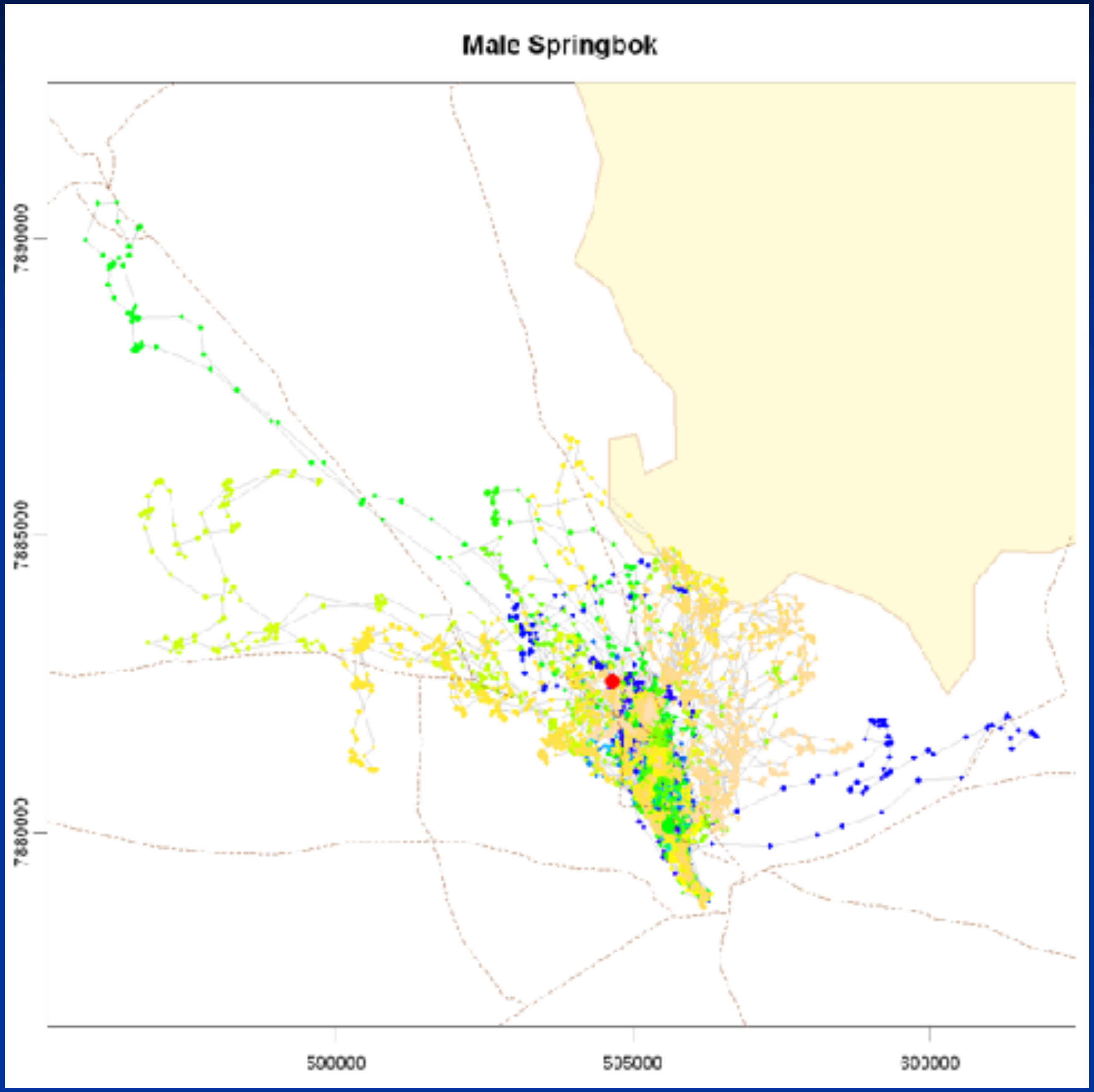


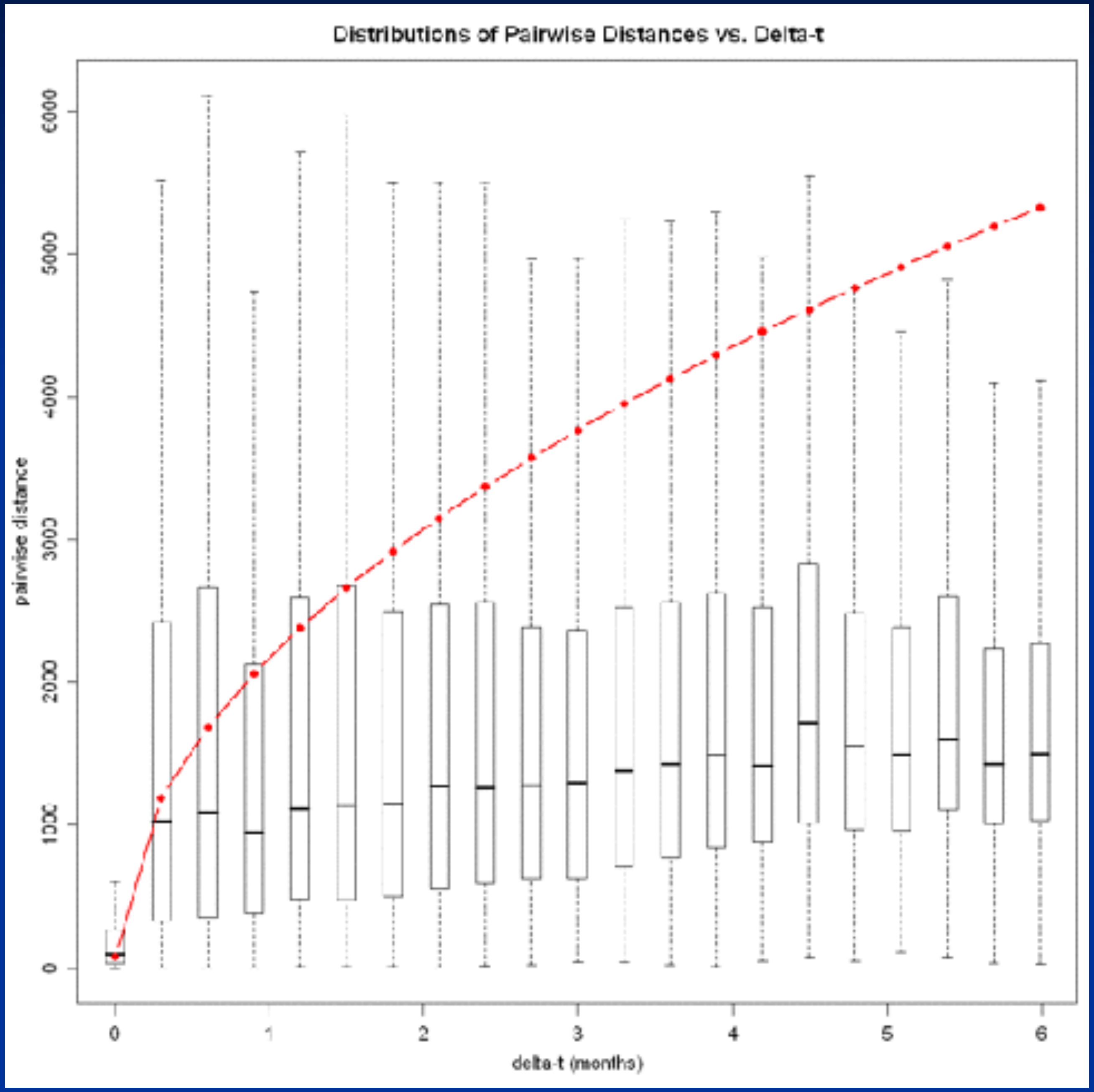
Hulls sorted by mean number of locations per visit (inter-visit gap = 24 time steps). Hulls with the longest duration were found around the edges of patches where the animal was programmed to ‘bounce back’ and got stuck. Also patch 3 where the animal remained the longest during a single visit.

territorial male



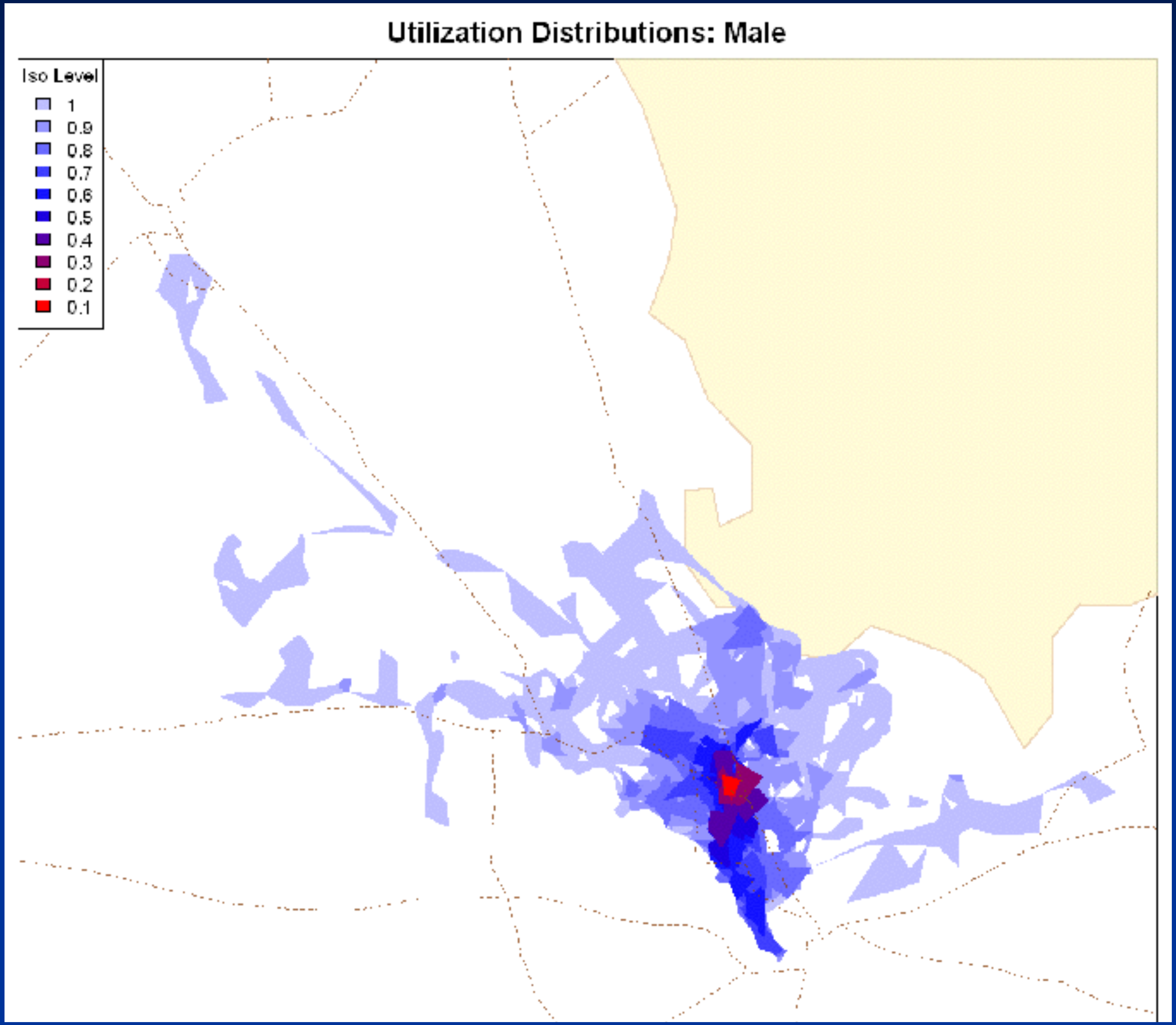
Male Springbok:
Null model fit



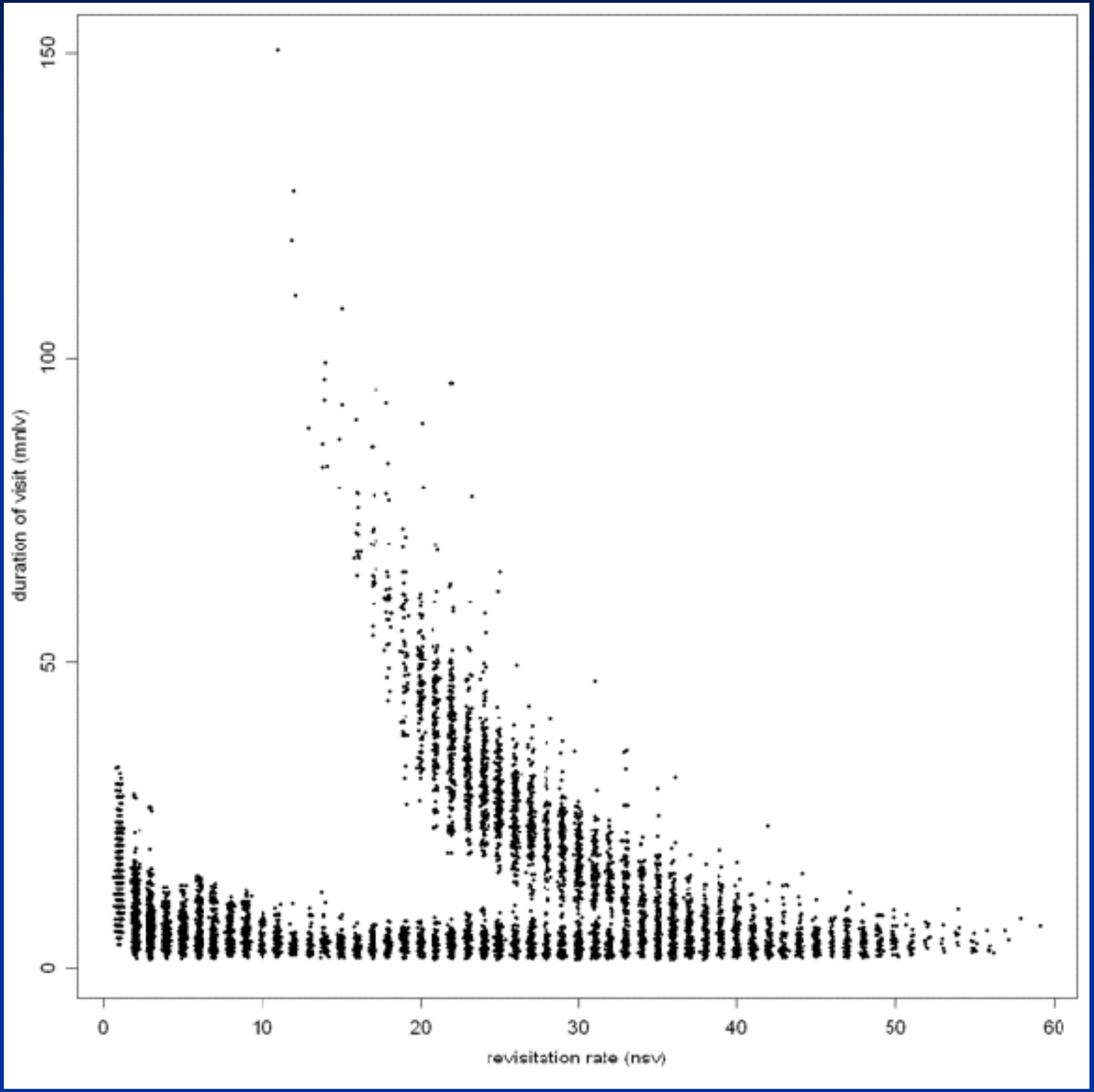


On left, maps of the male springbok in Etosha National Park, Namibia. The colors of the points reflect temporal continuity; tan lines are roads, and yellow polygons are salt pans. On the right, box plots show the distribution of the net displacement of all pairs of points sampled Δt apart (x-axis), with the predicted Gaussian diffusion distance from the random walk null model specified in Equation 2 overlaid in red.

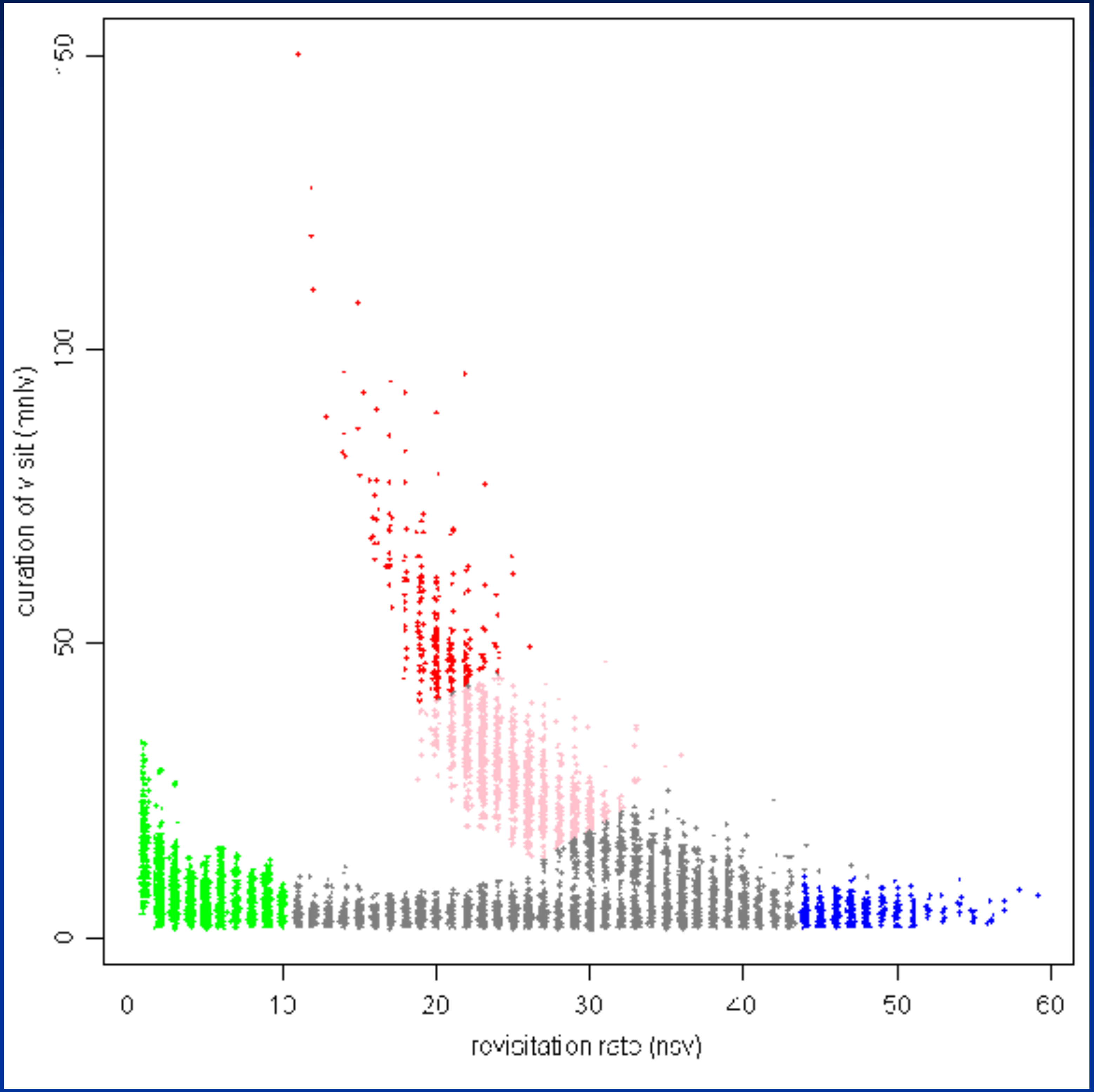
Male Springbok:
Density Isopleths



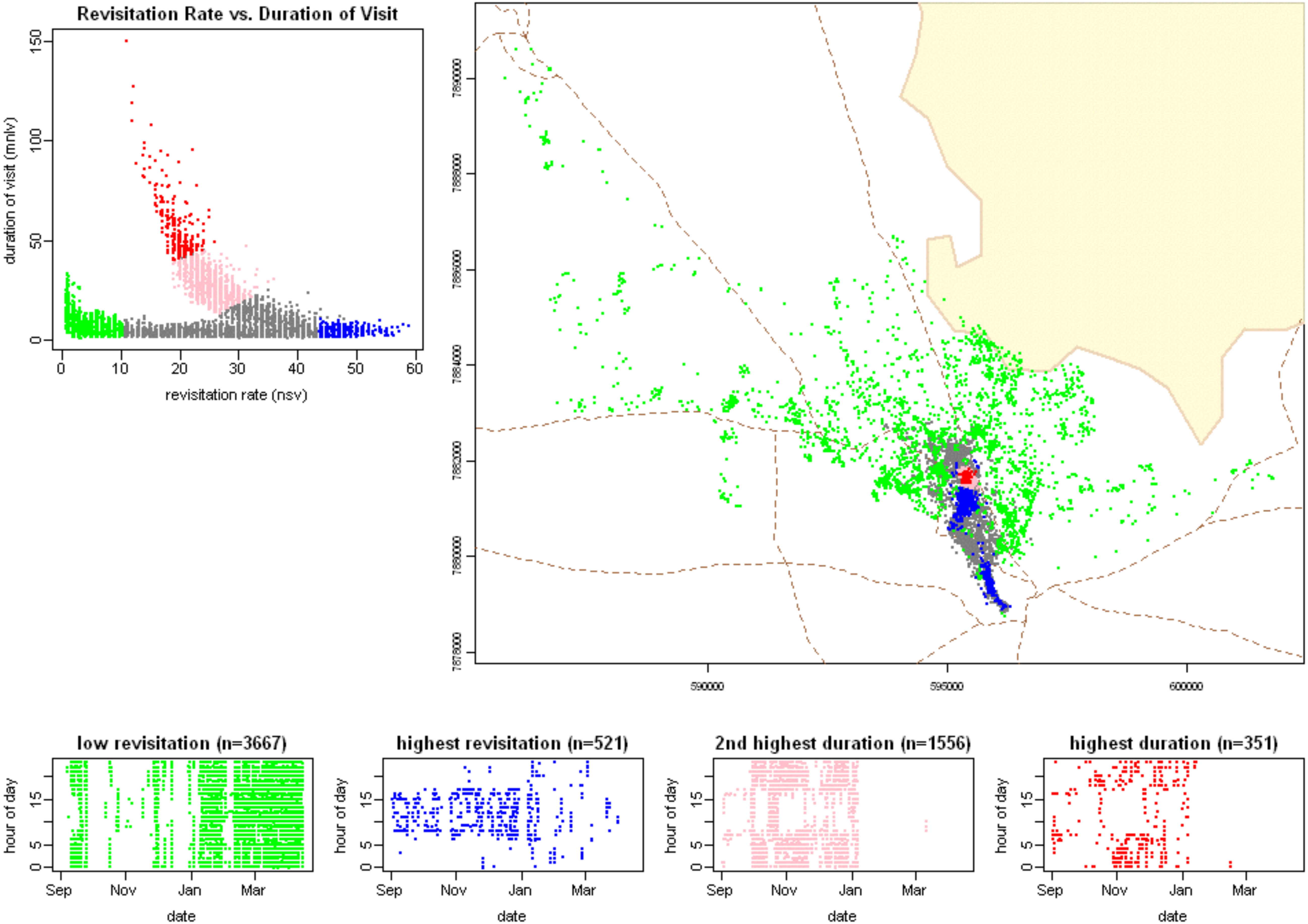
Male Springbok:
Hulls in Time-Use Space



Male Springbok:
Hulls in Time-Use Space

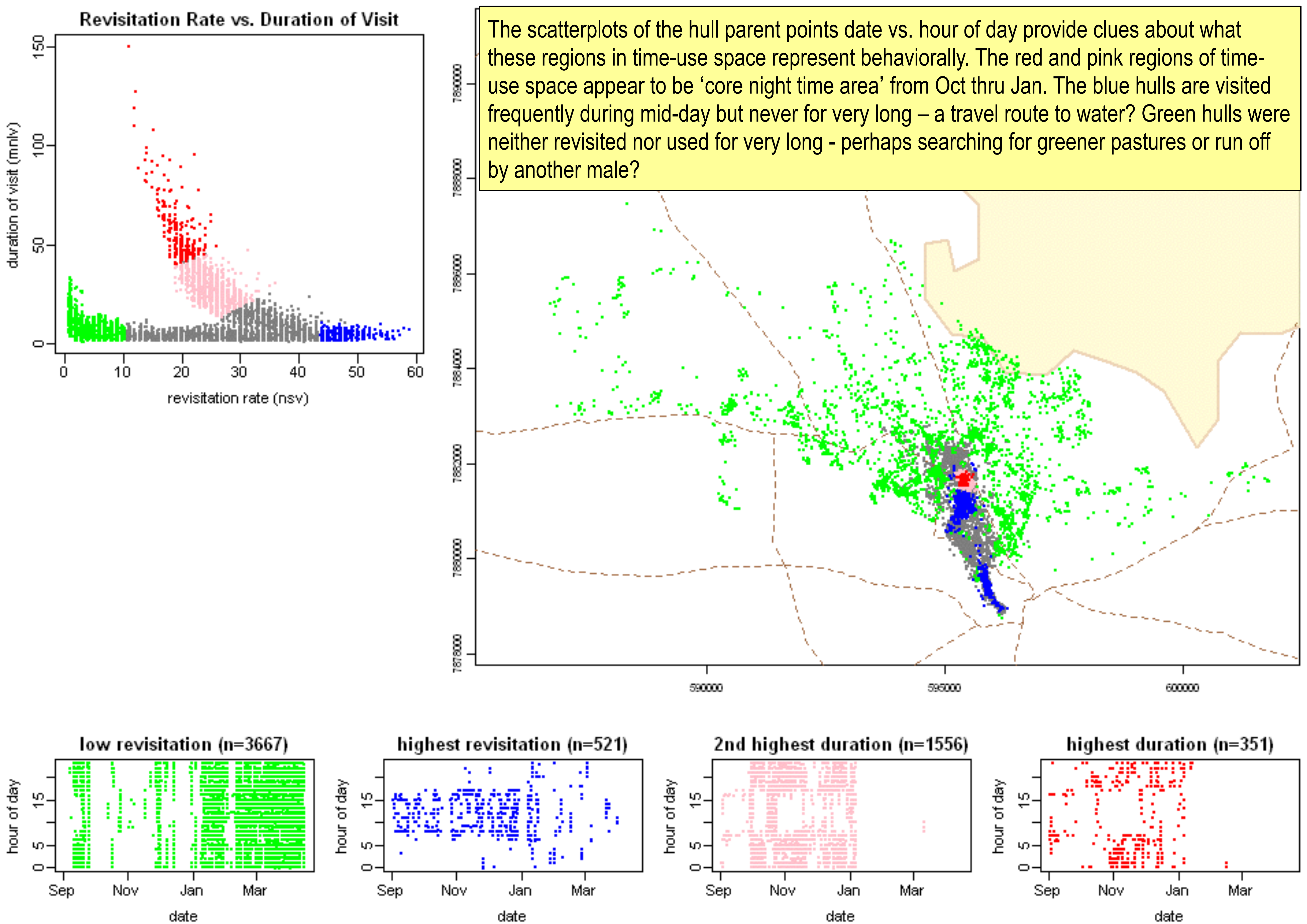


Hull Parent Points Colored by Time Use Regions

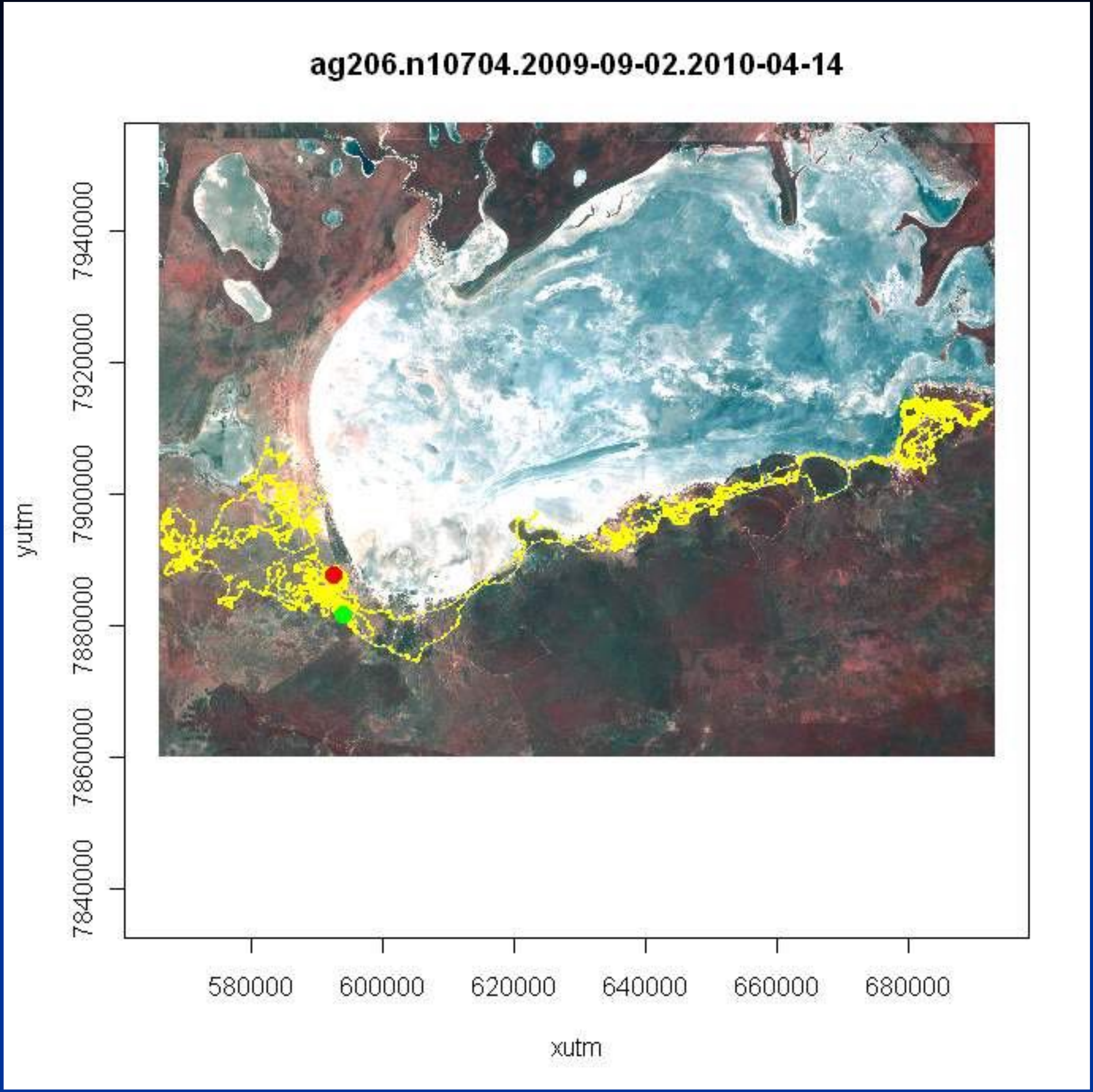


Hull Parent Points Colored by Time Use Regions

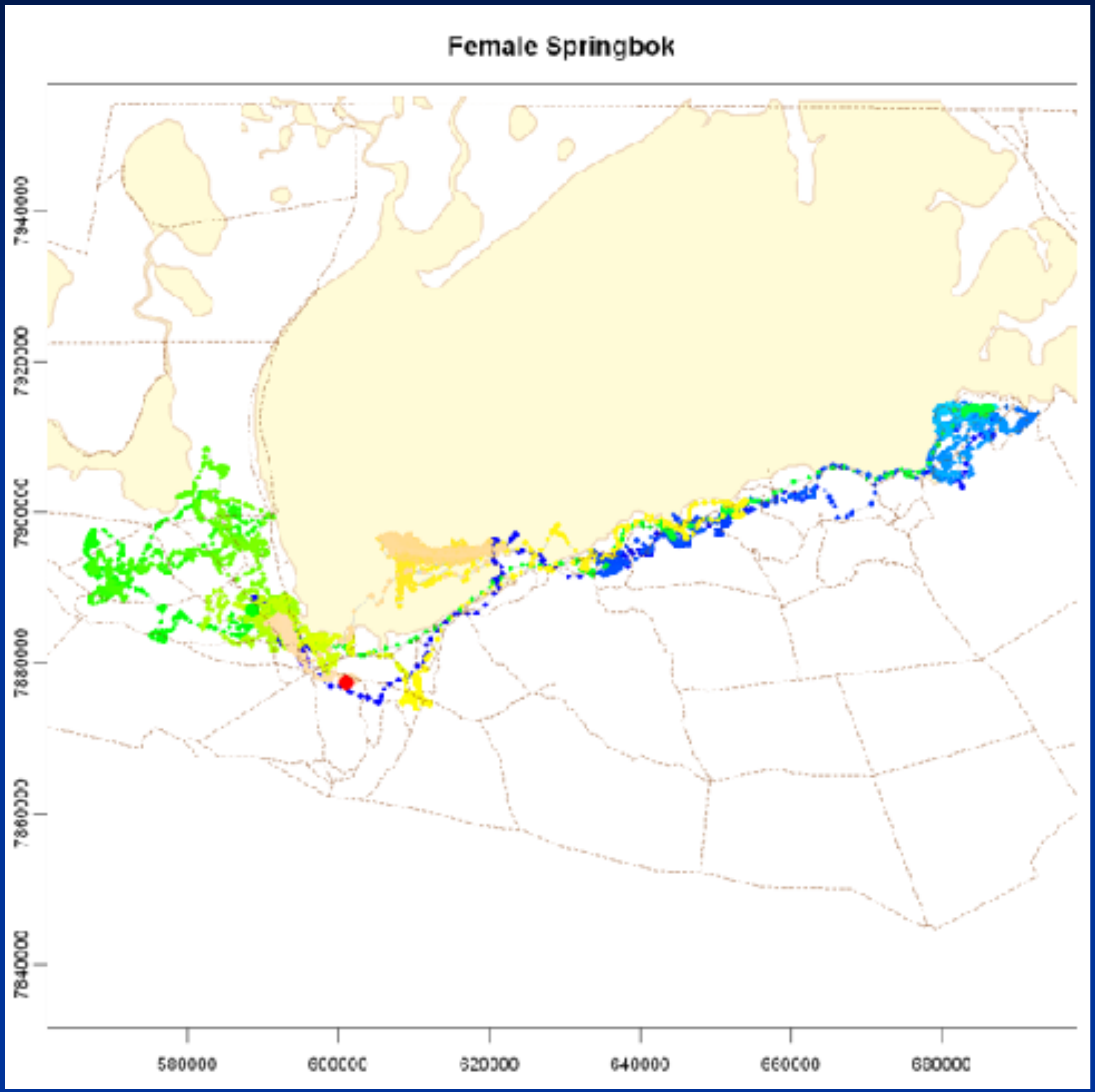
The scatterplots of the hull parent points date vs. hour of day provide clues about what these regions in time-use space represent behaviorally. The red and pink regions of time-use space appear to be 'core night time area' from Oct thru Jan. The blue hulls are visited frequently during mid-day but never for very long – a travel route to water? Green hulls were neither revisited nor used for very long - perhaps searching for greener pastures or run off by another male?

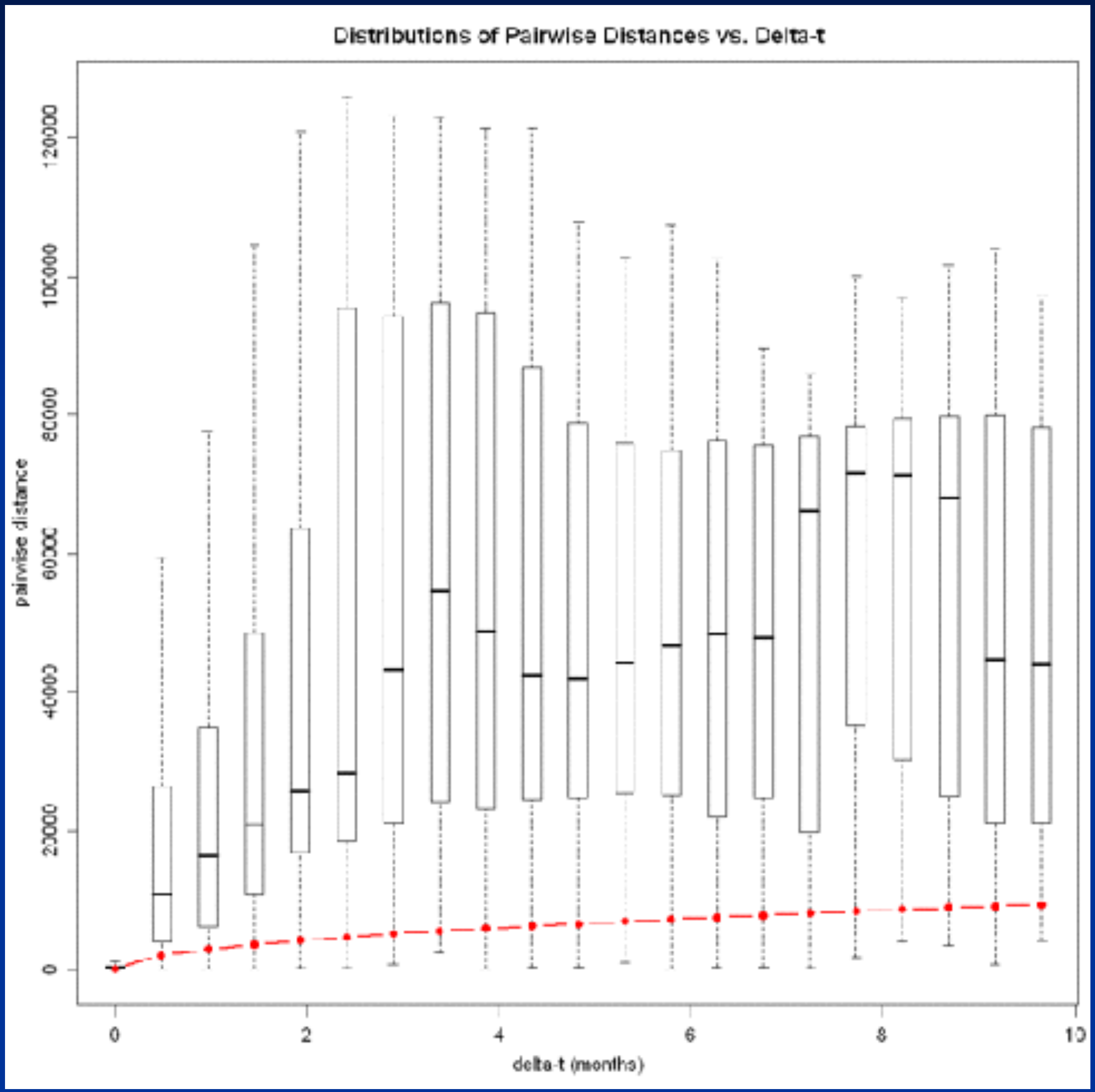


female


ag206.n10704.2009-09-02.2010-04-14
yutm
7840000
7860000
7880000
7900000
7920000
7940000
580000
600000
620000
640000
660000
680000
xutm

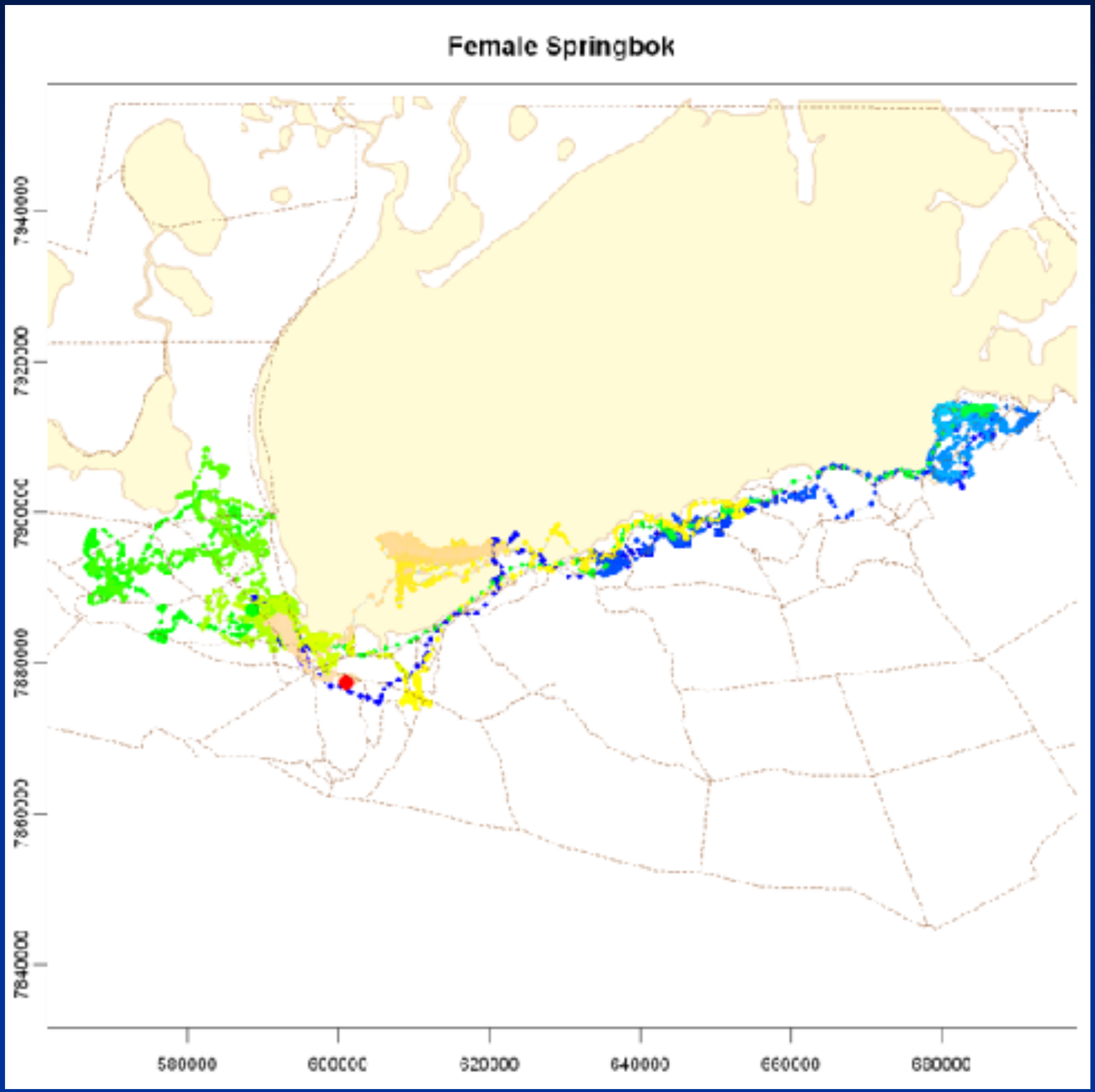
Female Springbok:
Null model fit

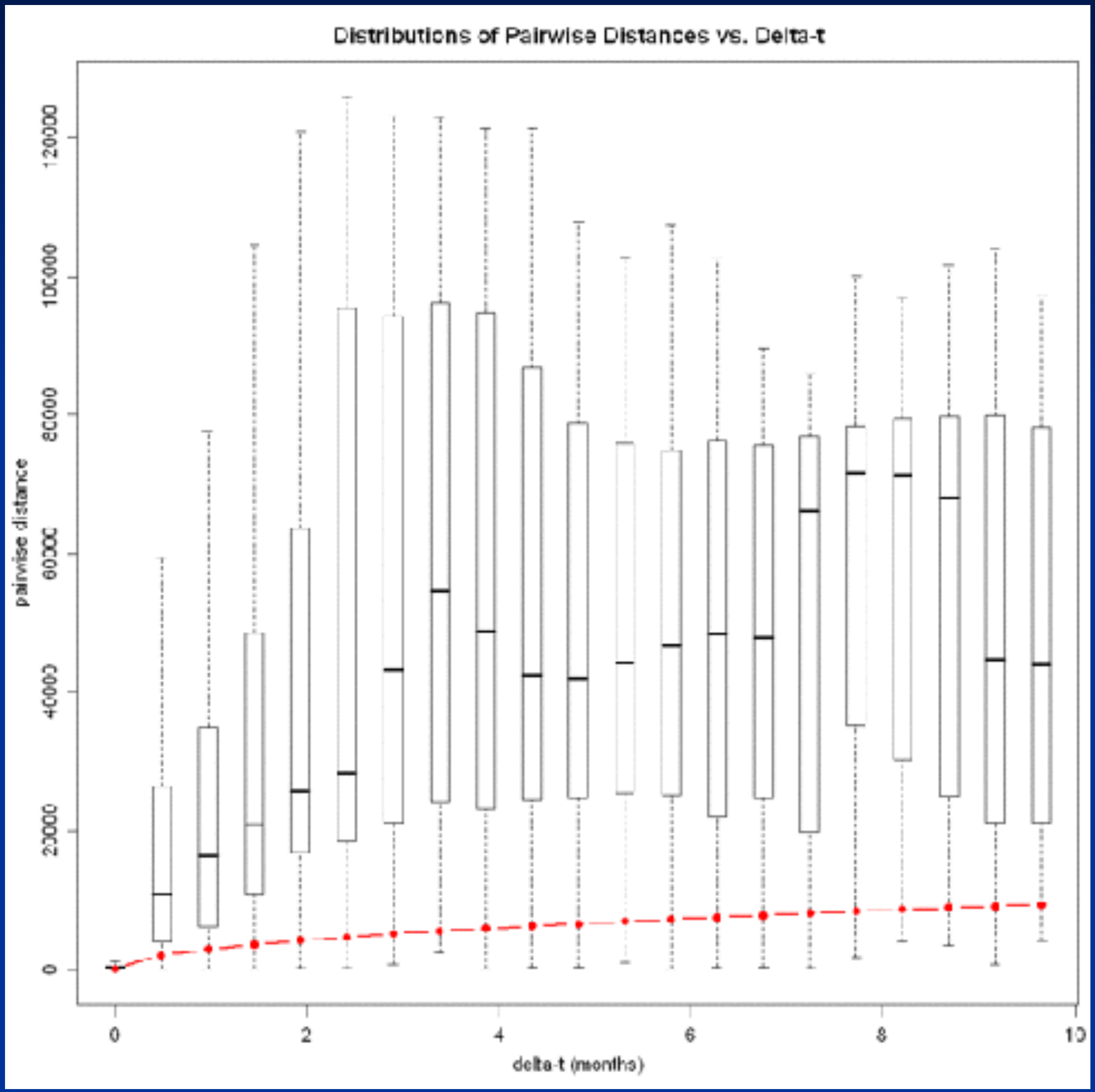




On left, maps of the female springbok in Etosha National Park, Namibia. The colors of the points reflect temporal continuity; tan lines are roads, and yellow polygons are salt pans. On the right, box plots show the distribution of the net displacement of all pairs of points sampled Δt apart (x-axis), with the predicted Gaussian diffusion distance from the random walk null model specified in Equation 2 overlaid in red.

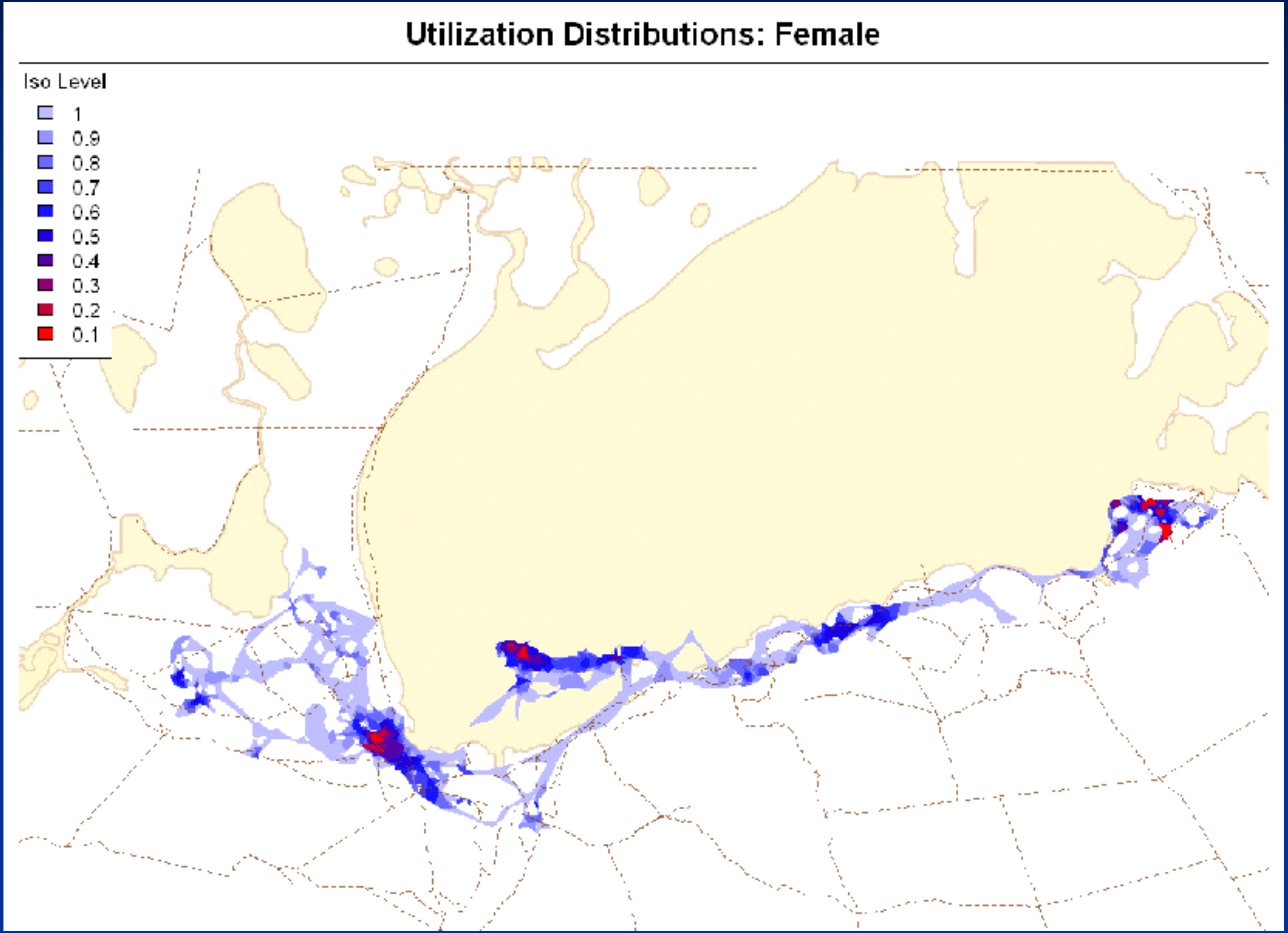
Female Springbok:
Null model fit



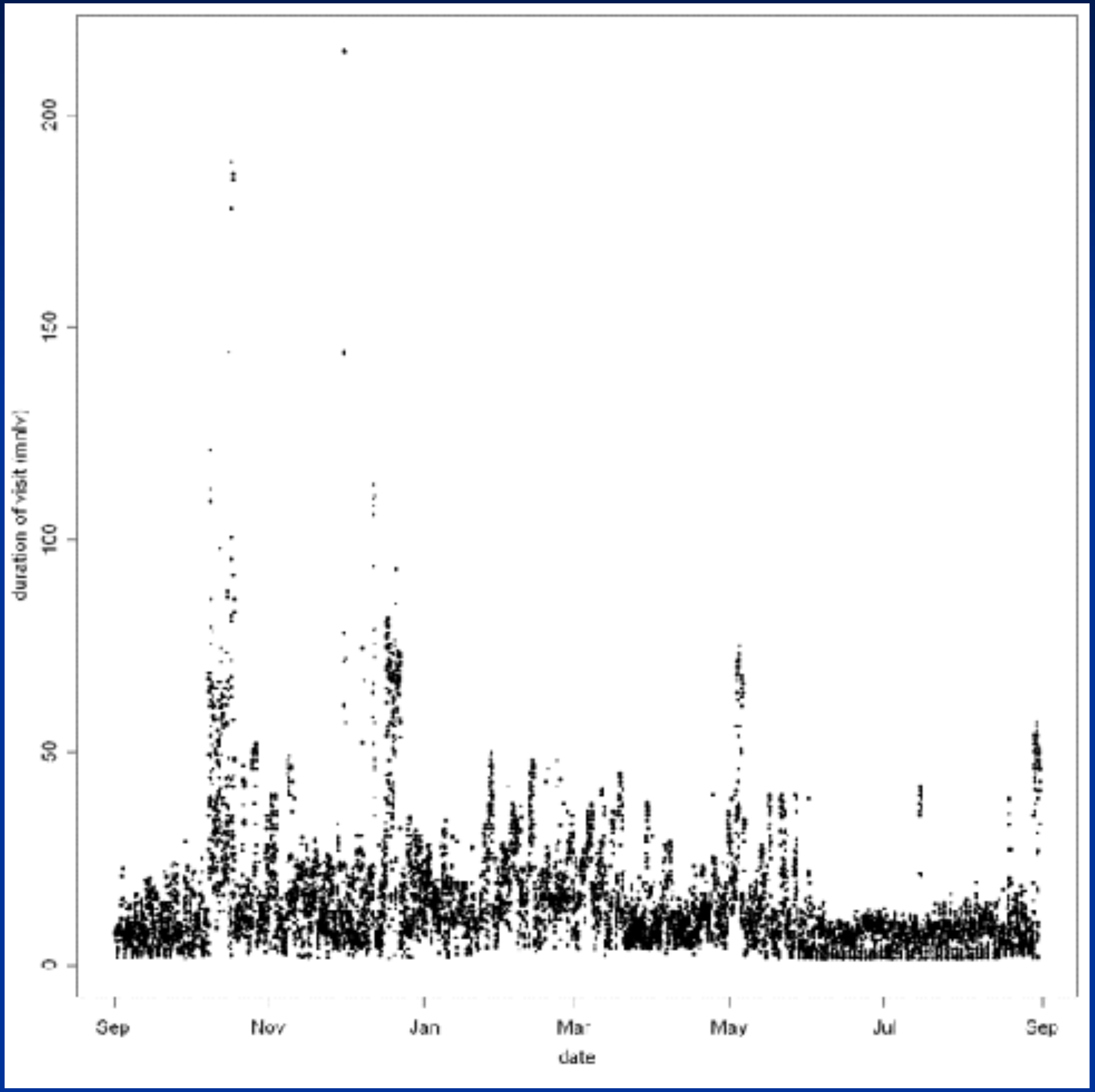


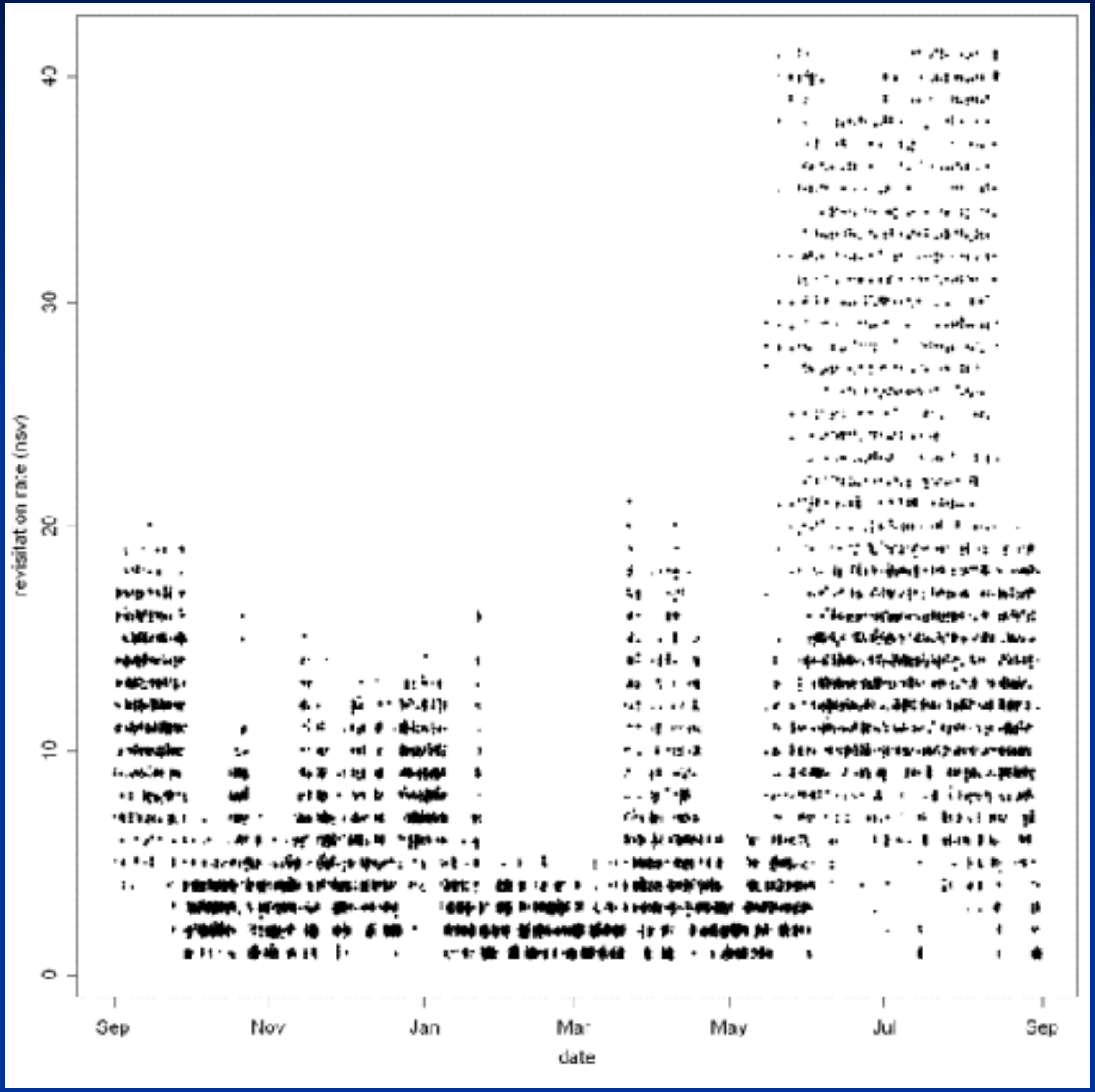
On left, maps of the female springbok in Etosha National Park, Namibia. The colors of the points reflect temporal continuity; tan lines are roads, and yellow polygons are salt pans. On the right, box plots show the distribution of the net displacement of all pairs of points sampled Δt apart (x-axis), with the predicted Gaussian diffusion distance from the random walk null model specified in Equation 2 overlaid in red.

Female Springbok:
Density Isopleths



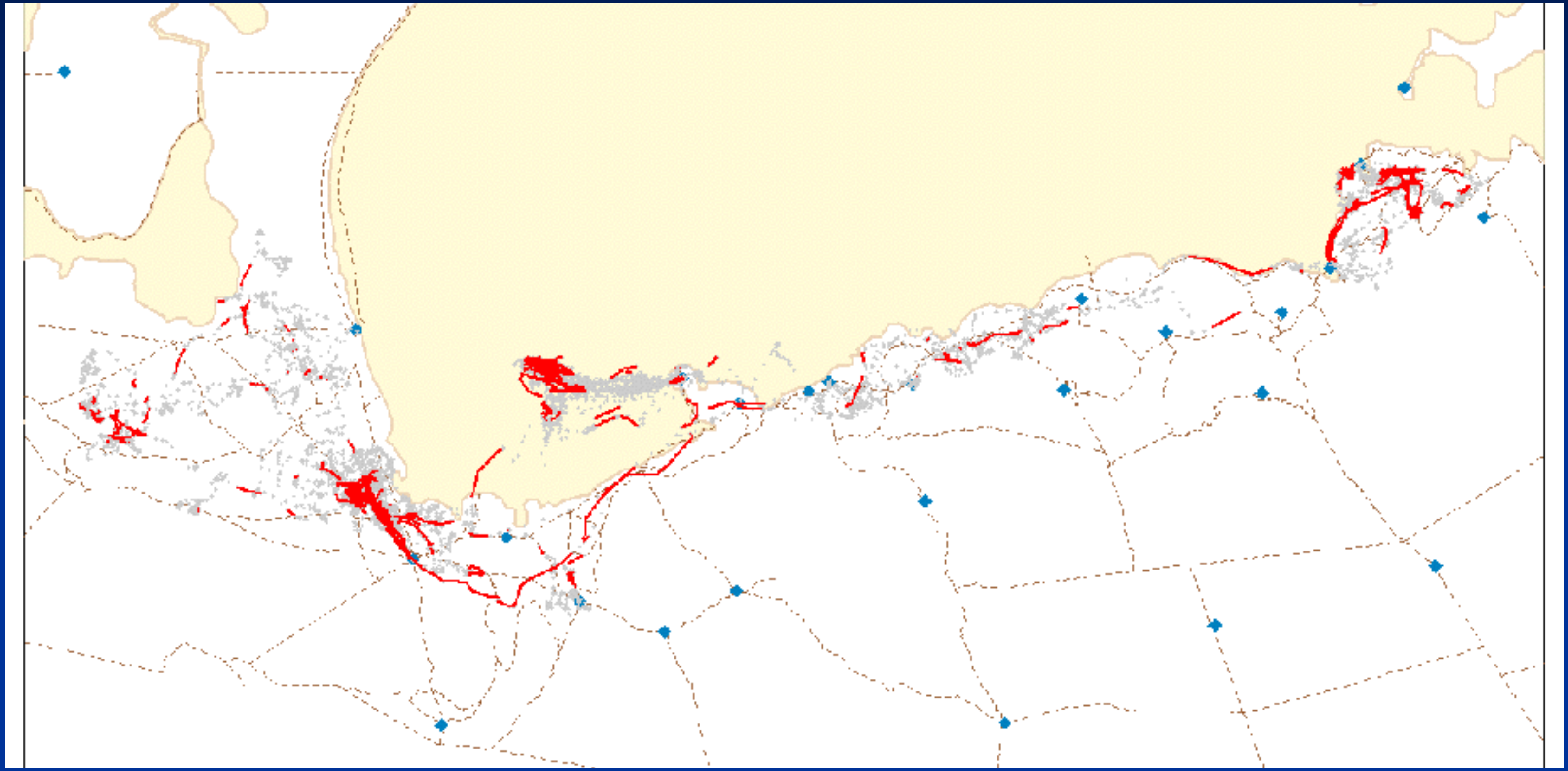
Female Springbok:
Hull revisitation rate and duration over time





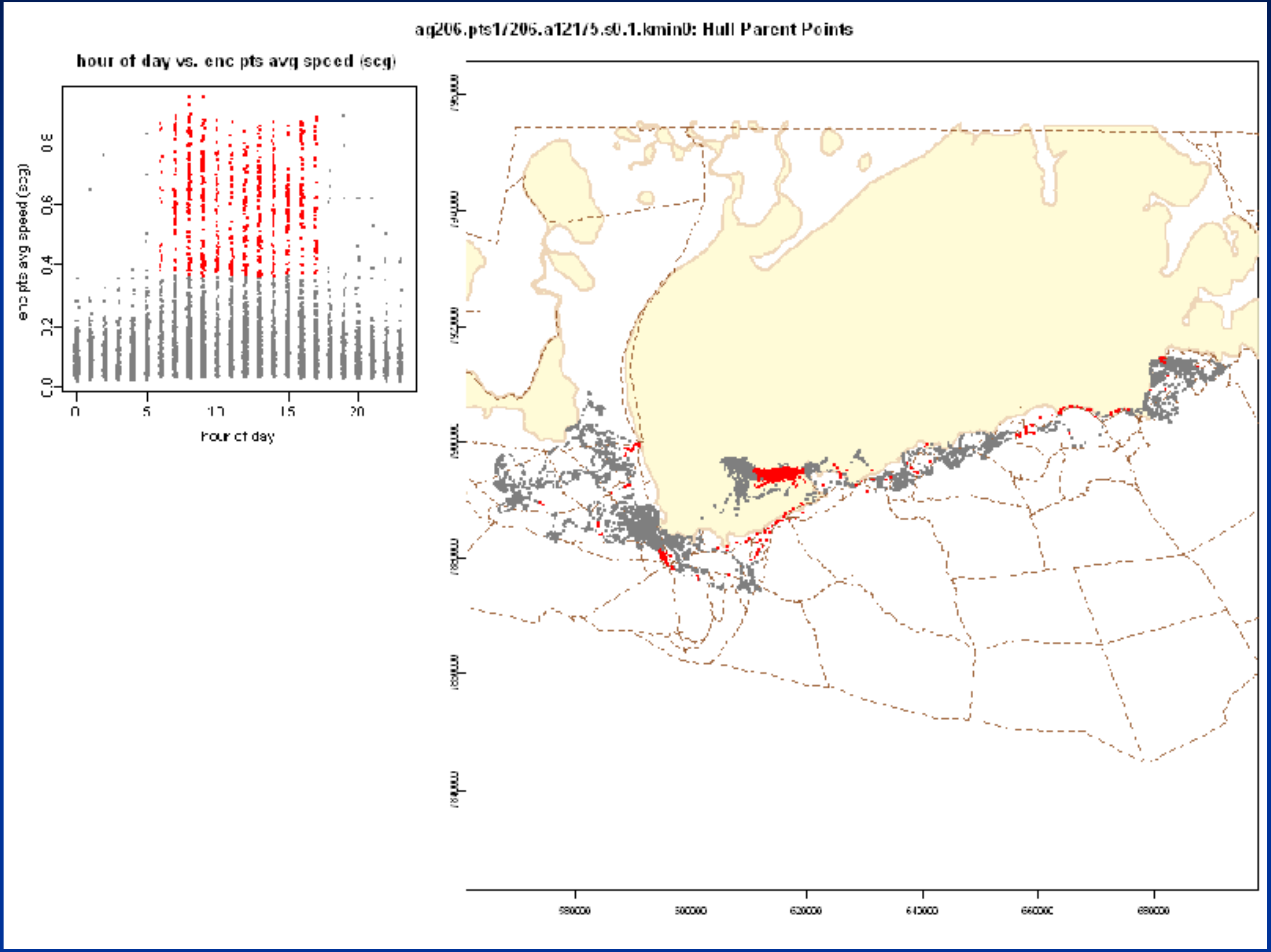
Plots of hull revisitation rate and visit duration over time for the female springbok. Separate visits defined by an inter-visit gap period of 1 day. Values have been 'jiggled' by 0.1 to better see point density.

Female Springbok:
Directional Routes



Map of directional routes for the female springbok derived from connecting the parent points of temporally contiguous hulls with a perimeter area ratio value in the top 35%. Perimeter area ratios have been smoothed with a temporal averaging function and scanning window of one time step. Blue dots are known water points.

Female Springbok:
Hour of day vs. Speed of enclosed pts



Conclusions

- Ancillary variables such as time can be incorporated into home range methods in such as a way as to extend the range in behavioral questions
- T-LoCoH shows promise in developing spatial distributions for movement phase and time use
- One-click solutions remain elusive
- Future Directions
  - Model diffusion distance from data
  - More sophisticated pattern detection
  - Hulls provide a platform for environmental variables, interactions between individuals
  - More assistants for parameter selection
  - Ecological applications