

Advanced Data Visualization

CS 6965

Spring 2018

Prof. Bei Wang Phillips

University of Utah



Lecture 14

Announcement

- Project 2 is due today! What I am looking for: efforts.
- Project 3 is posted today!

Topological structures and TTK

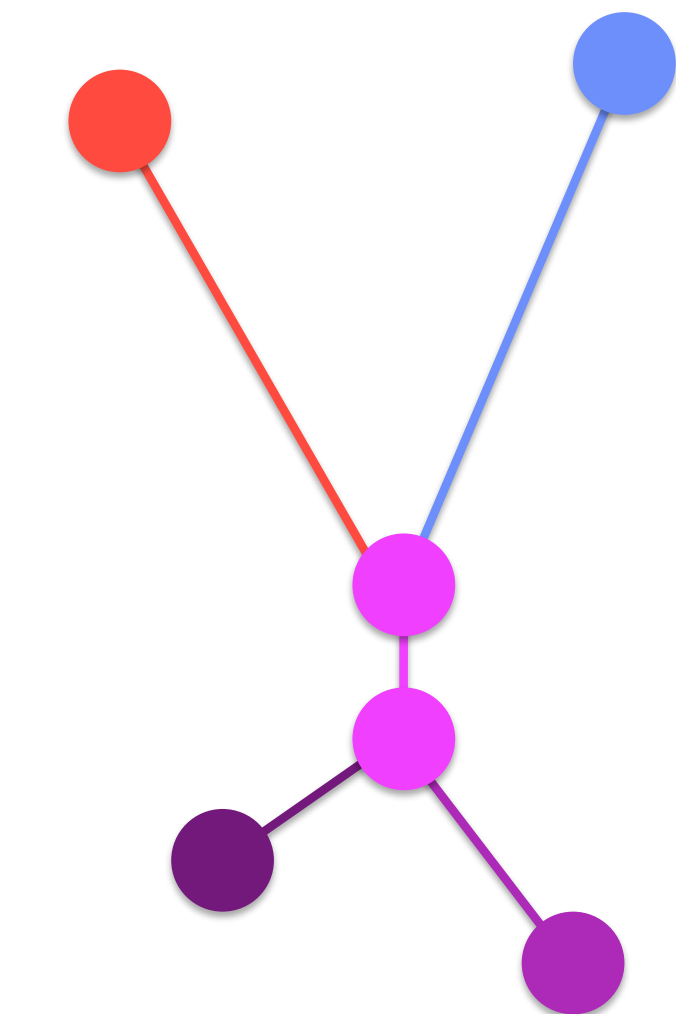
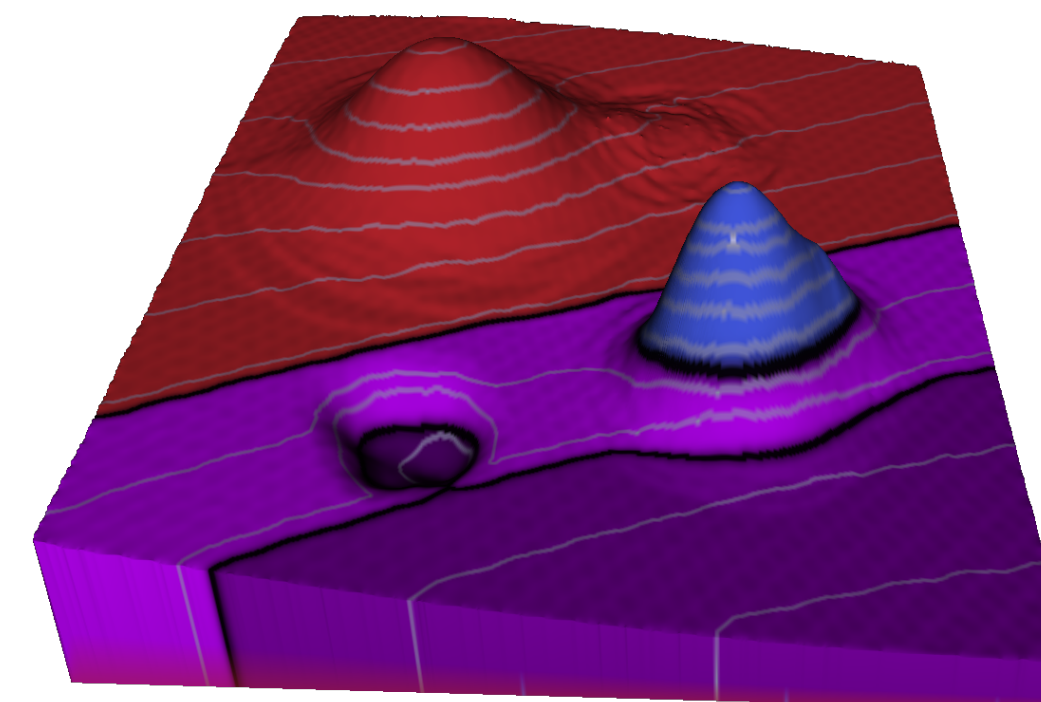
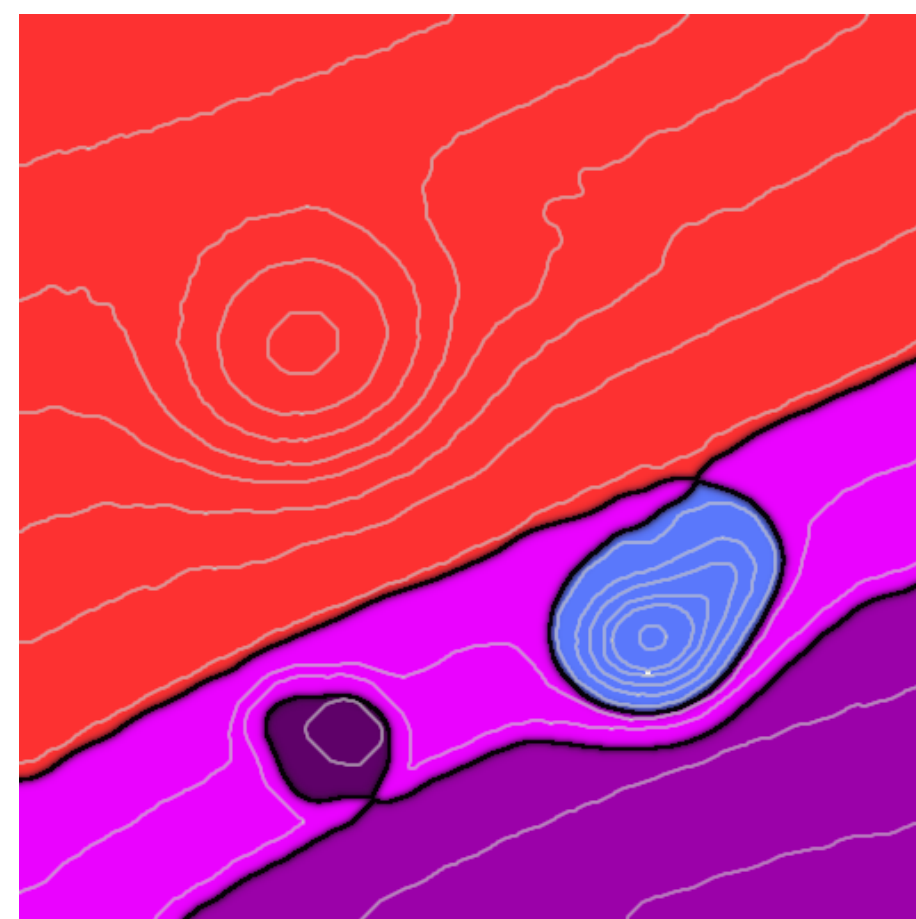
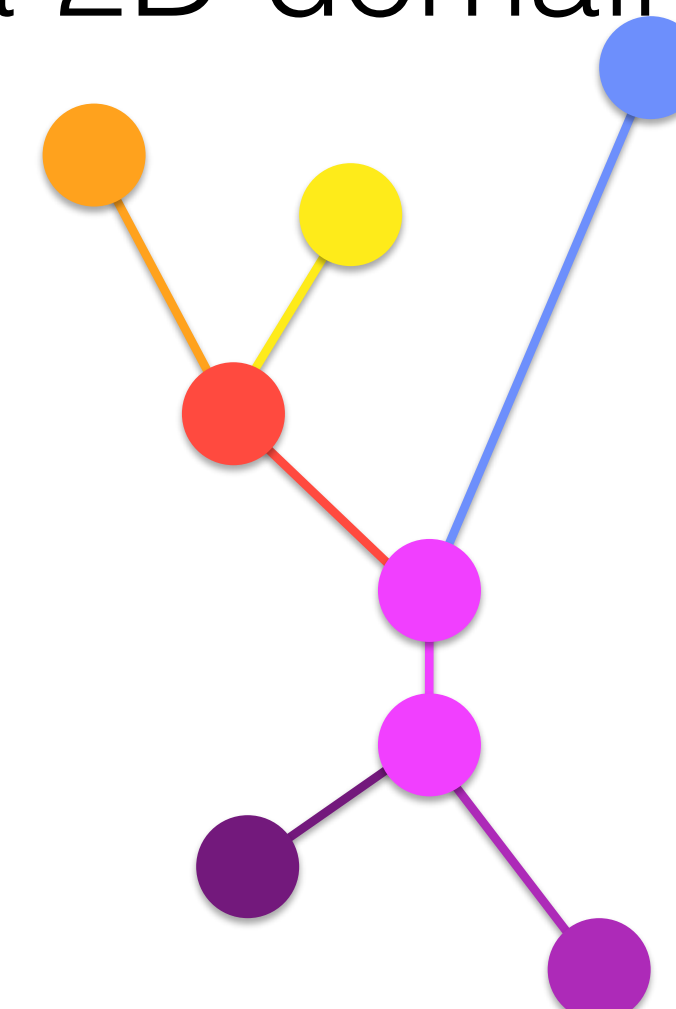
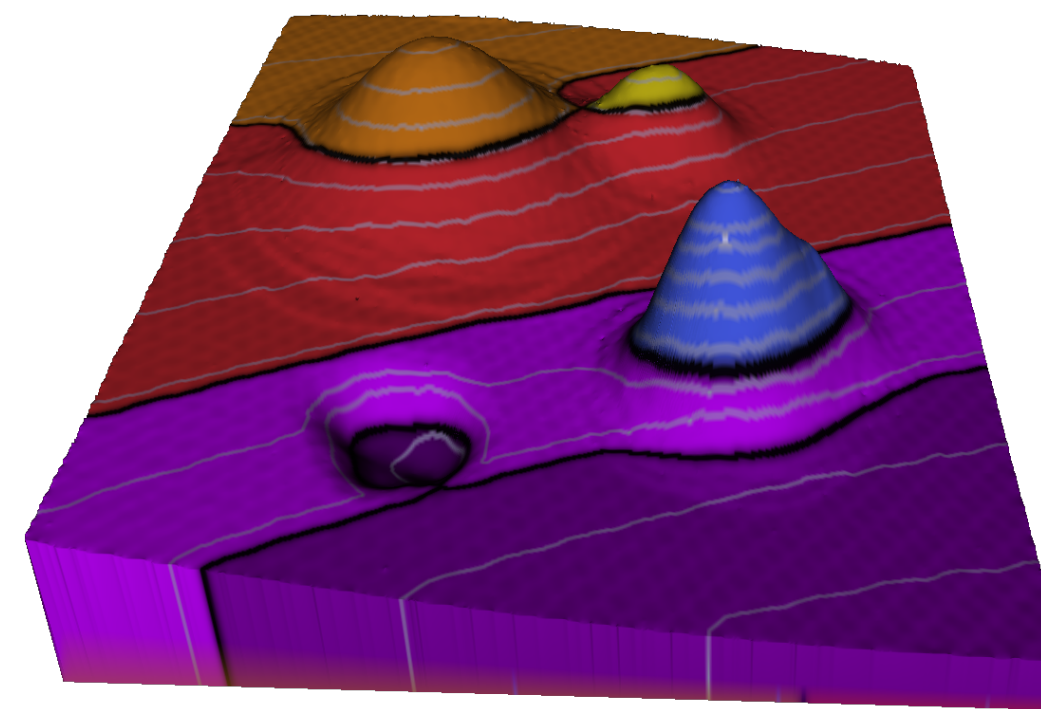
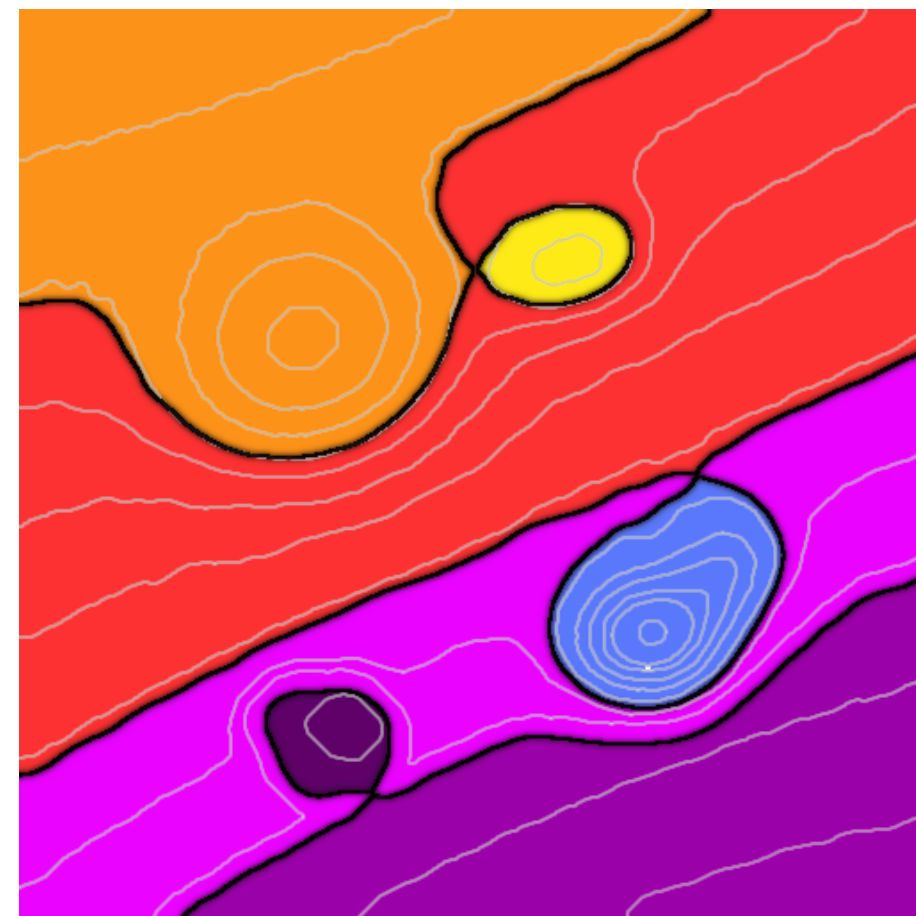


TOPO

Review: Contour Trees and Morse-Smale Complex

Data has shape

Elevation on a terrain: function on a 2D domain

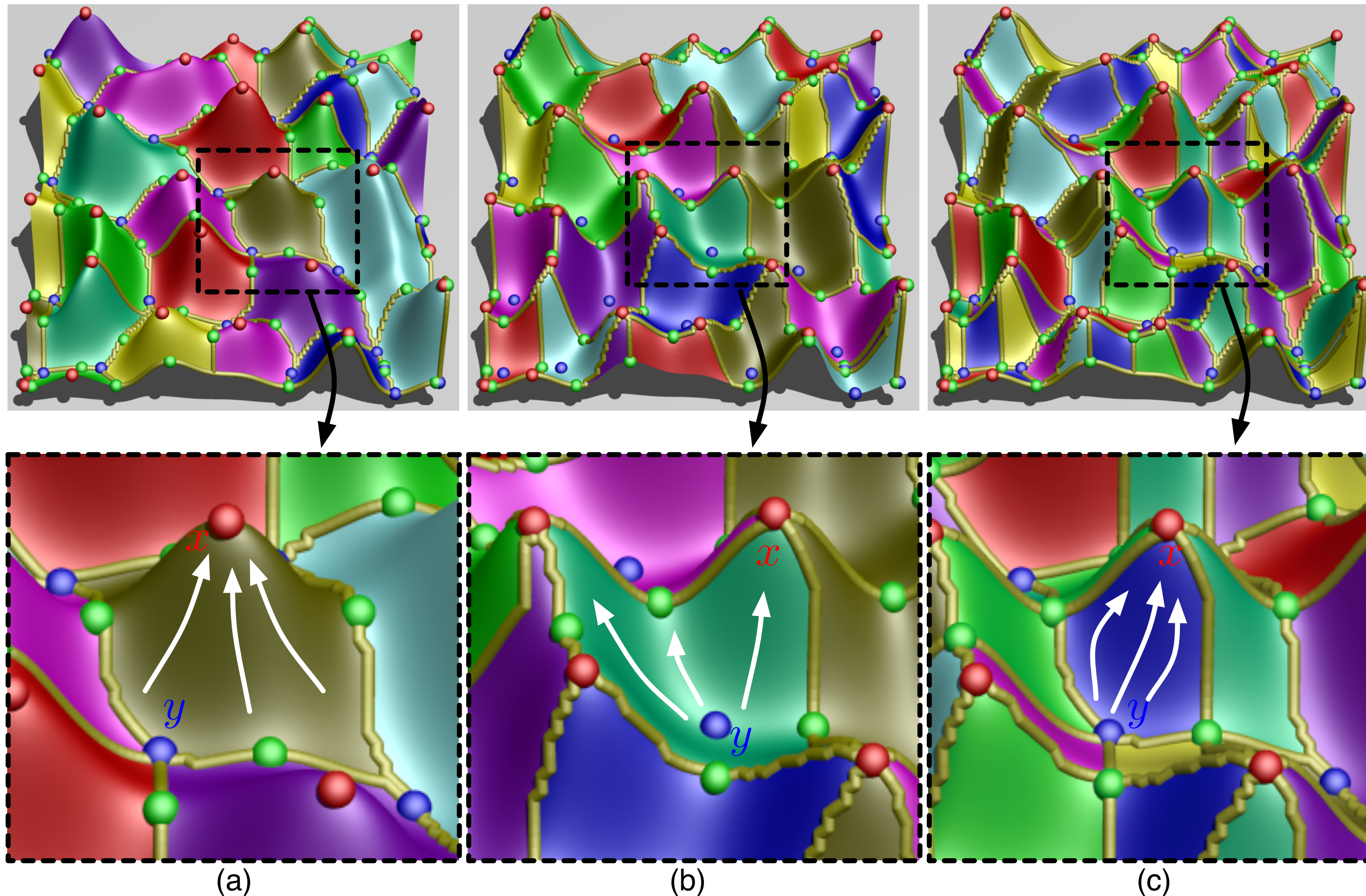


Shape of data?

Contour Tree

Data has shape

Elevation on a terrain: function on a 2D domain



Shape of data?

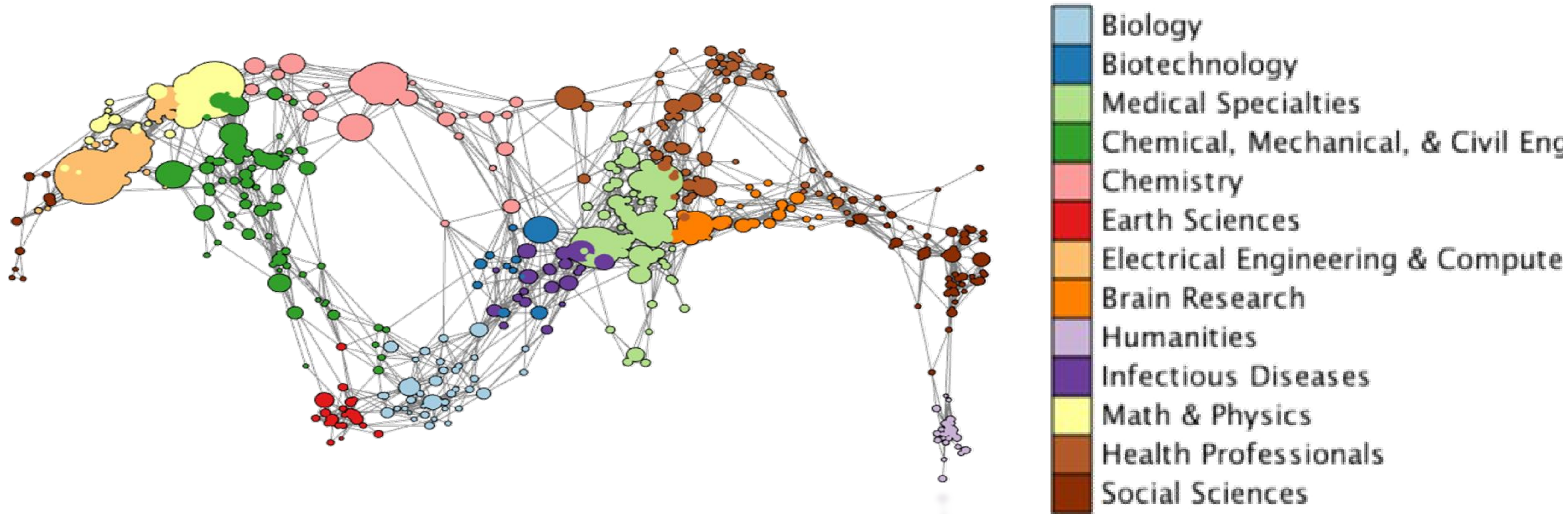
Morse-Smale Complex

A Map of Science Example

MAP OF SCIENCE?



MAP OF SCIENCE

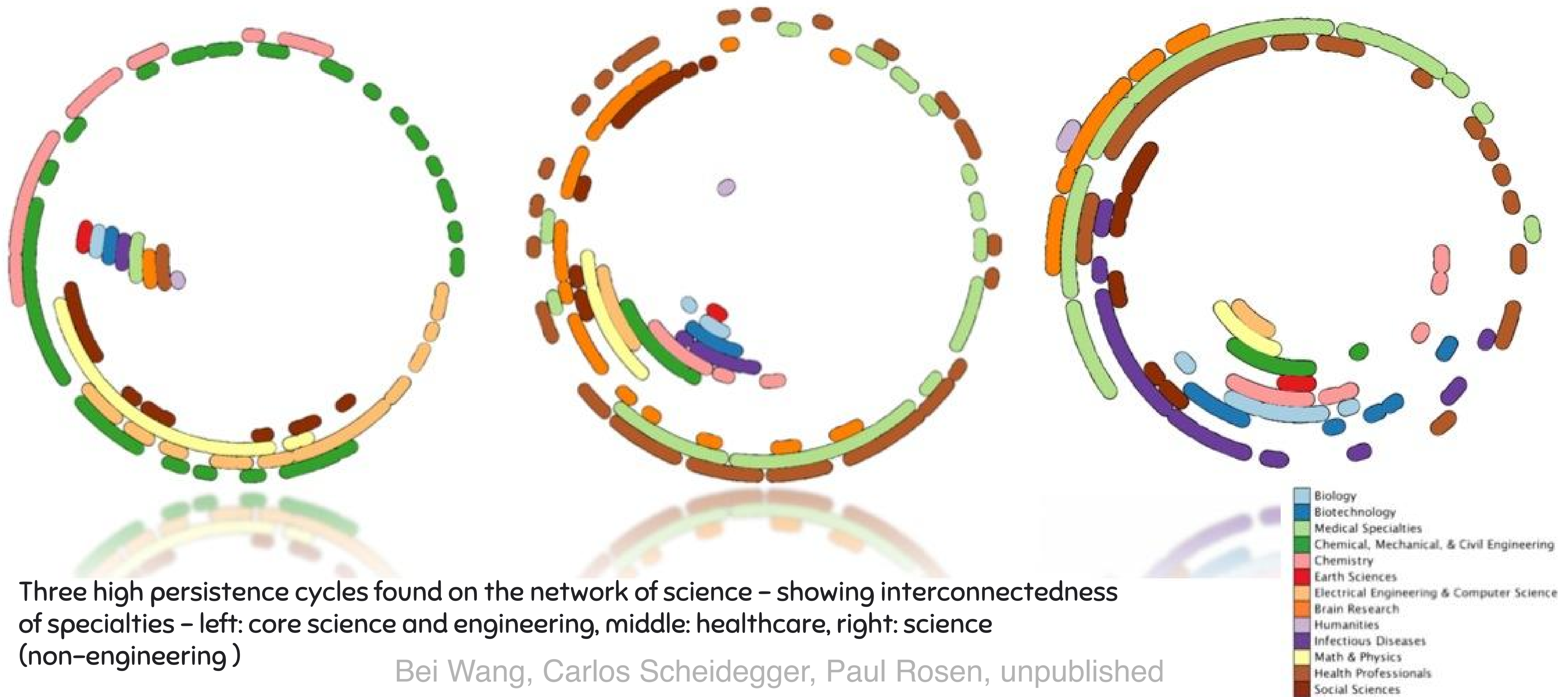


Mercator coordinate visualization of a spherically embedded graph representing the interconnectivity of science from data in [Borner et. al. 2012]



The network was embedded in a low-dim space that the authors concluded by visual inspection, that “the consensus map has a circular form”.

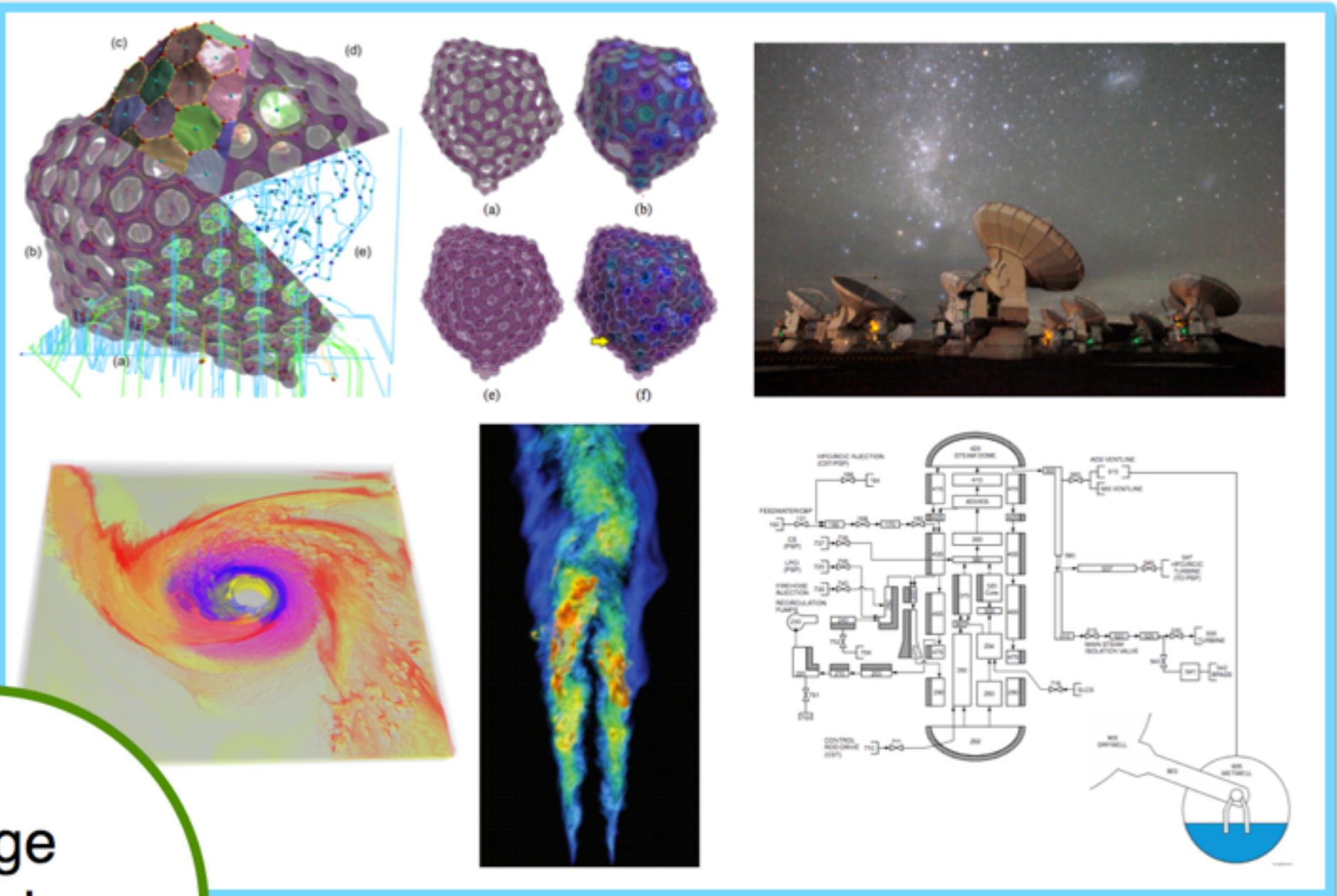
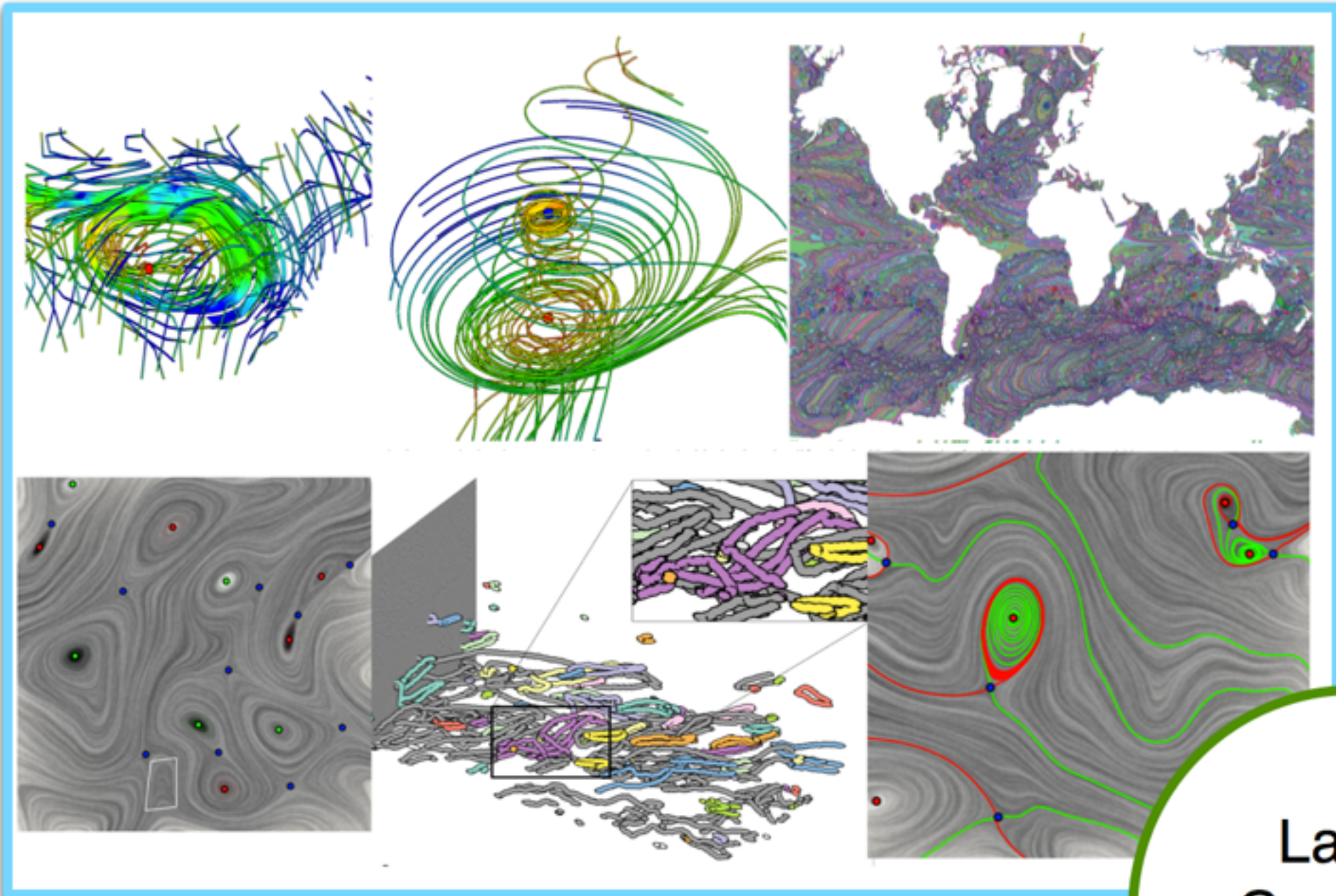
WITH TDA: WHAT IS THE SHAPE OF THE MAP OF SCIENCE?



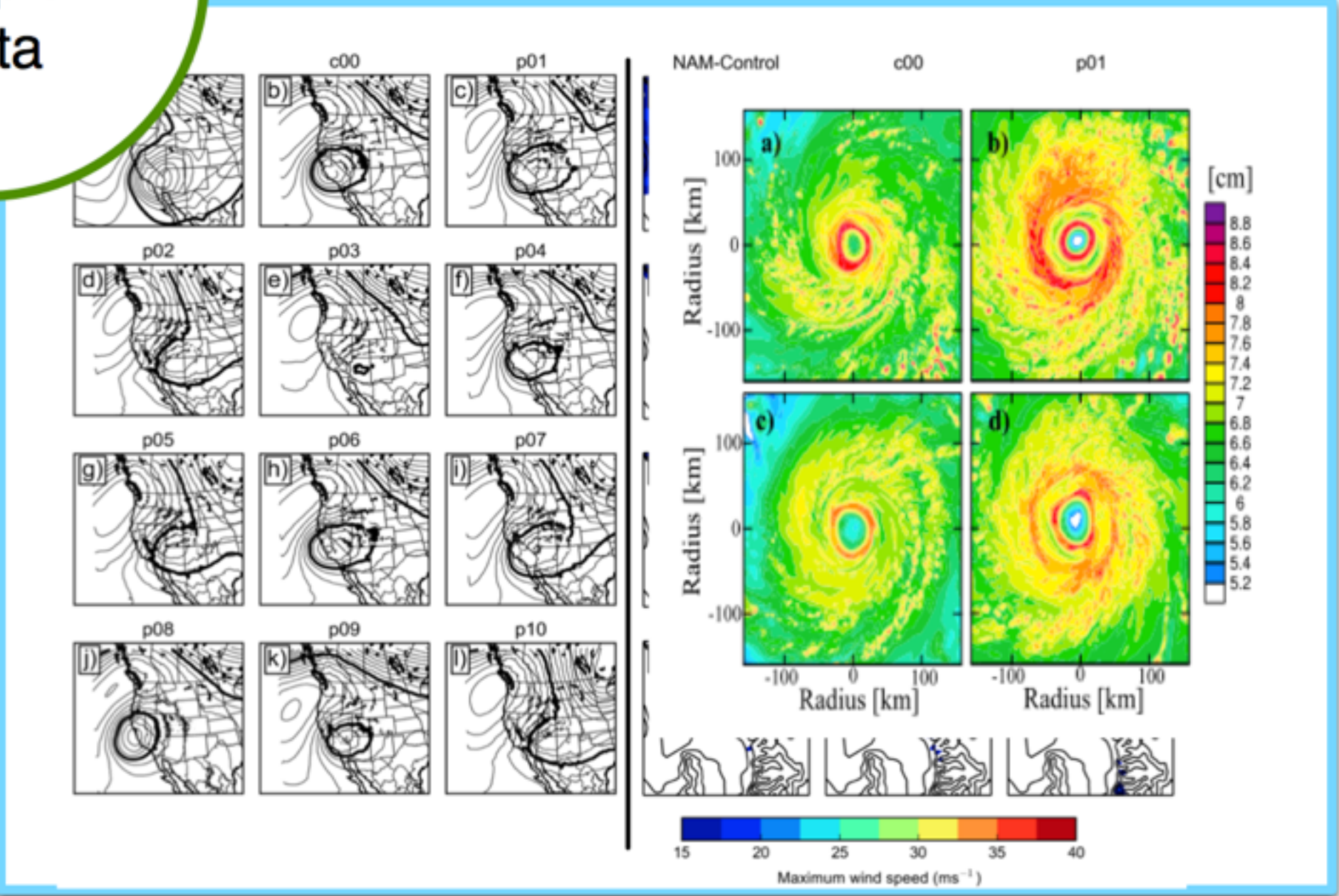
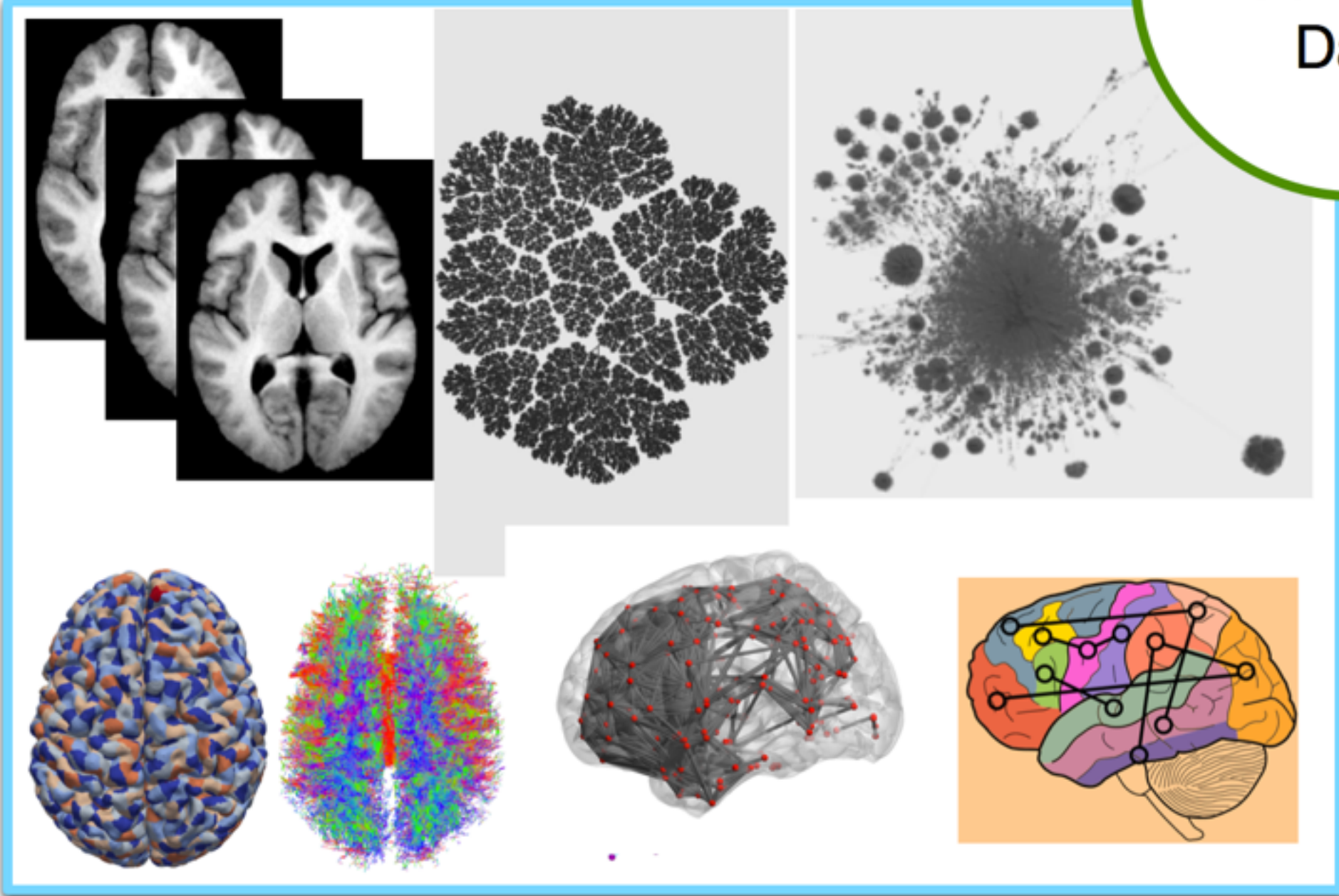
Three high persistence cycles found on the network of science – showing interconnectedness of specialties – left: core science and engineering, middle: healthcare, right: science (non-engineering)

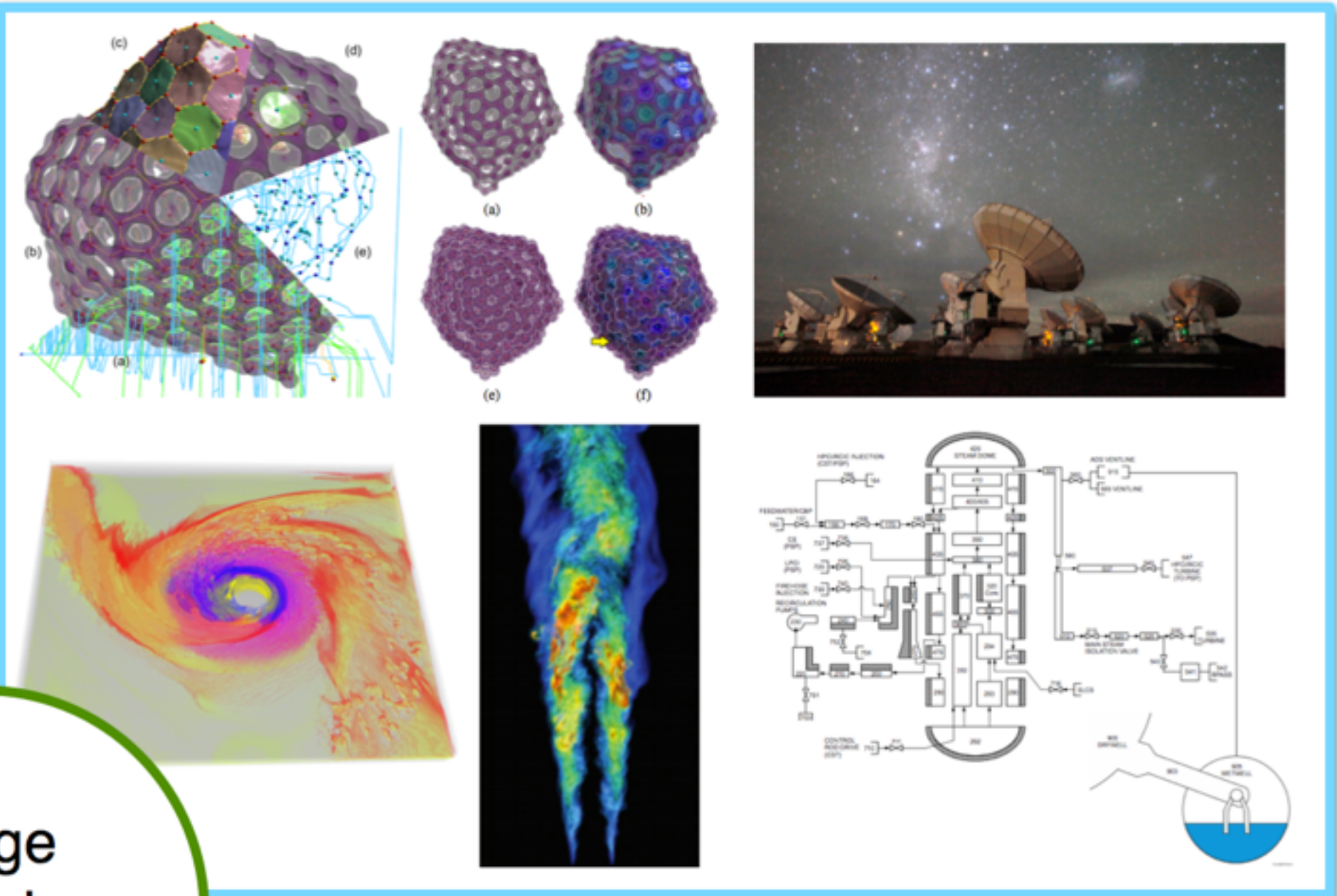
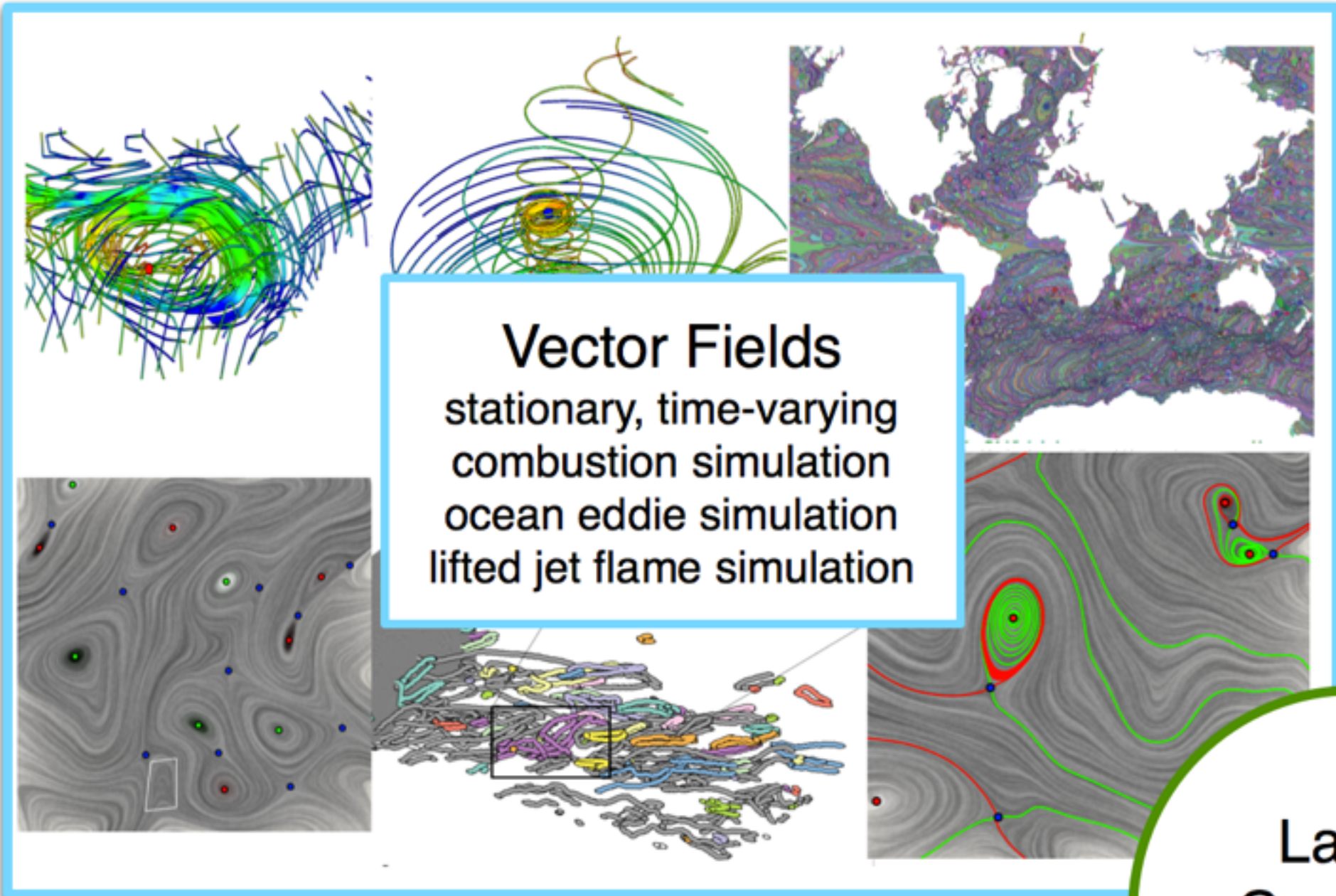
Bei Wang, Carlos Scheidegger, Paul Rosen, unpublished

What is data?

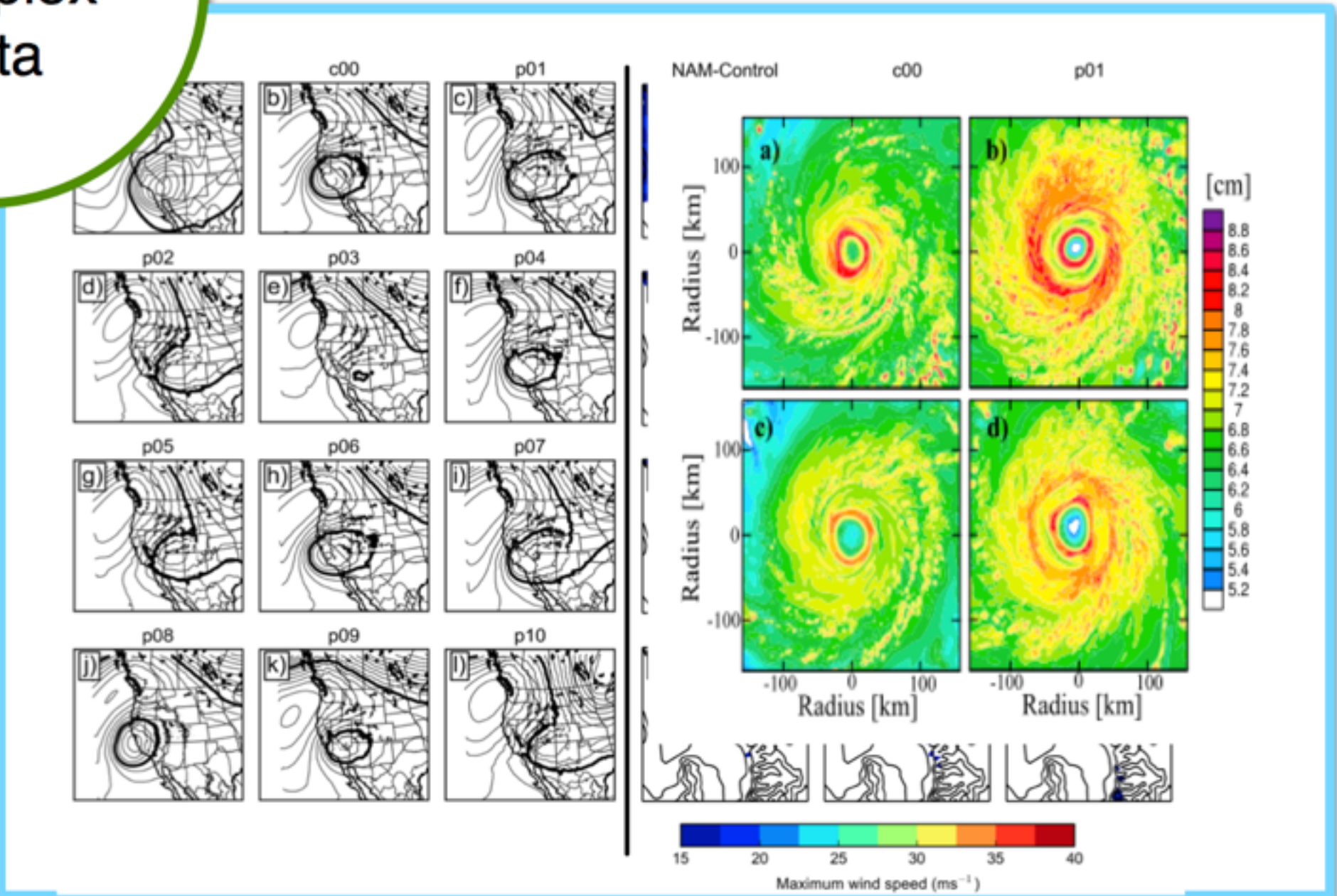
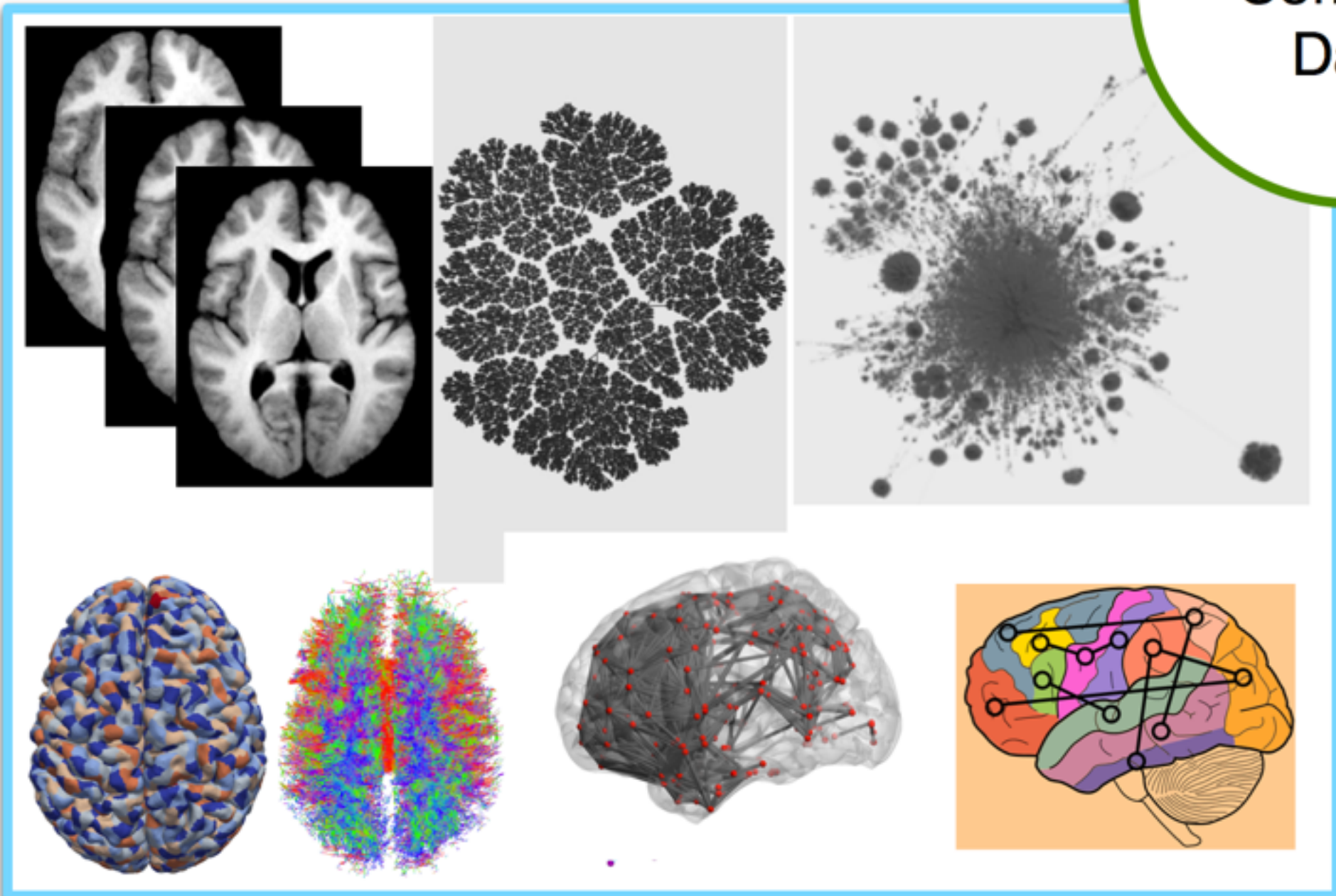


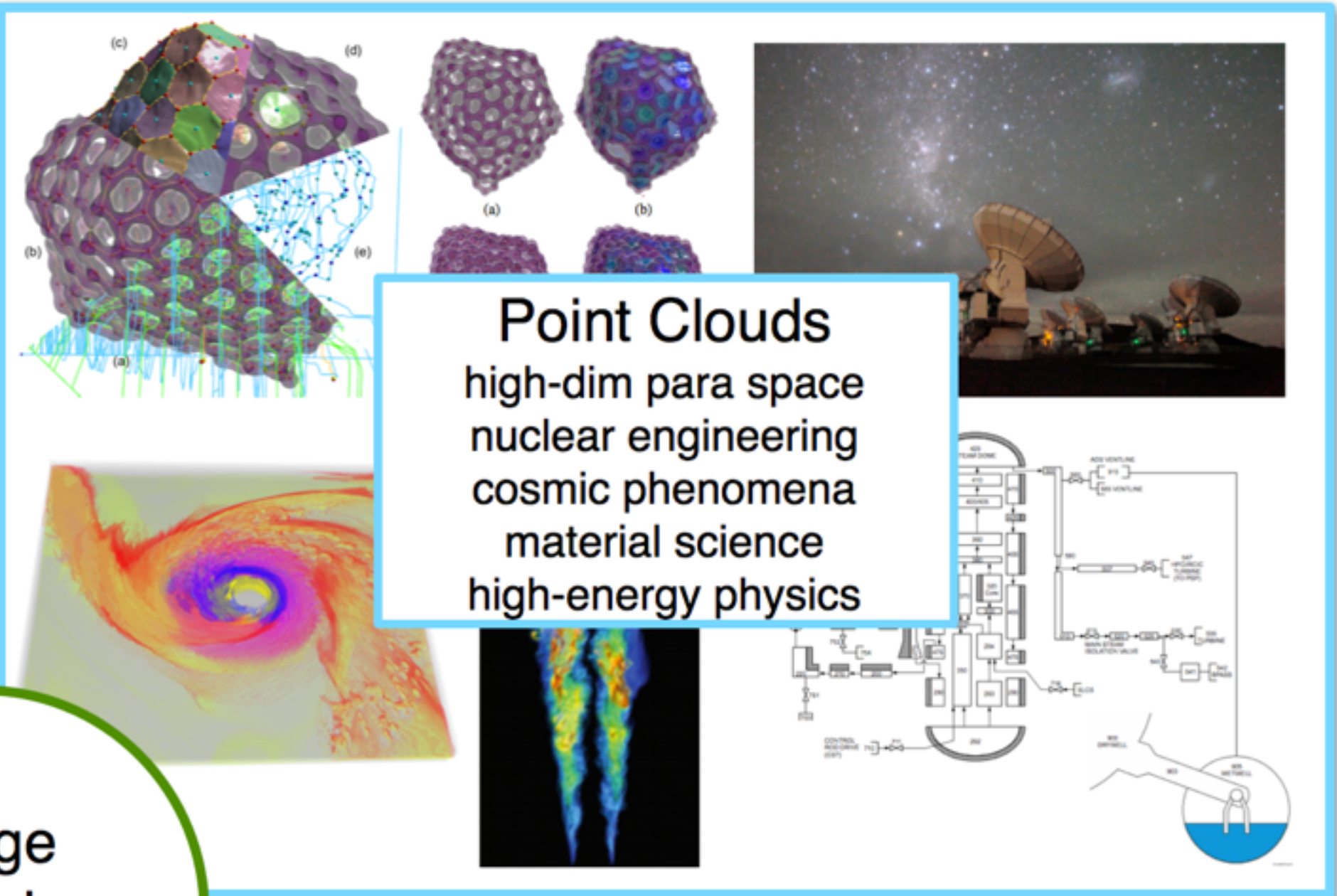
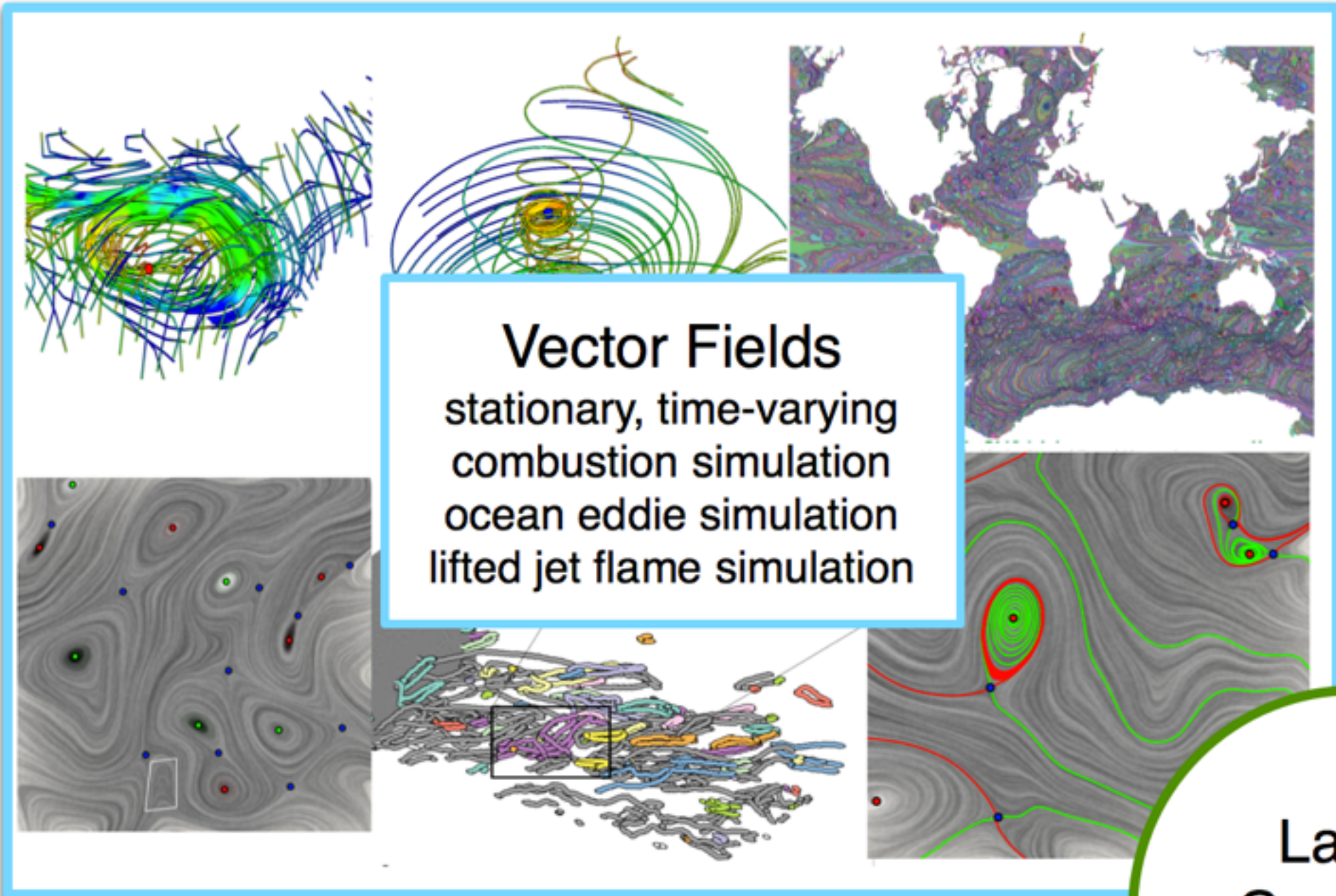
Large Complex Data



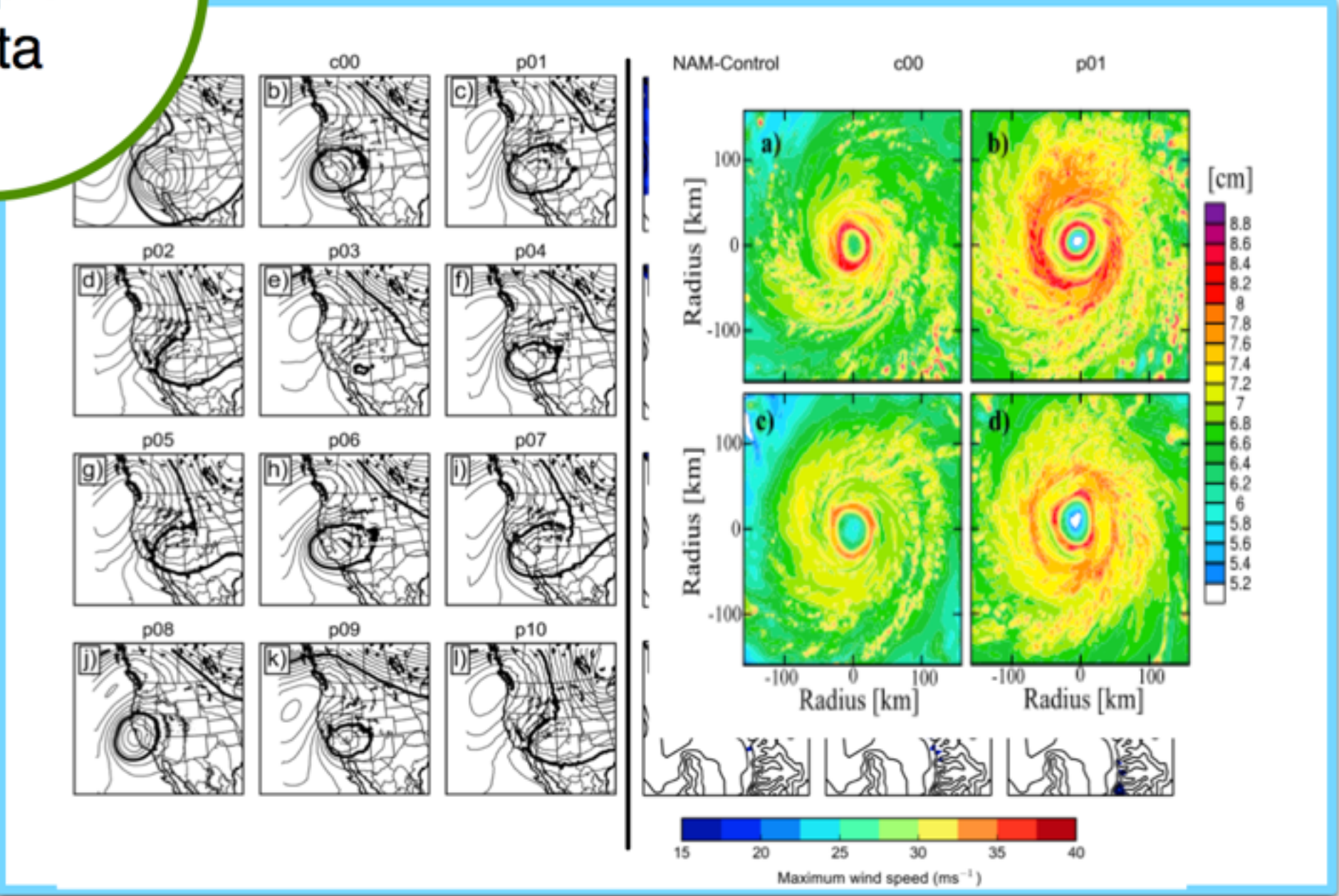
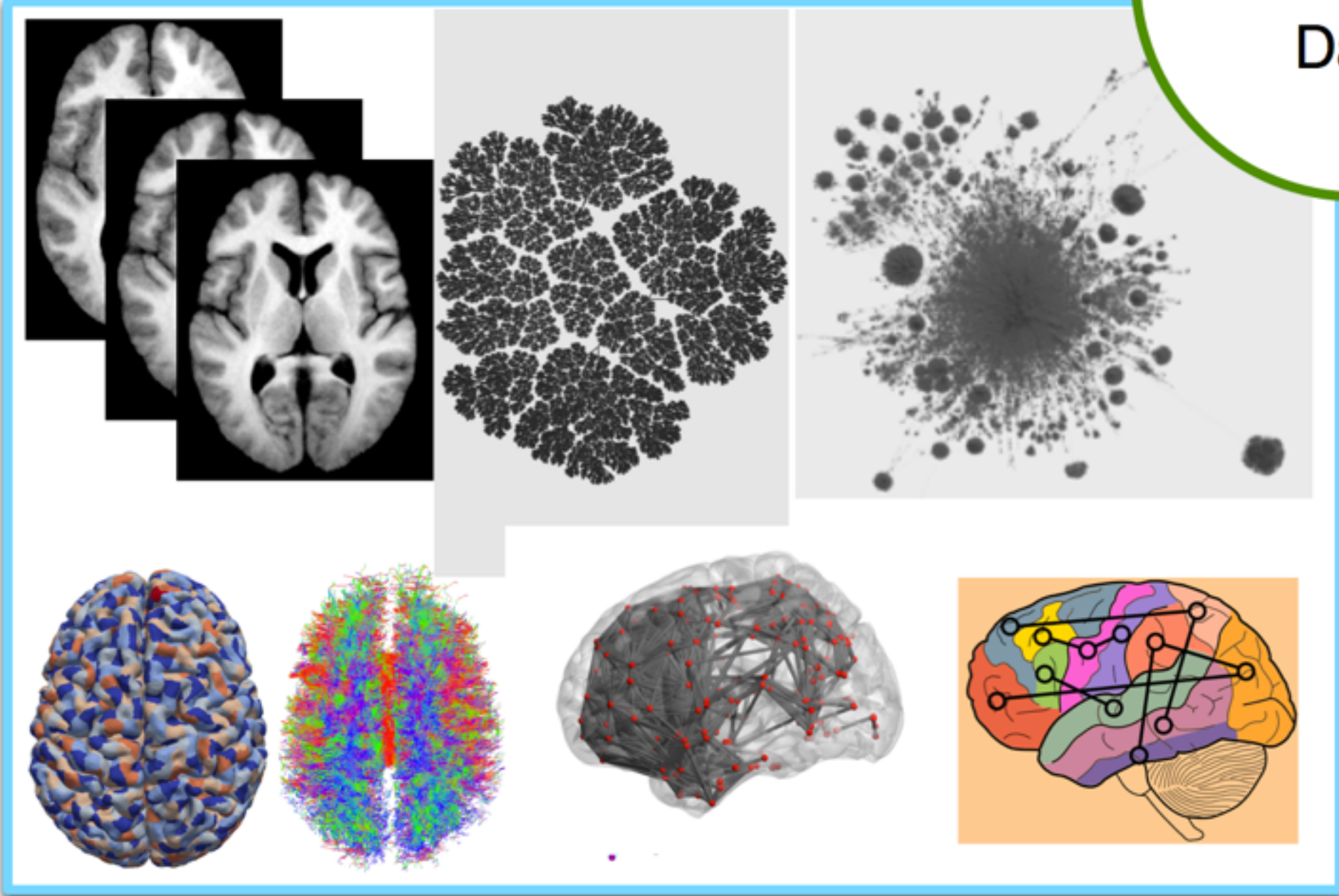


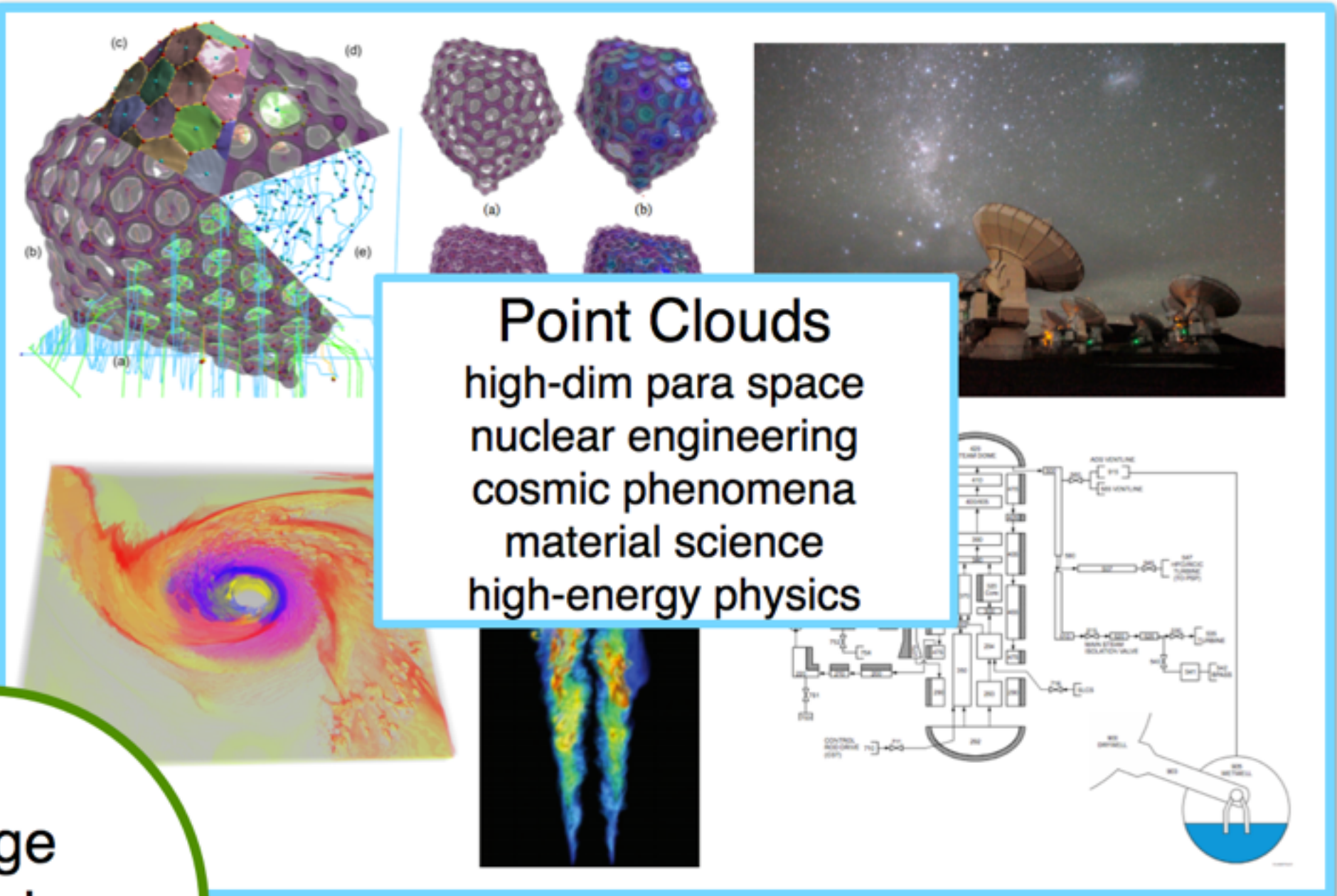
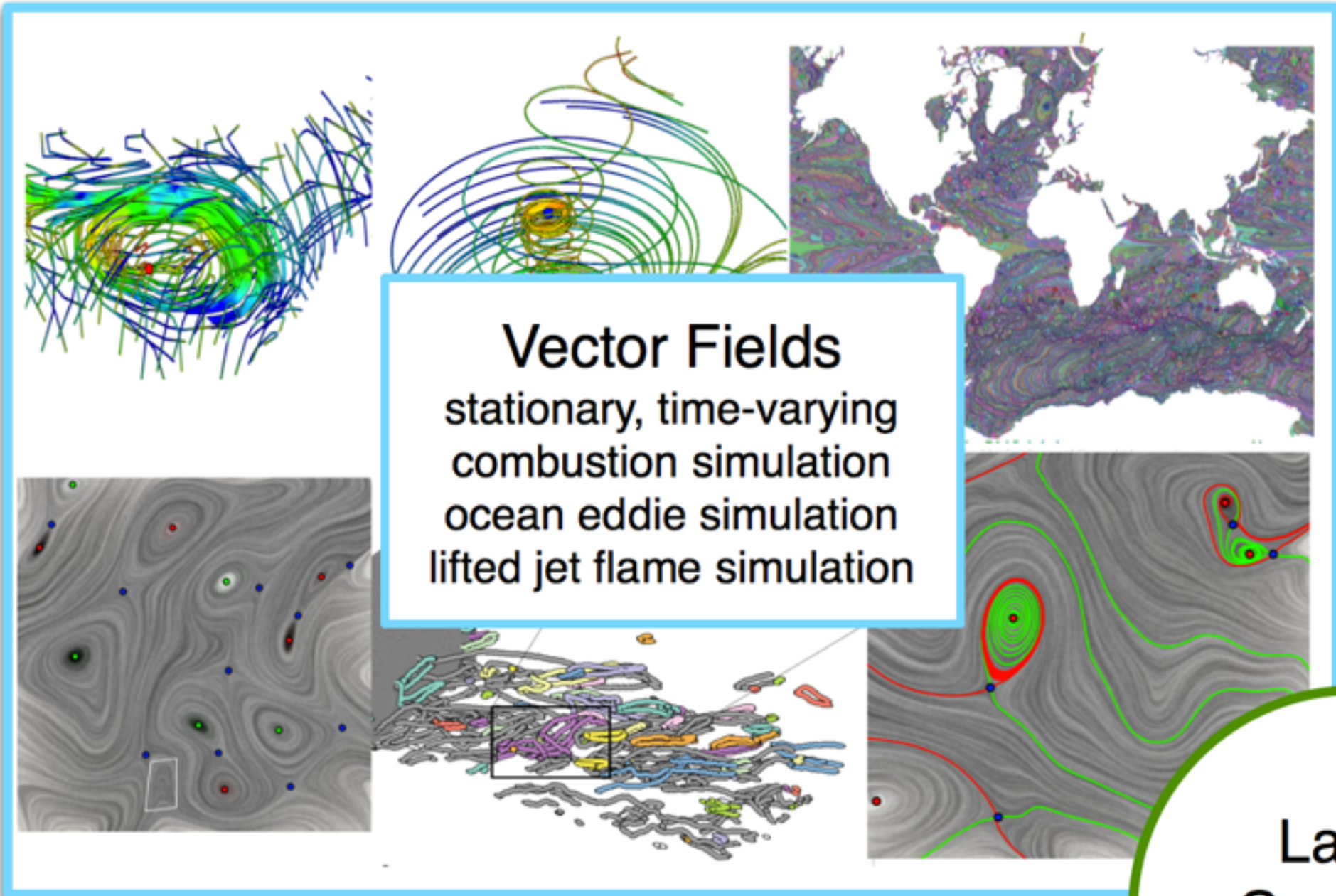
Large
Complex
Data



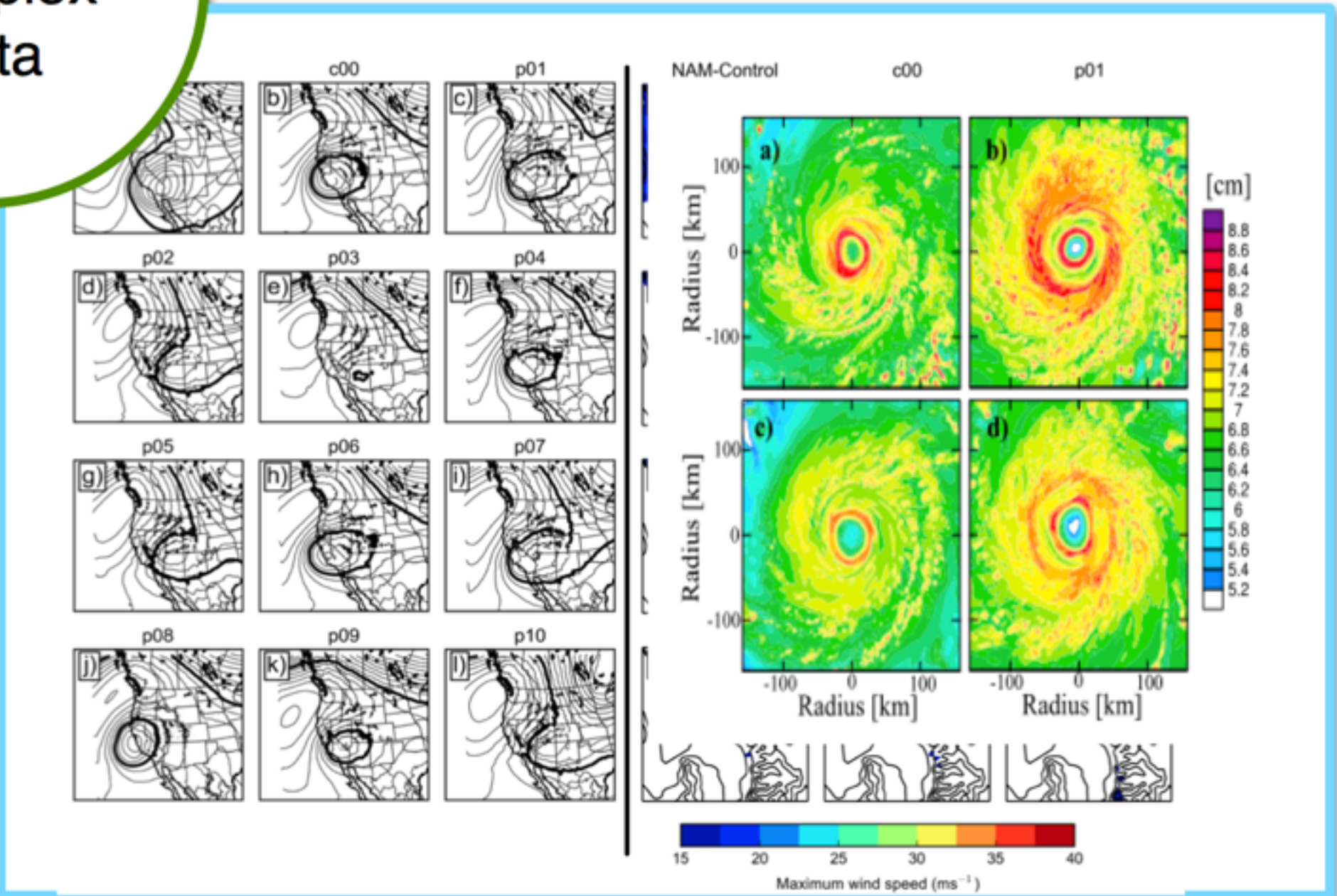
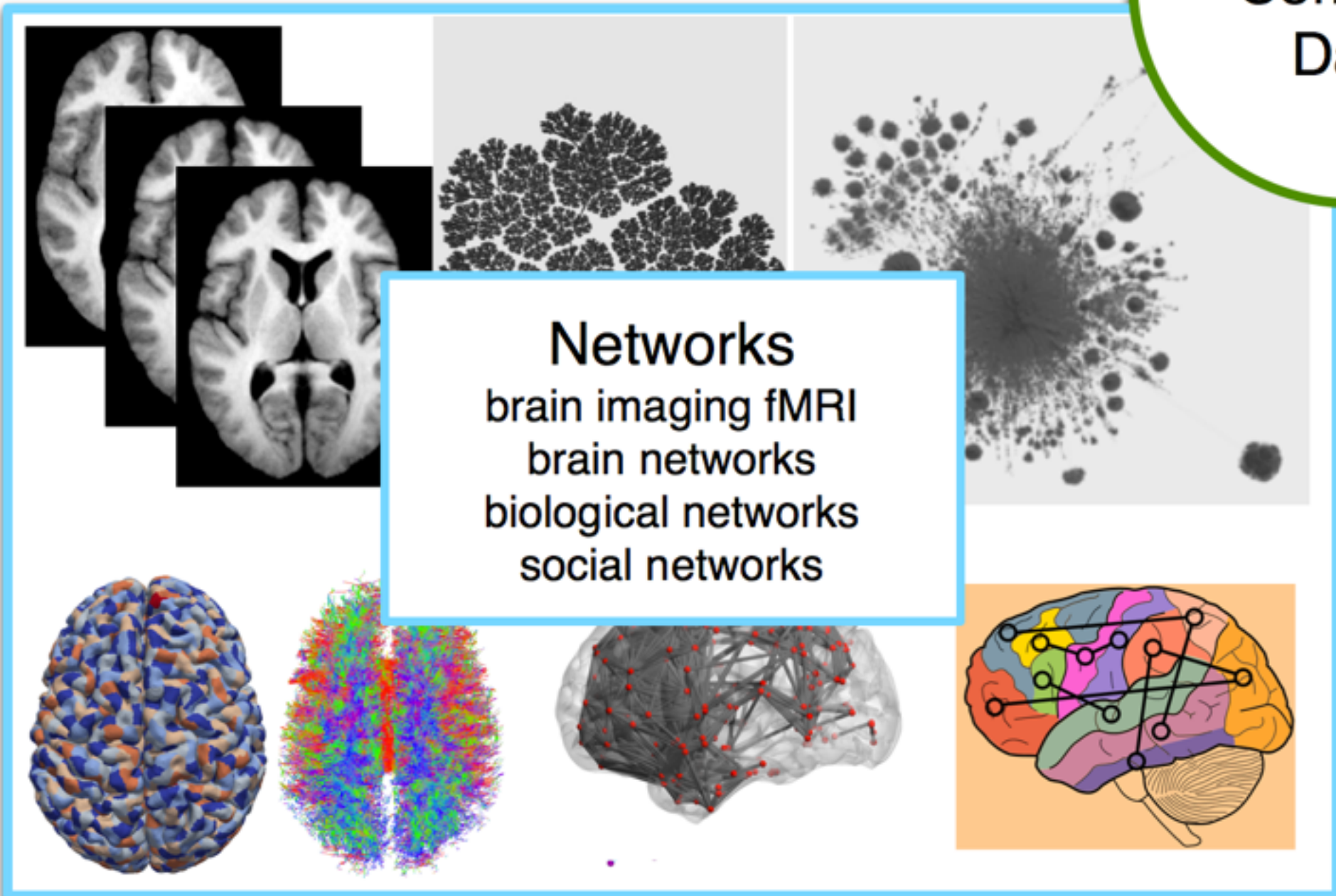


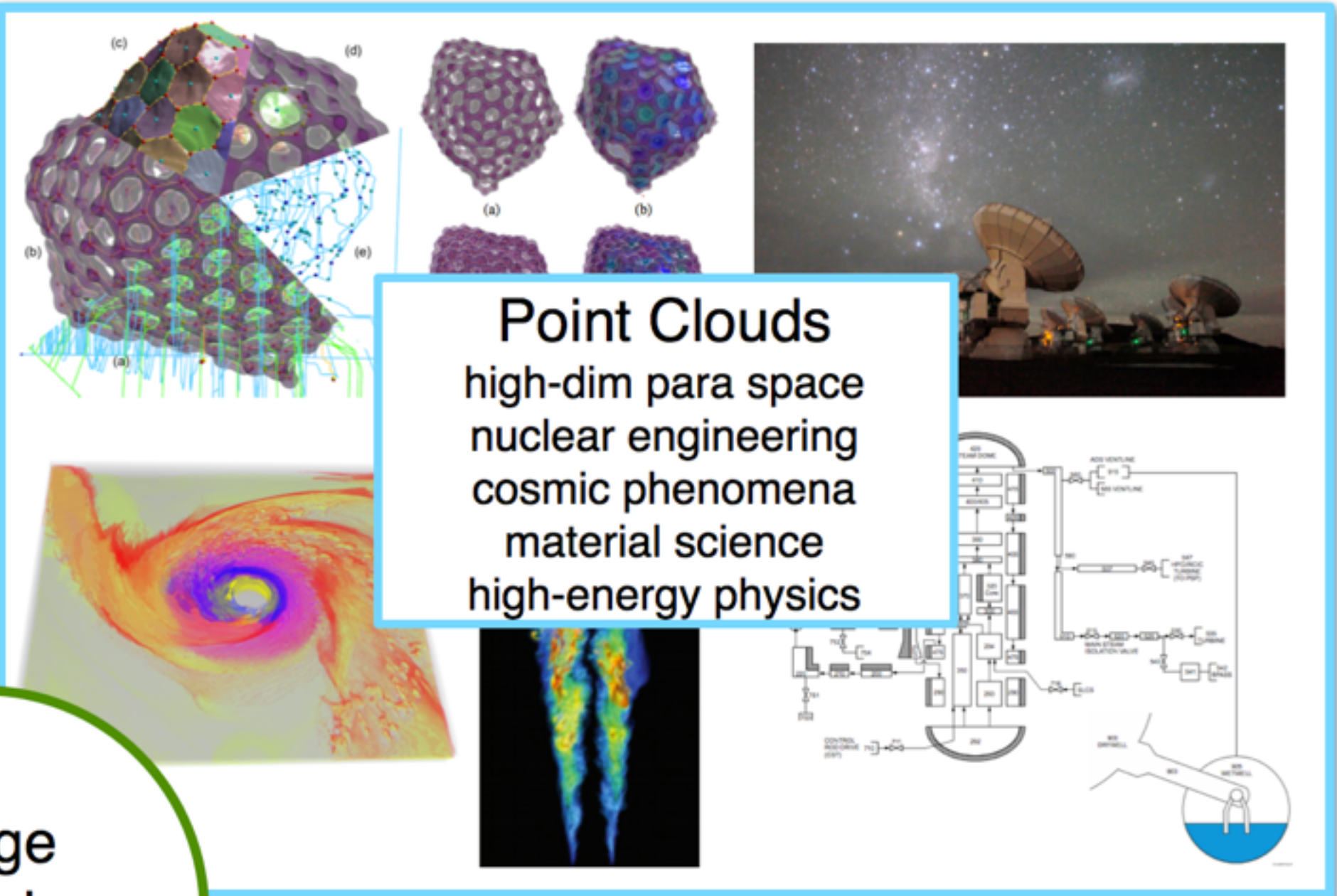
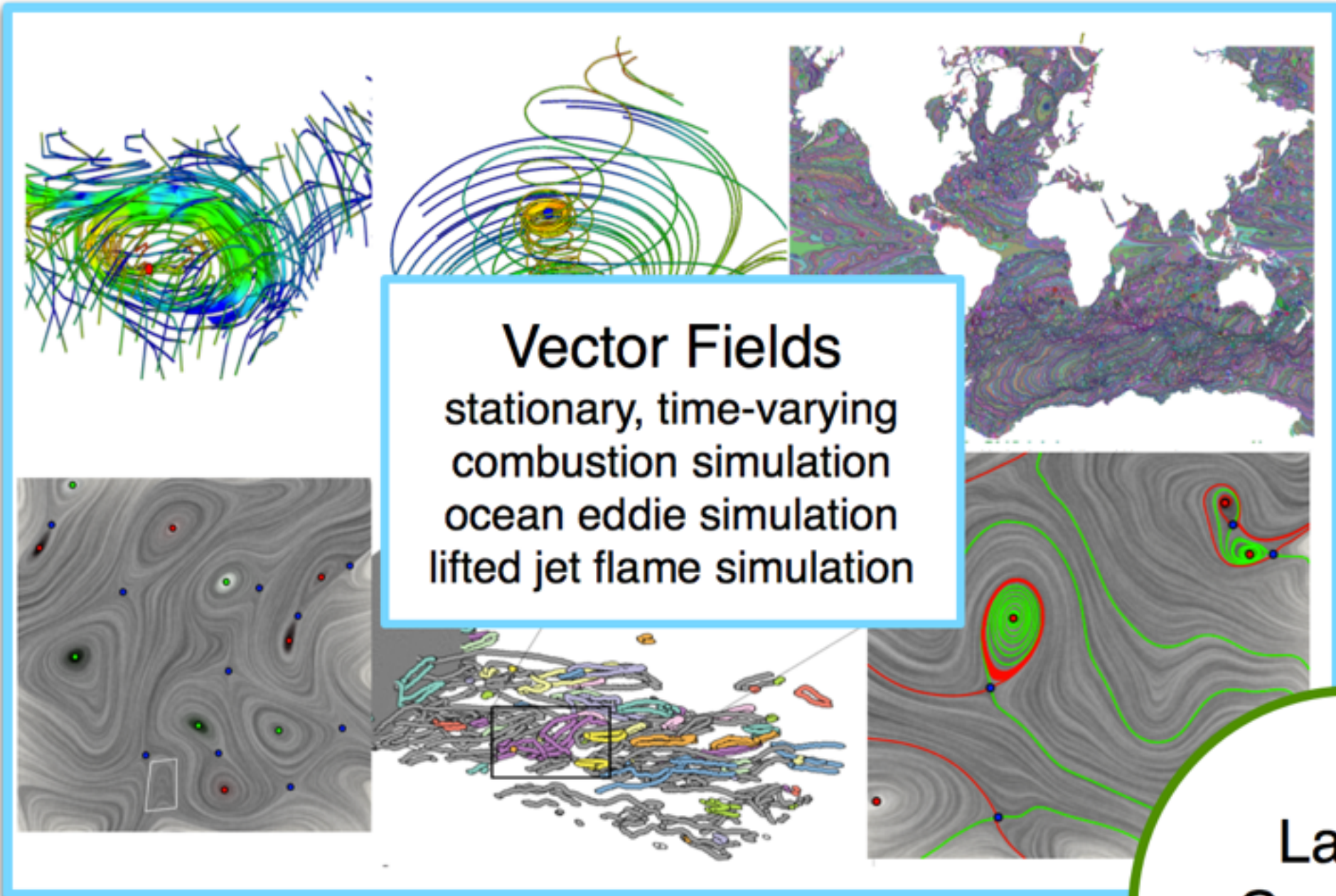
Large
 Complex
 Data



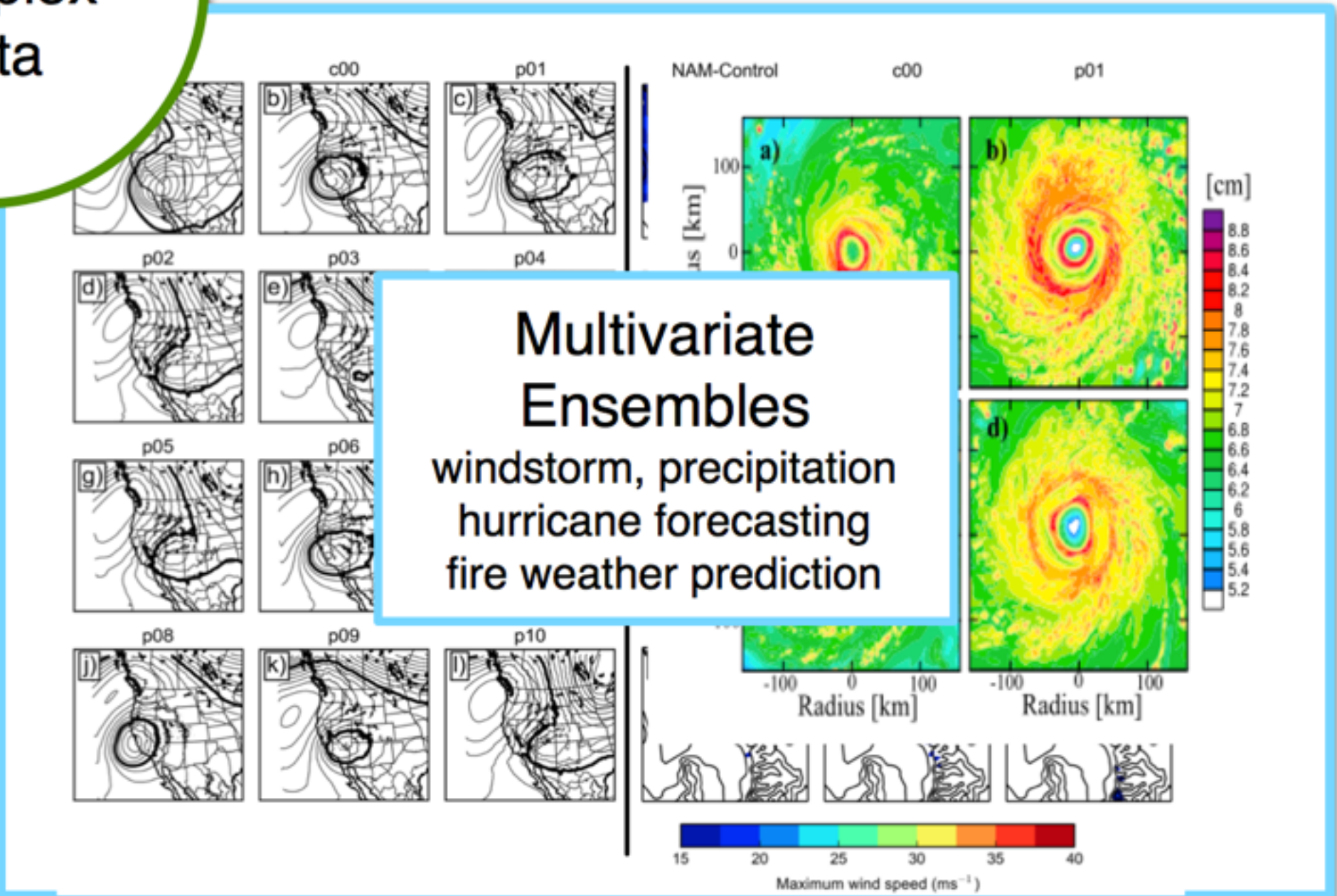
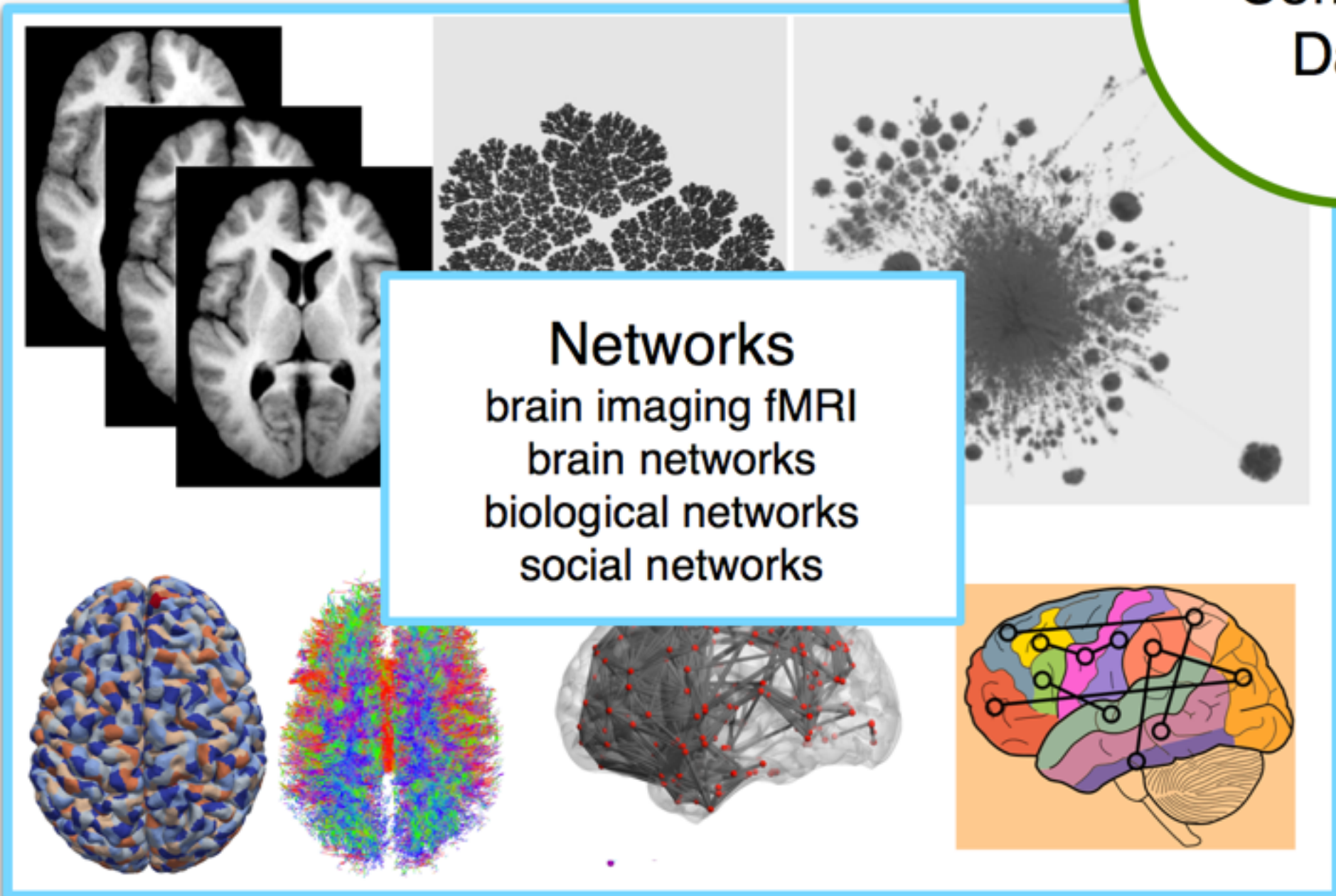


Large
Complex
Data





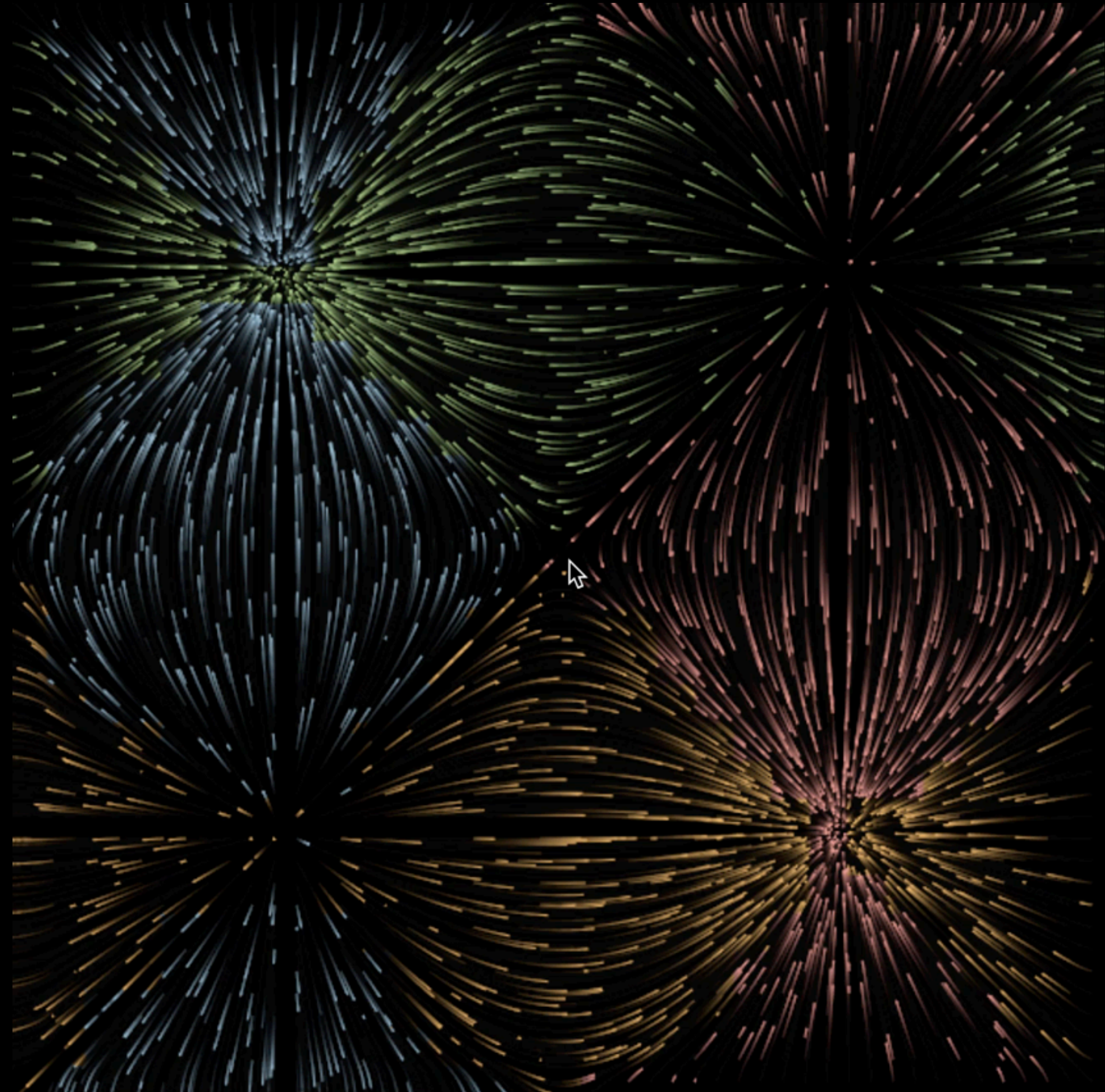
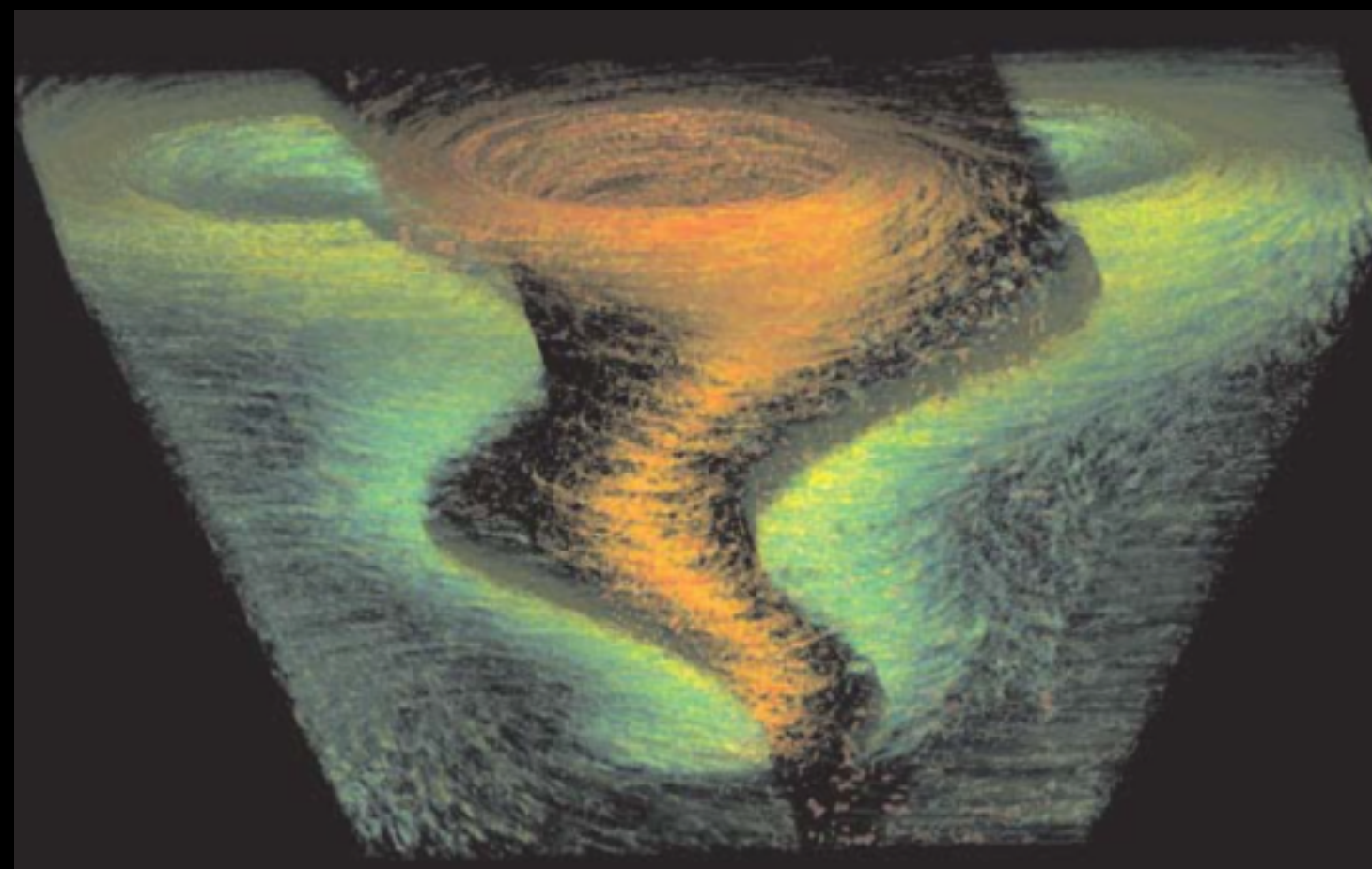
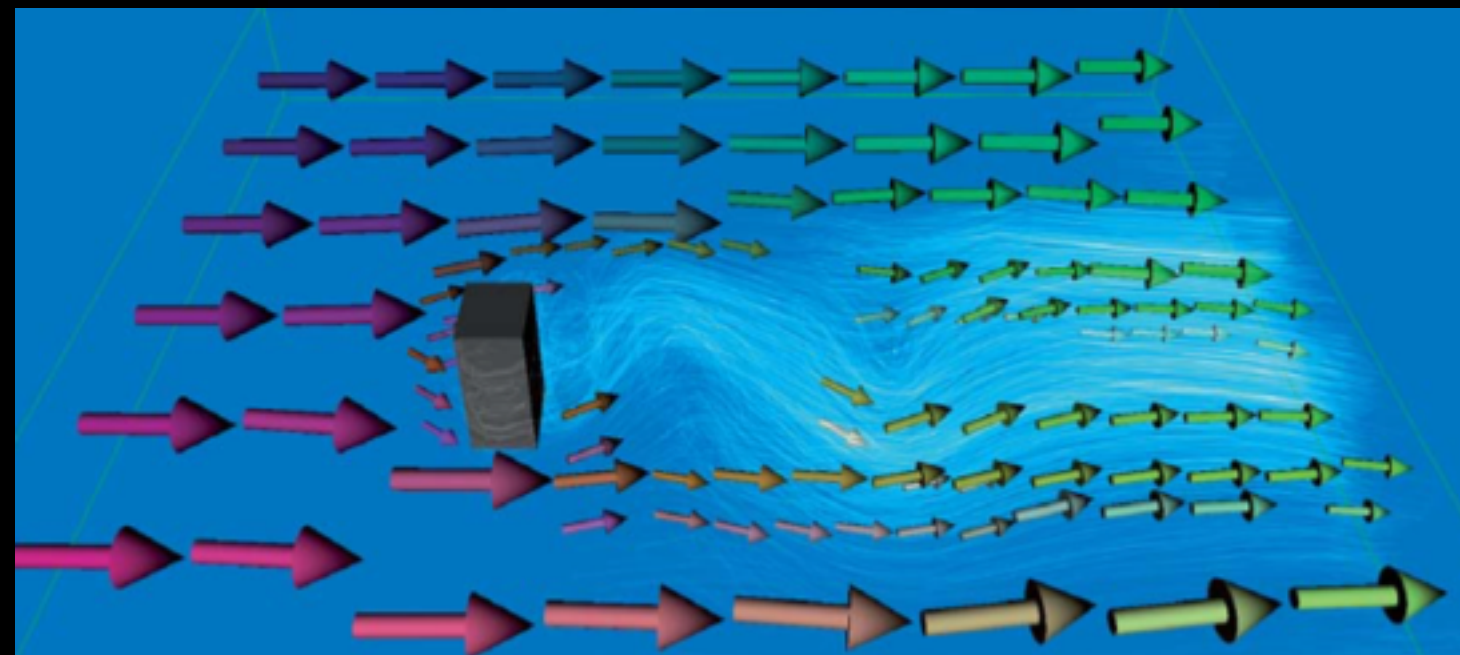
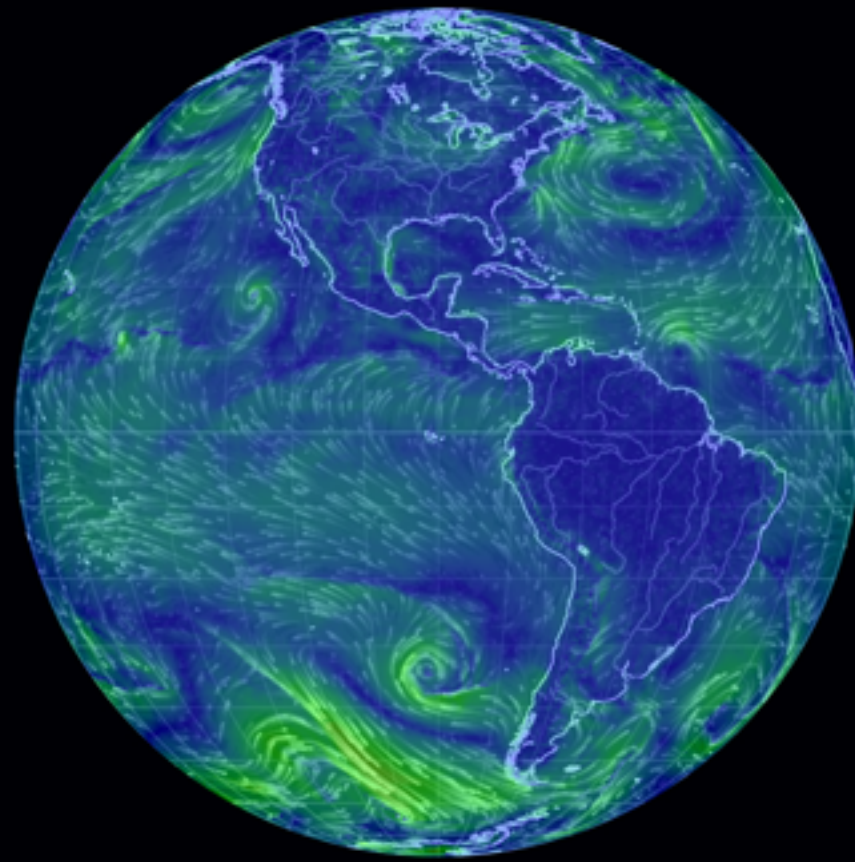
Large
Complex
Data



Vector Fields

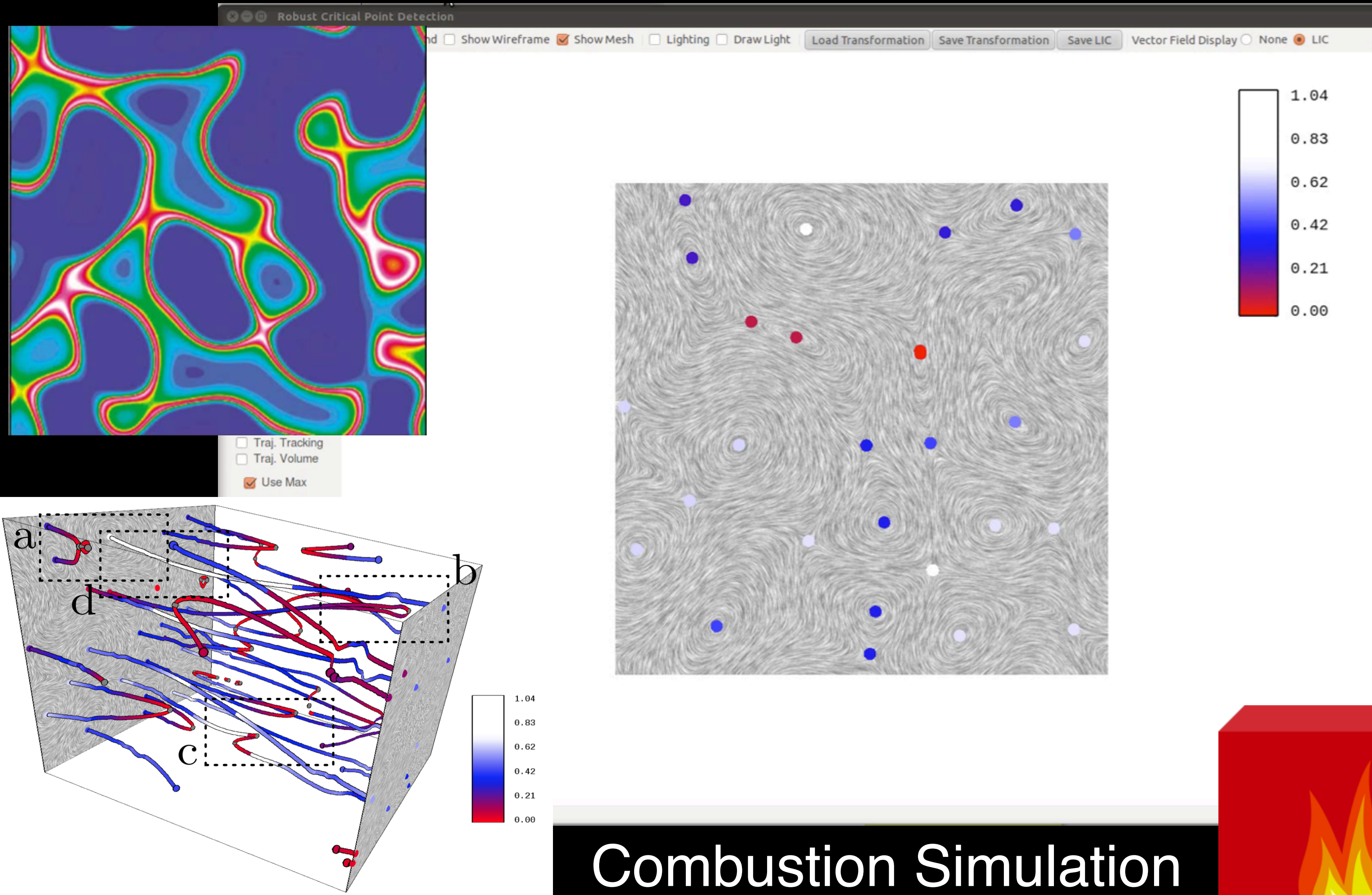
Combustion and Ocean

Make the flow patterns visible & Interpretable



Sources: Dan Maljovec, Cameron Beccario, [Correa, Silver, Chen 2007]
[Burger, Kondratieva, Kruger, Westermann 2008]

Quantify feature stability

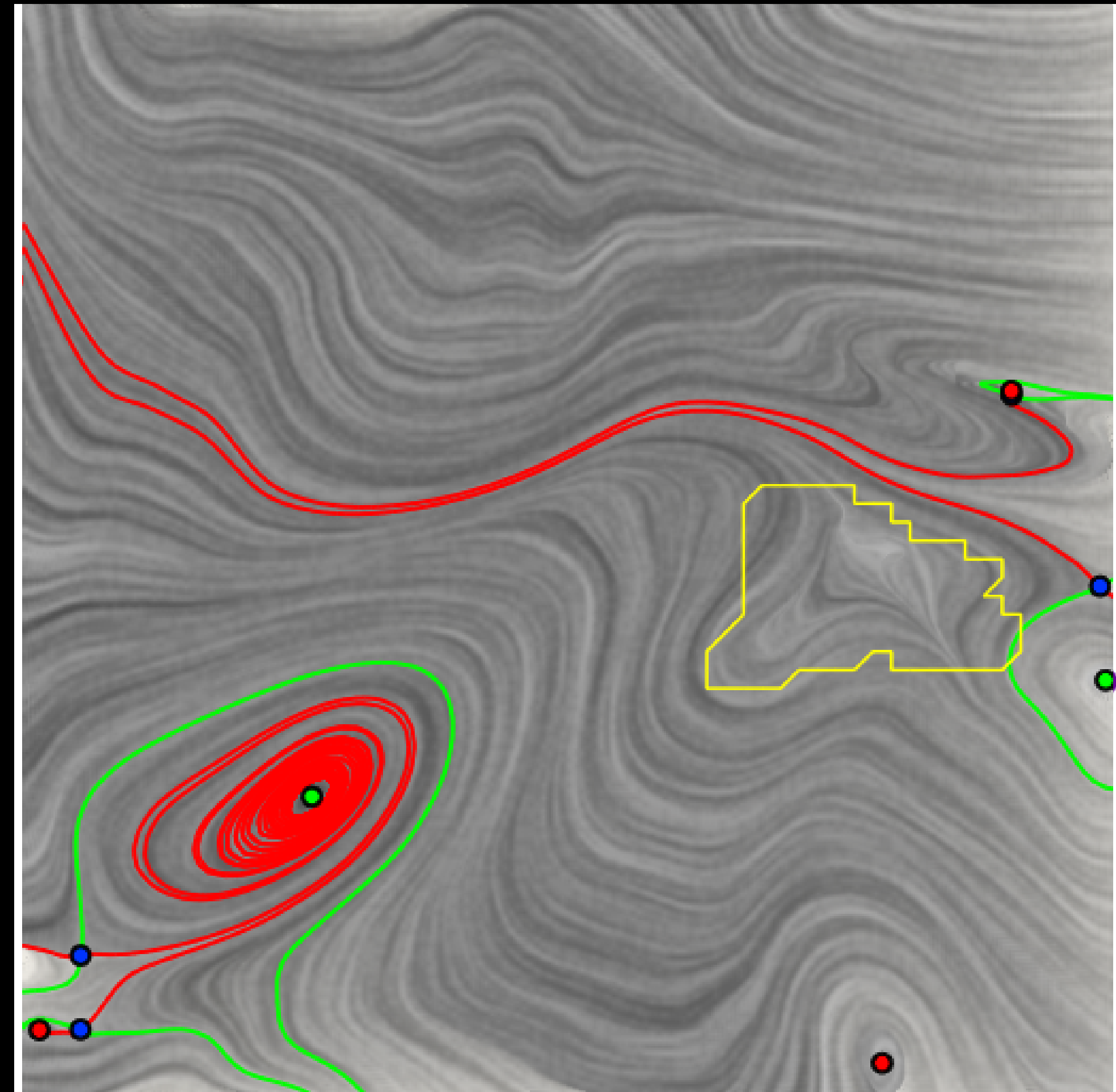
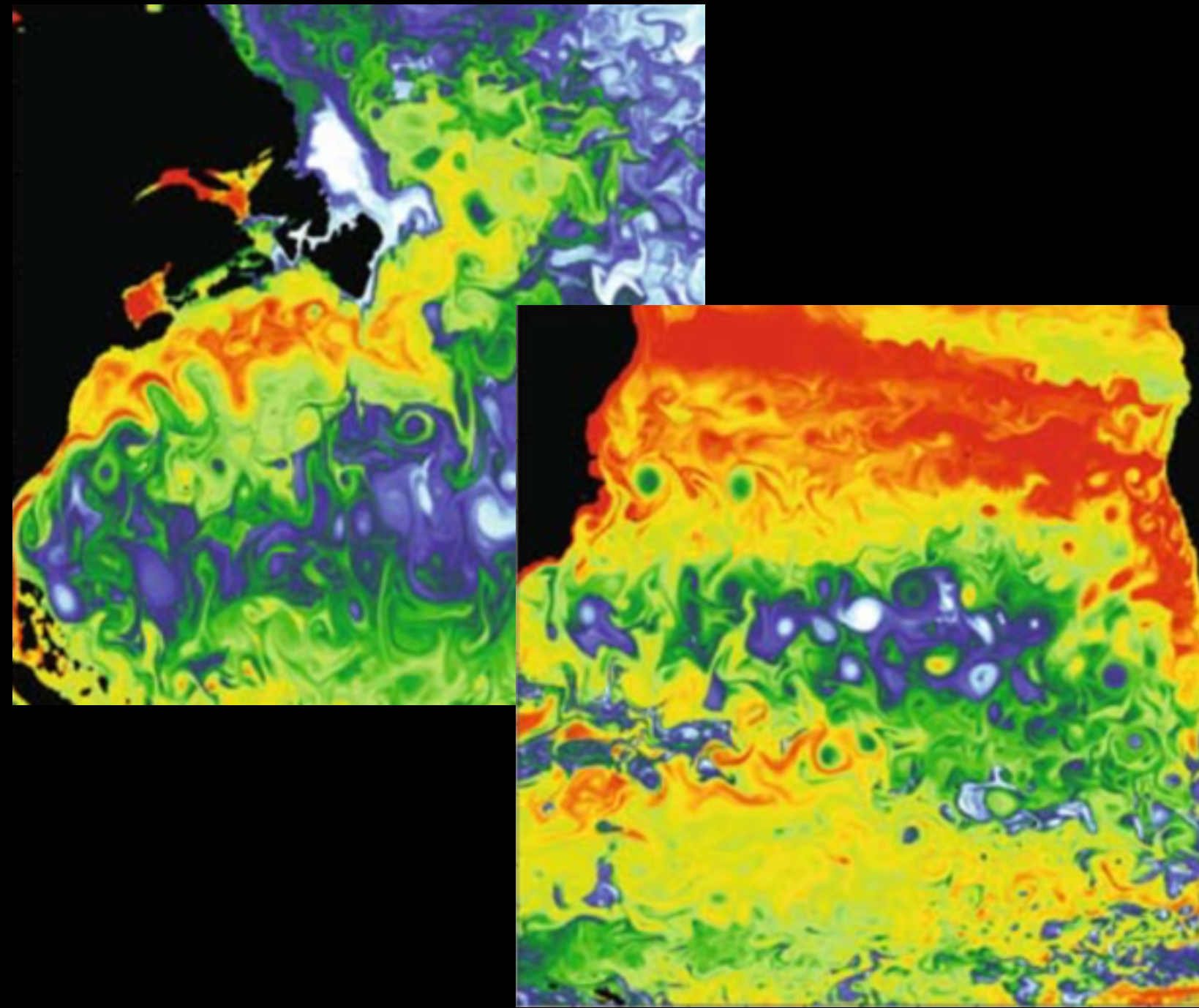


Combustion Simulation

Simulation: [Hawkes, Sankaran, Pebay, Chen 2006]



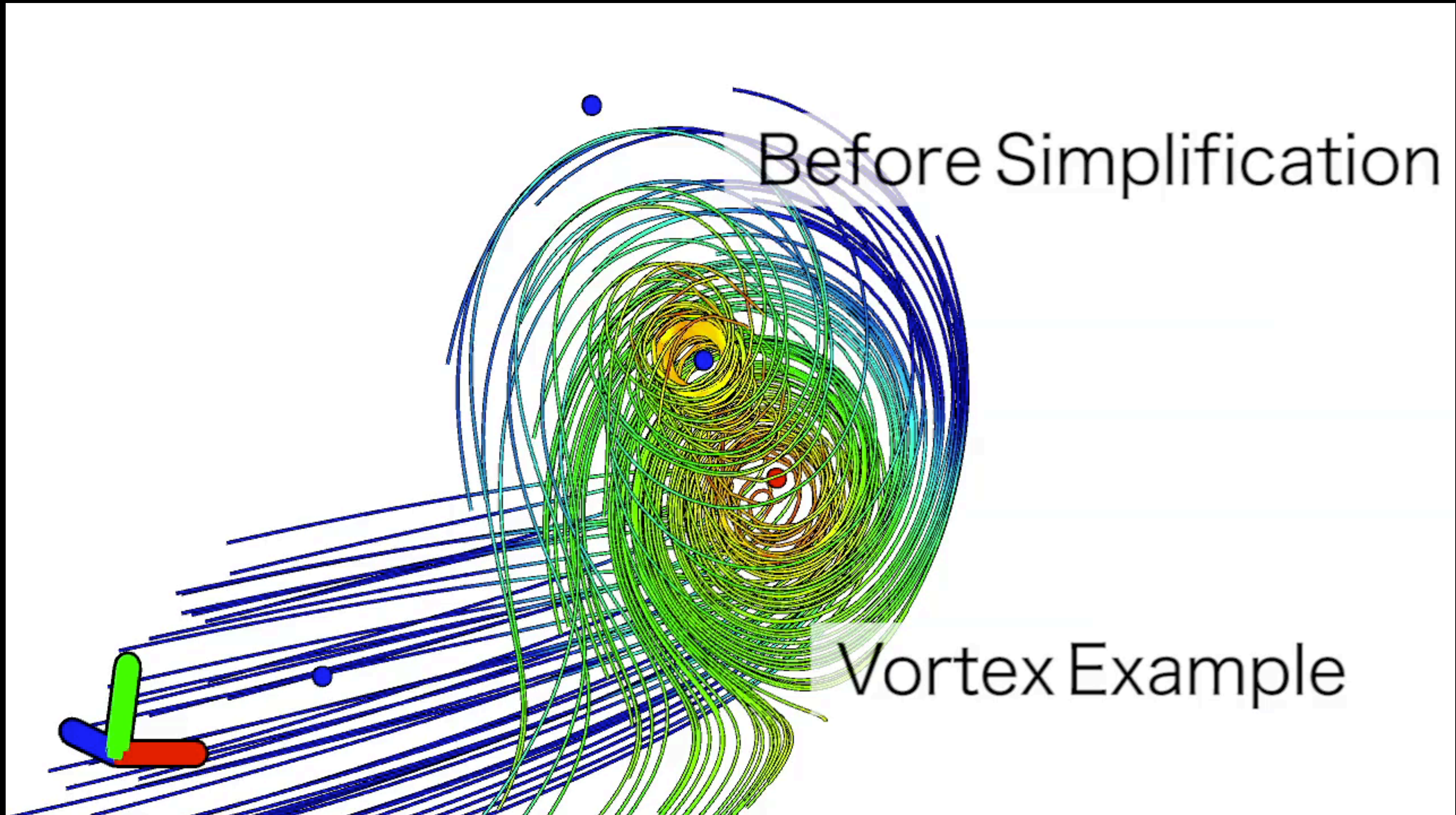
Separate features from noise at multi-scale



Ocean Eddy Simulation

Map: Courtesy of SlidesCarnival & Unsplash
Simulation: [Maltrud, Bryan, Peacock 2010]

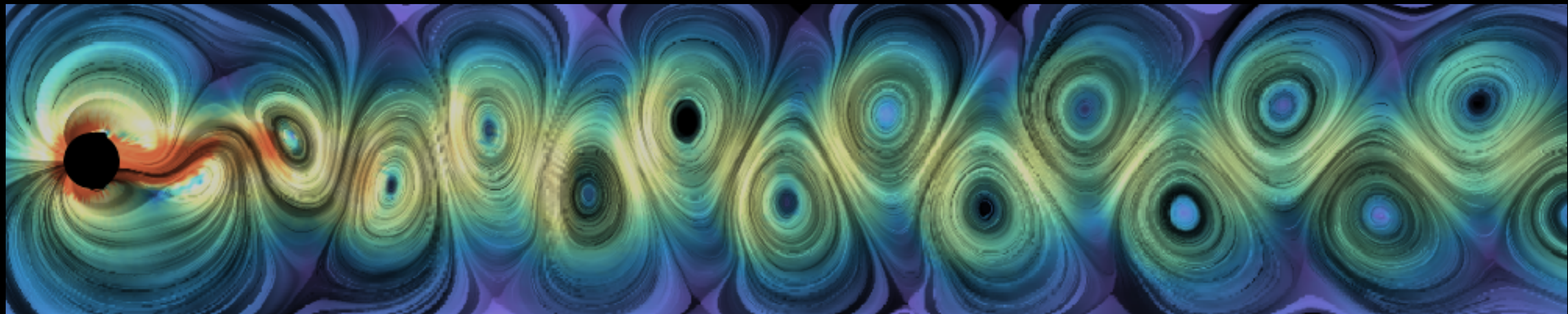
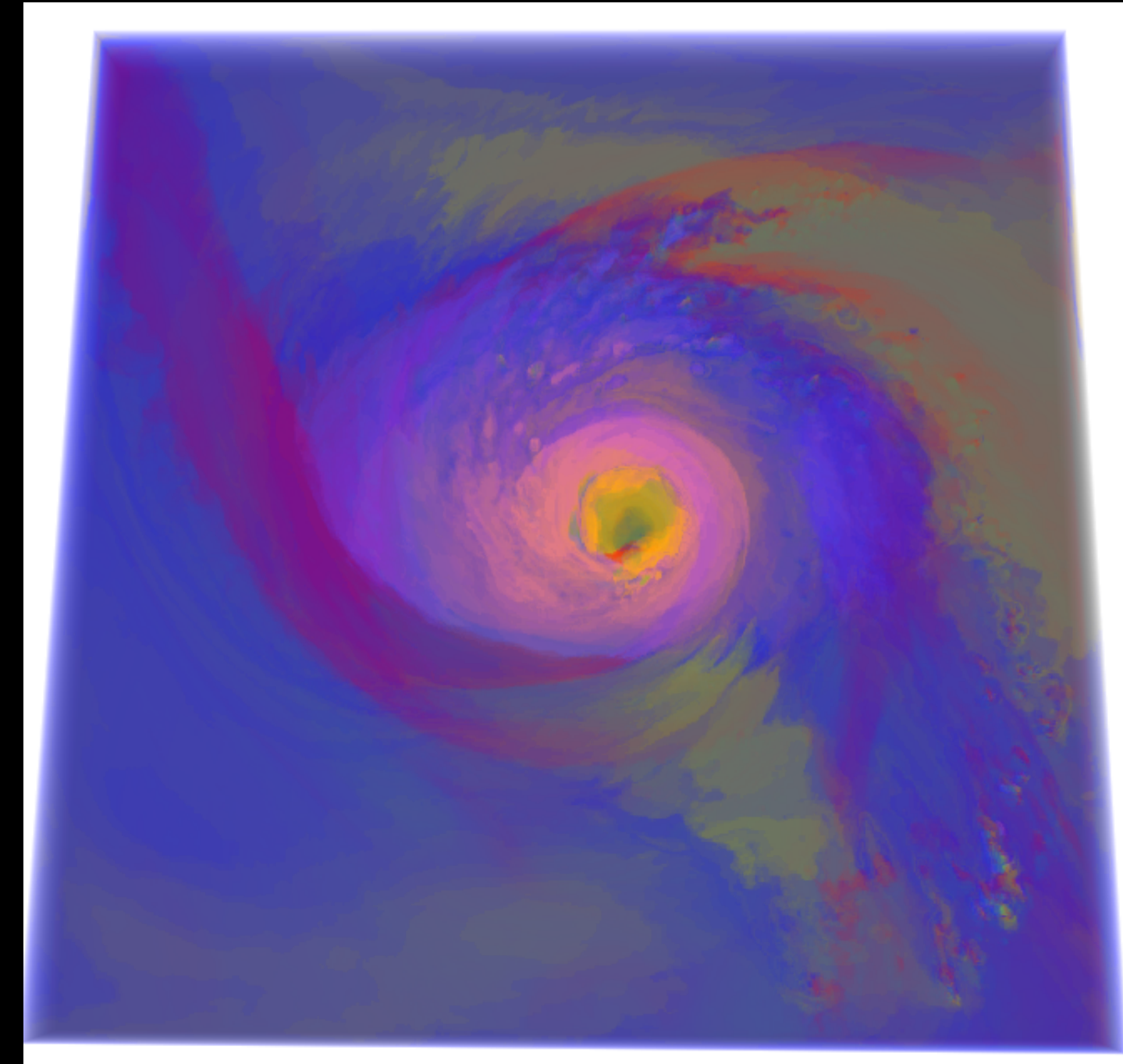
Visualize flow in 3D



Understand turbulent flow



Source: NASA

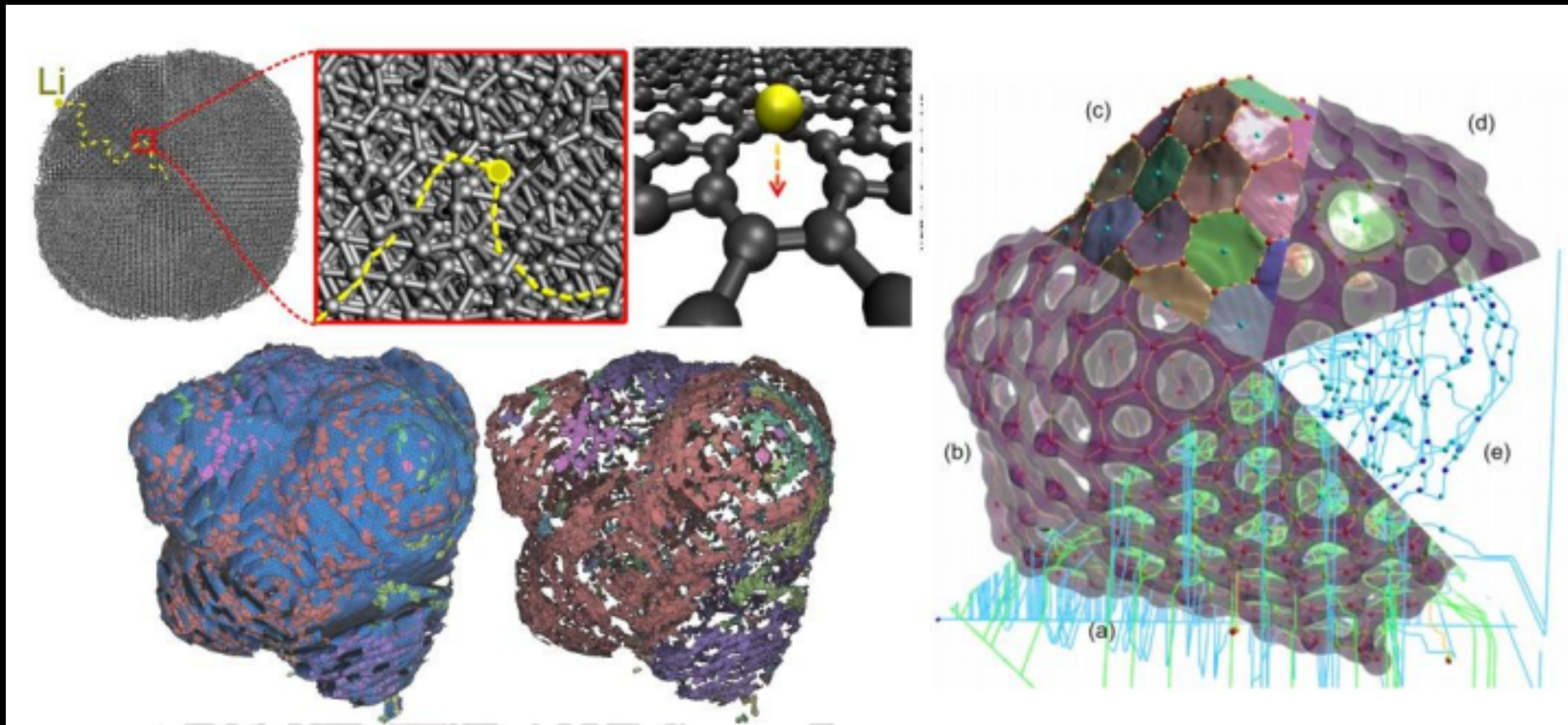


Material Science
Your iPhone Battery



How long can
your battery last?

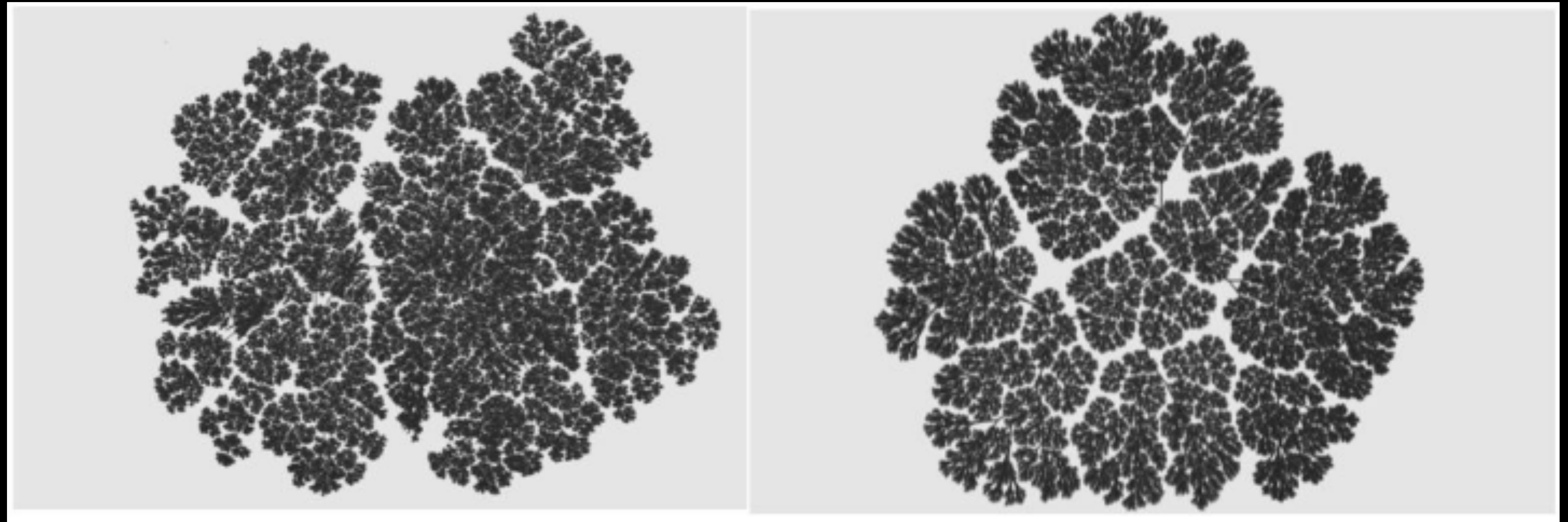
Ion diffusion geometry extraction in battery



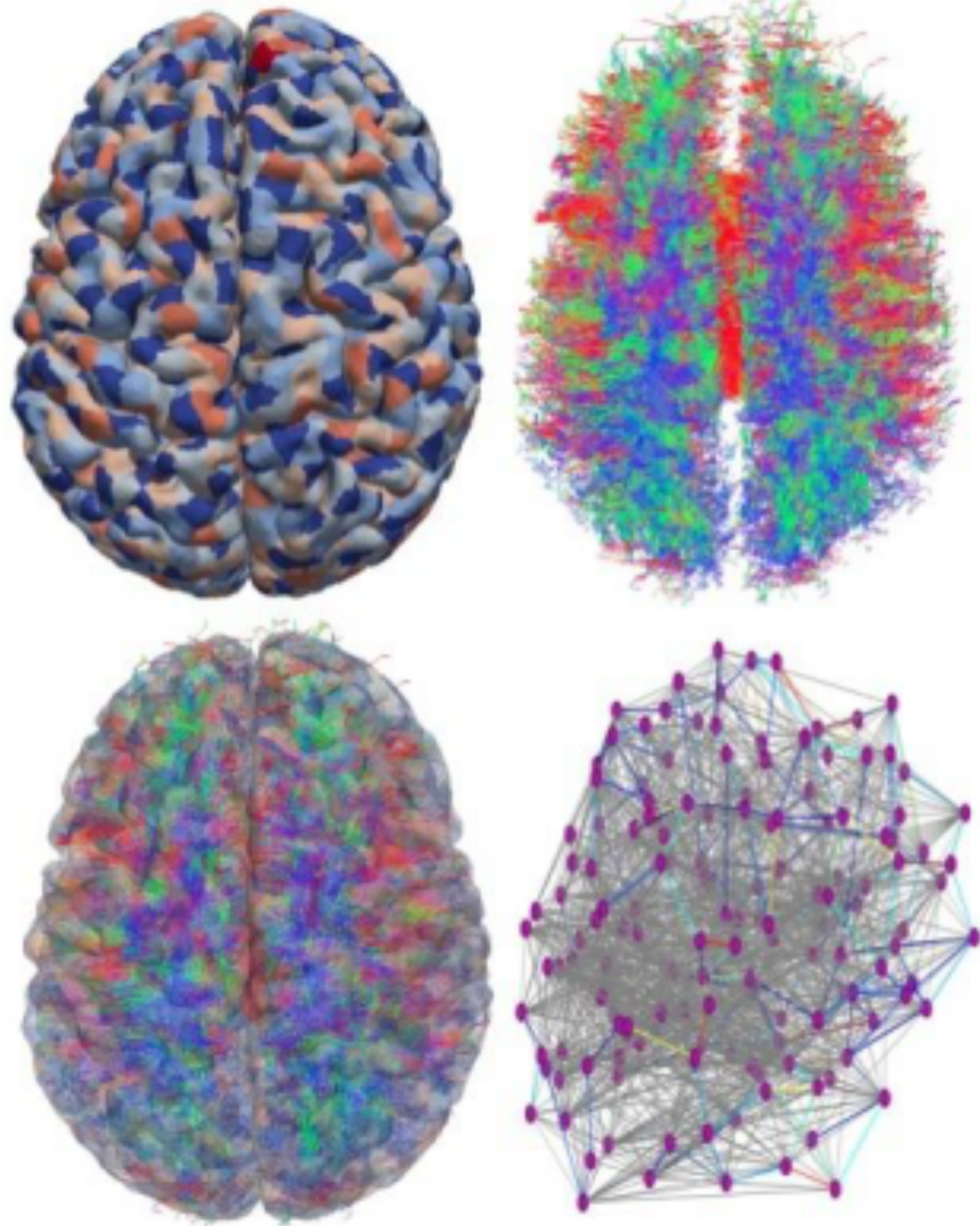
Networks

Brain networks

Inadequate Network Visualization

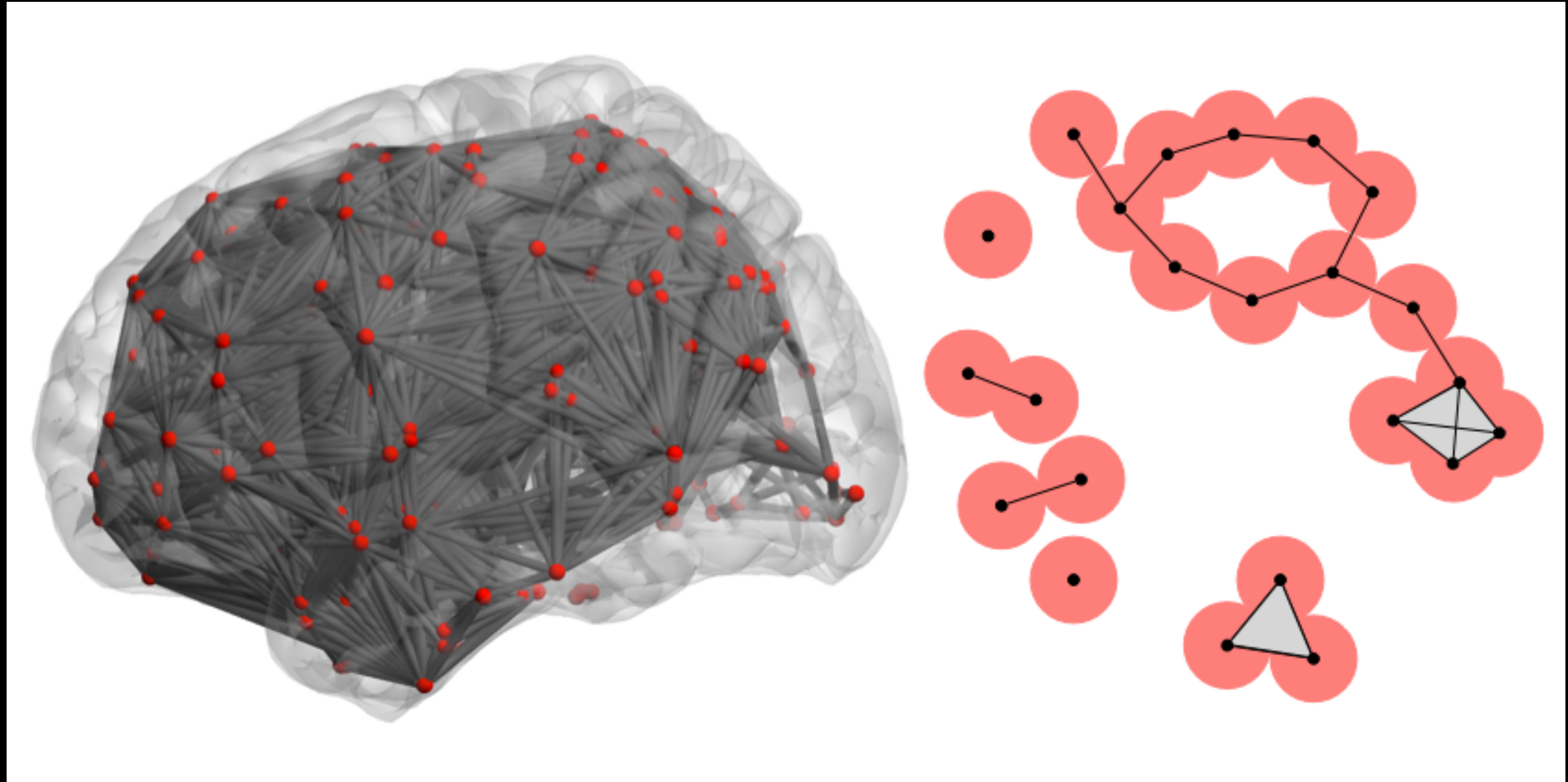


Brain Network Visualization



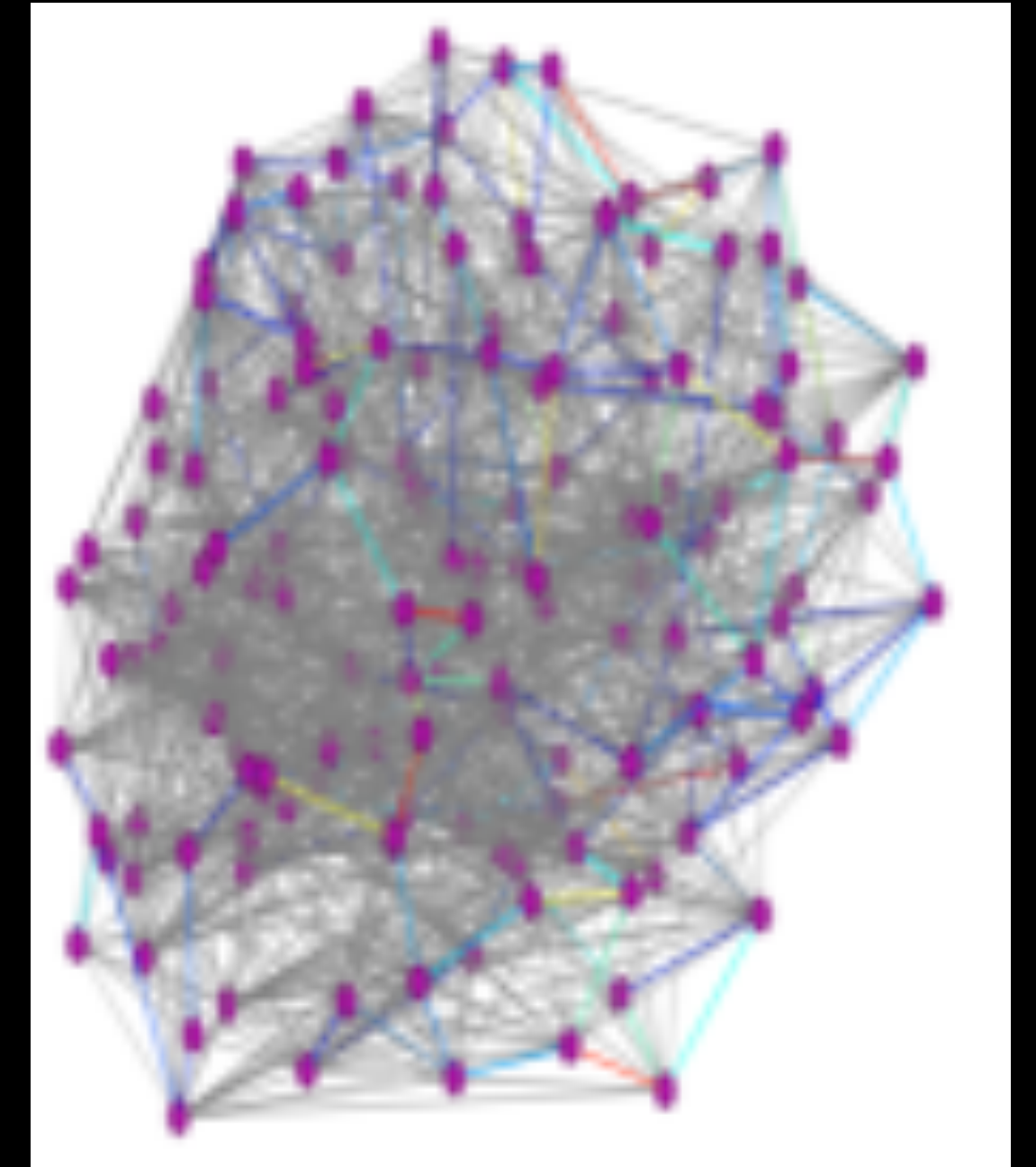
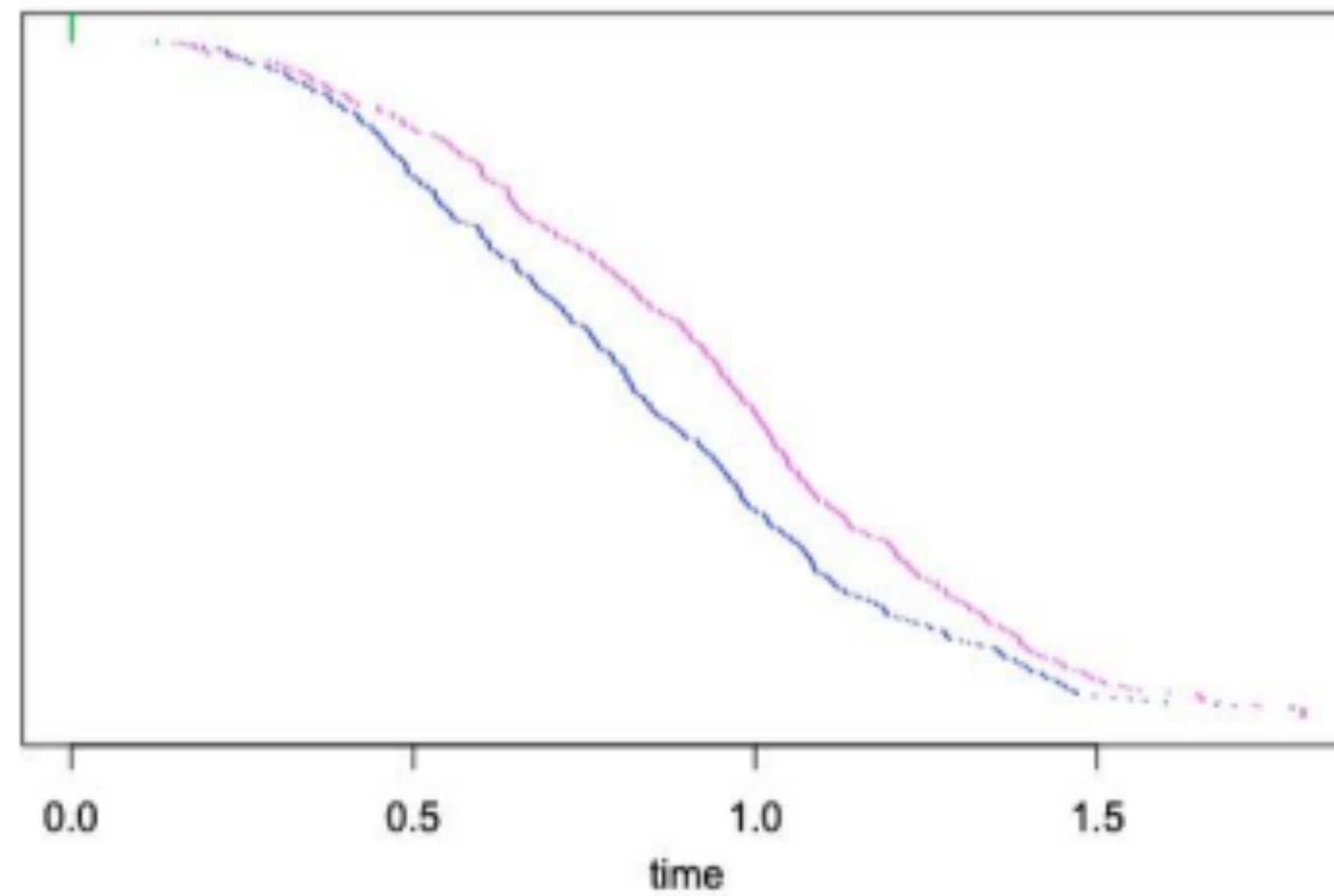
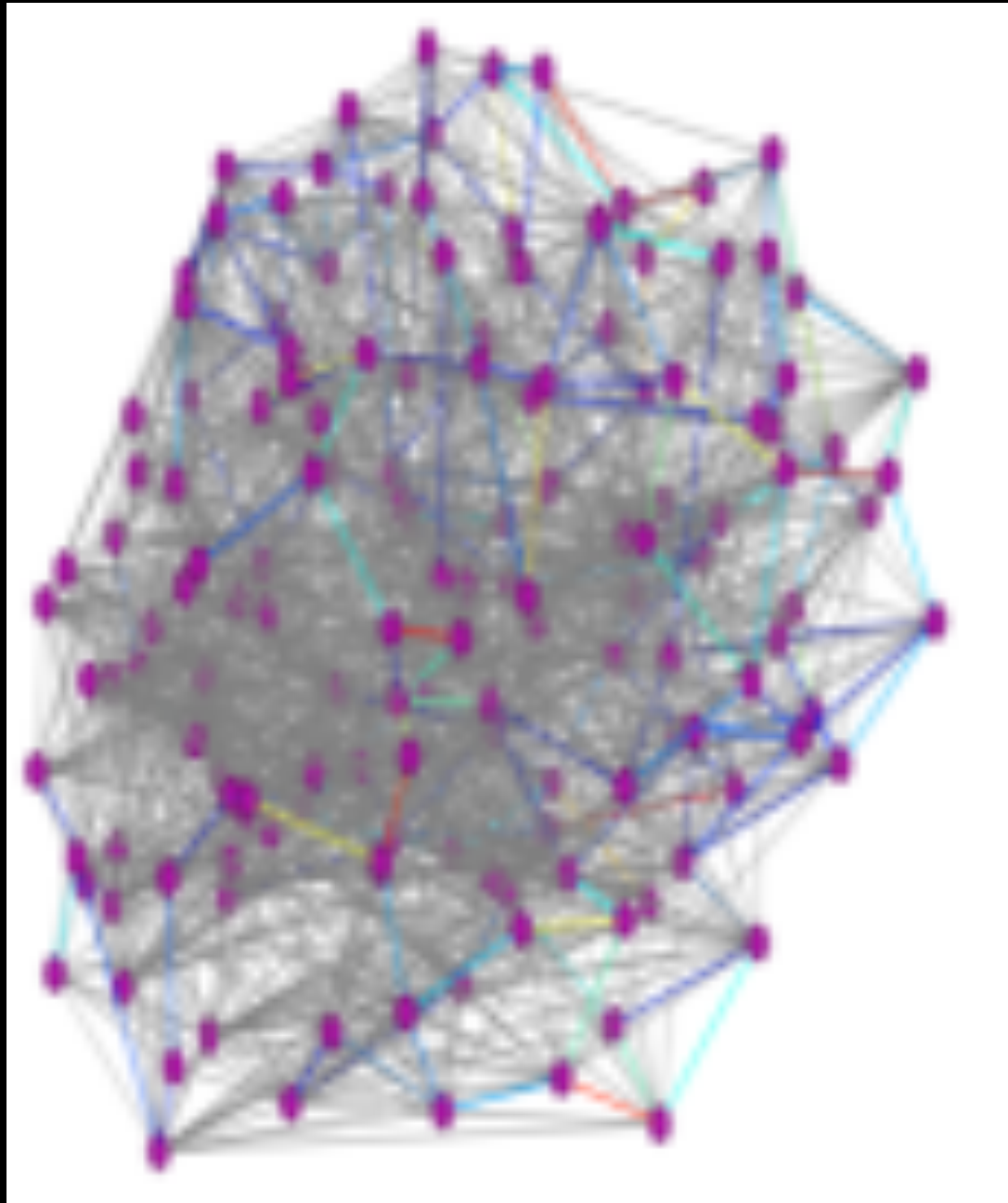
Avoid network hairballs
while preserving structure?

Topology and brain networks

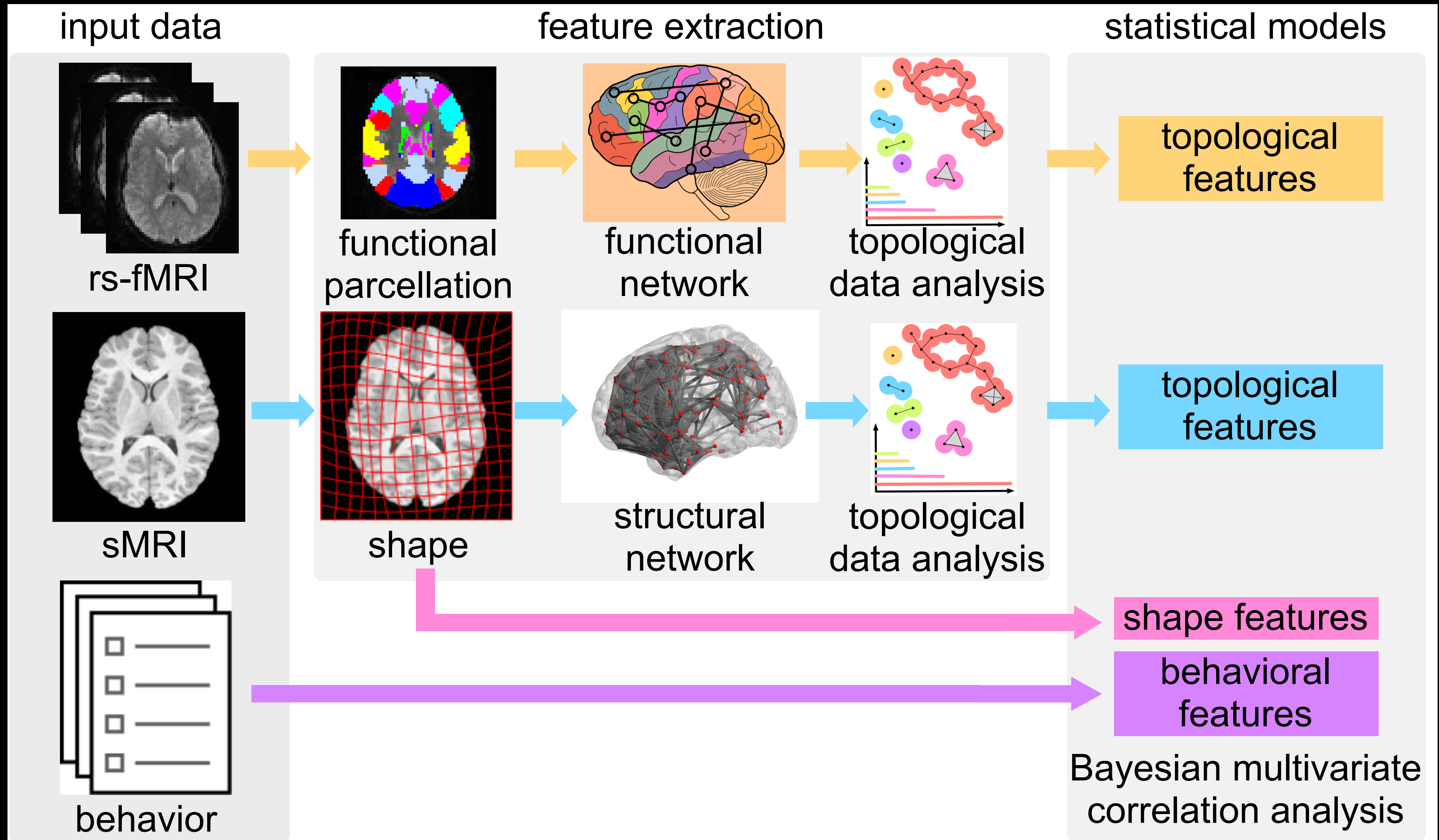


Autism Brain Networks

Can we tell autism subject from control?



Autism Brain Networks



Astronomy

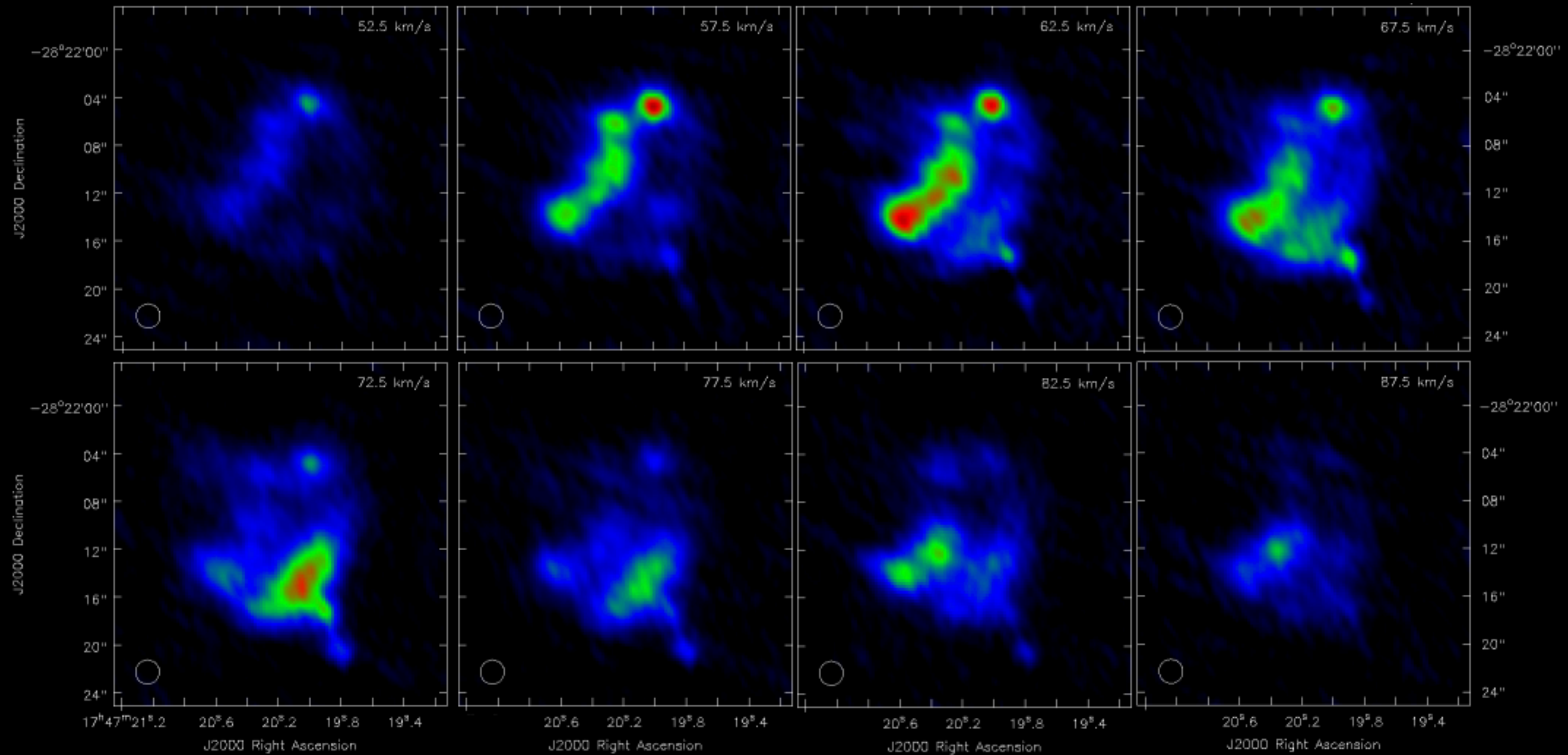
Telescopes and Black Holes

Largest radio telescopes in the world



Credit: ALMA

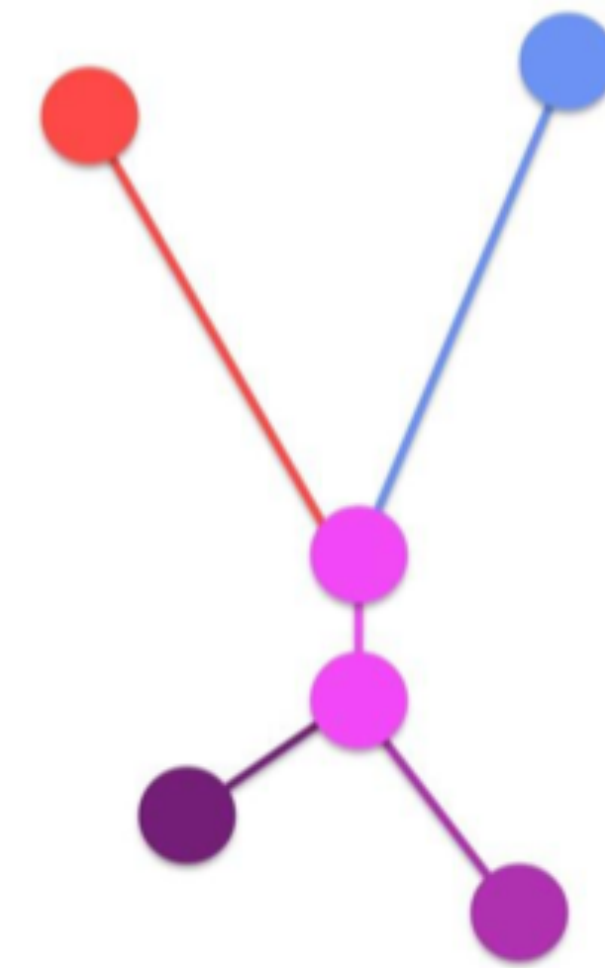
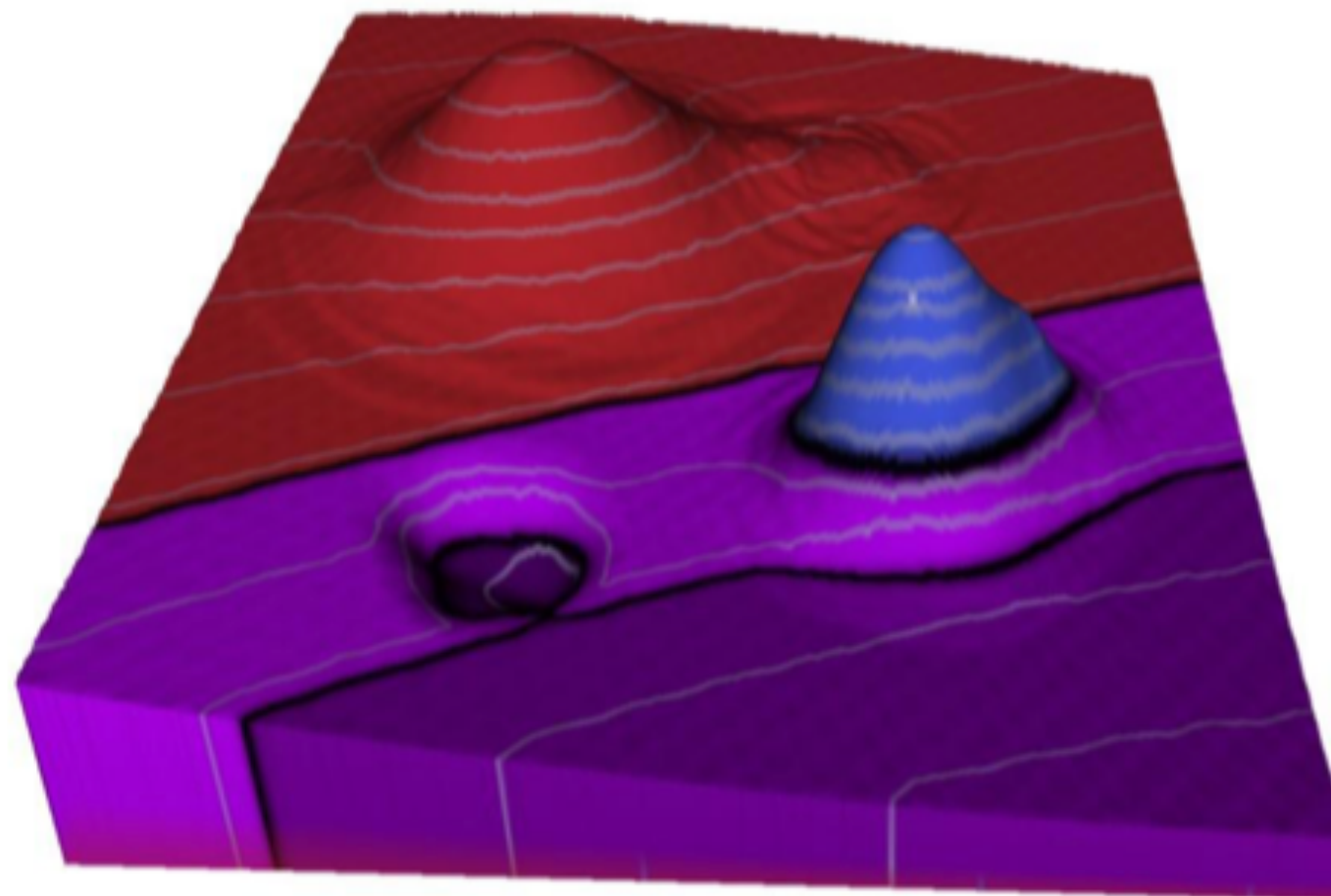
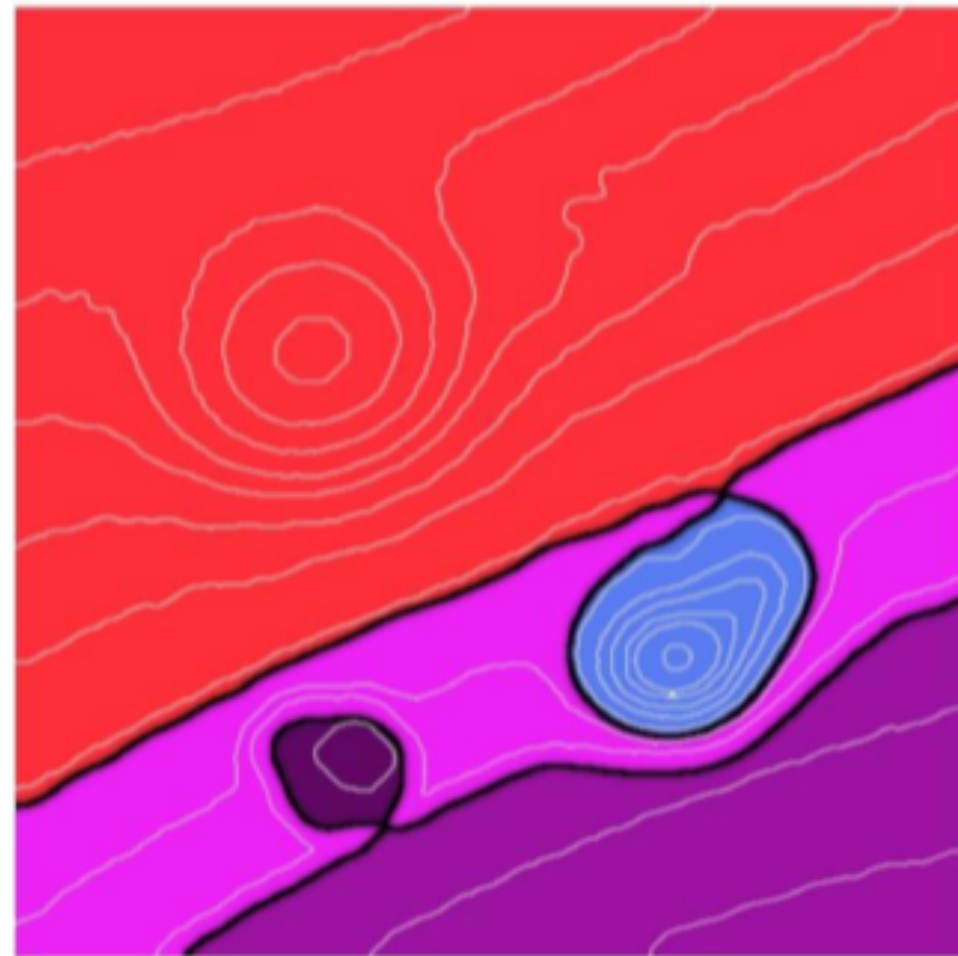
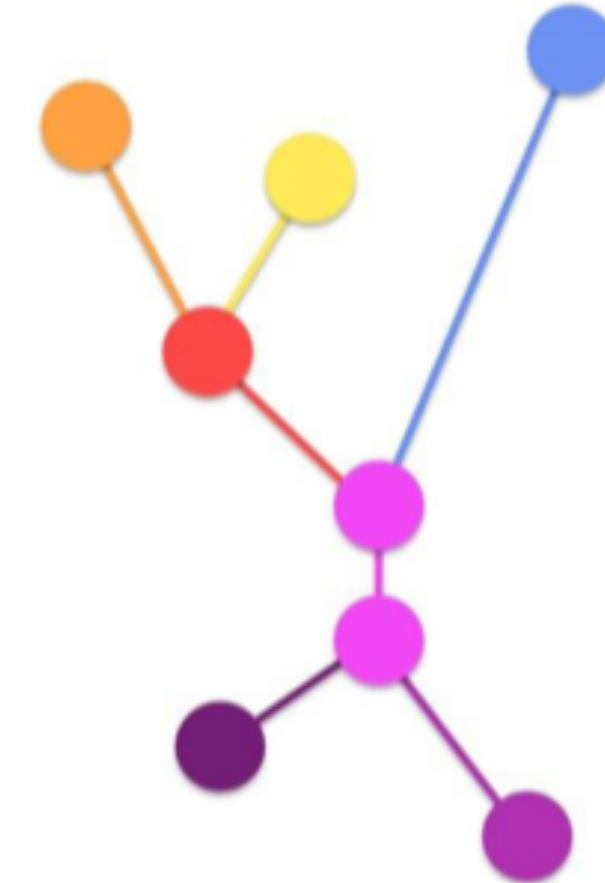
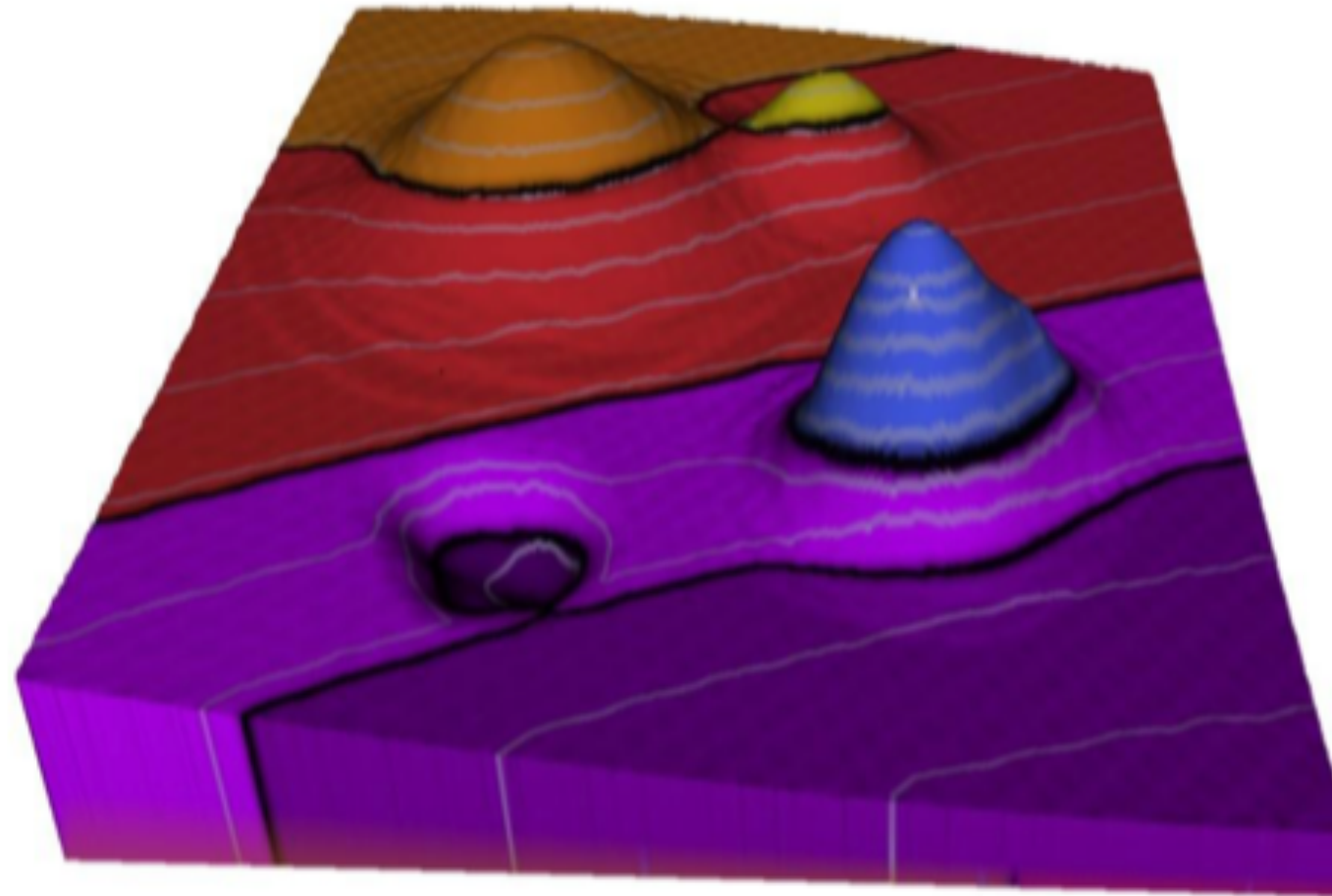
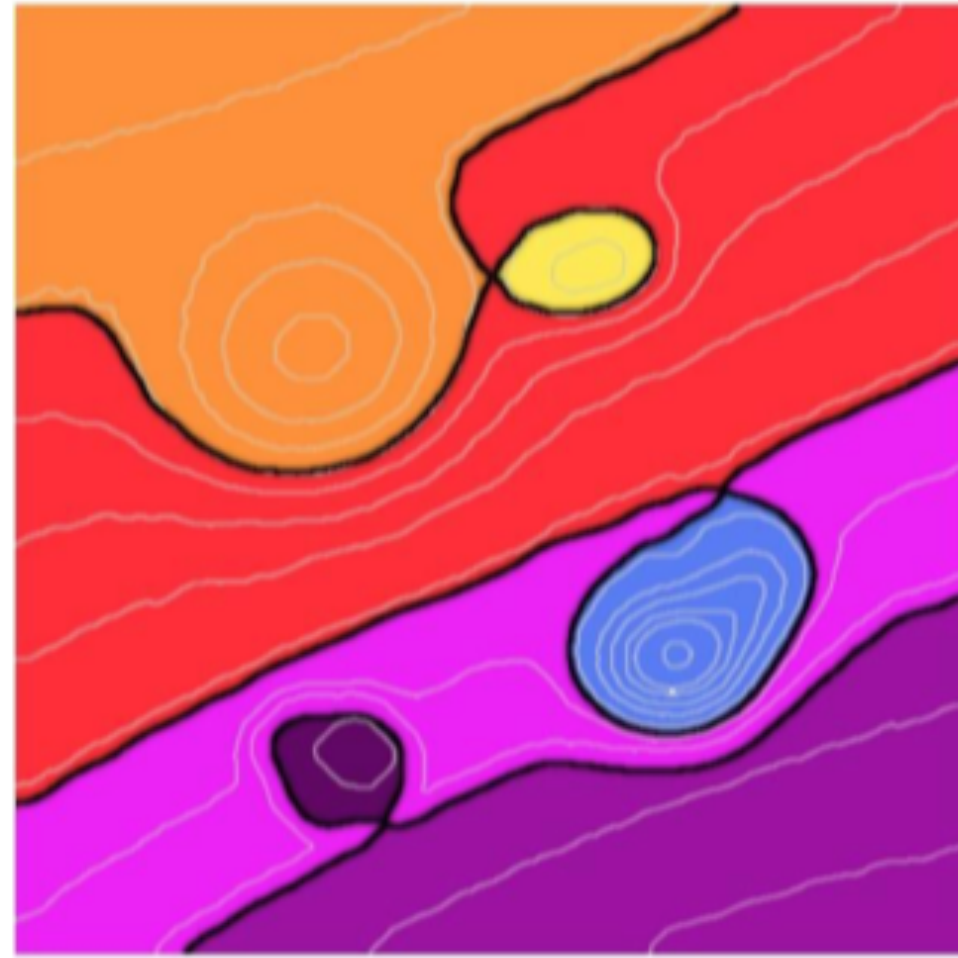
Radio telescope Data

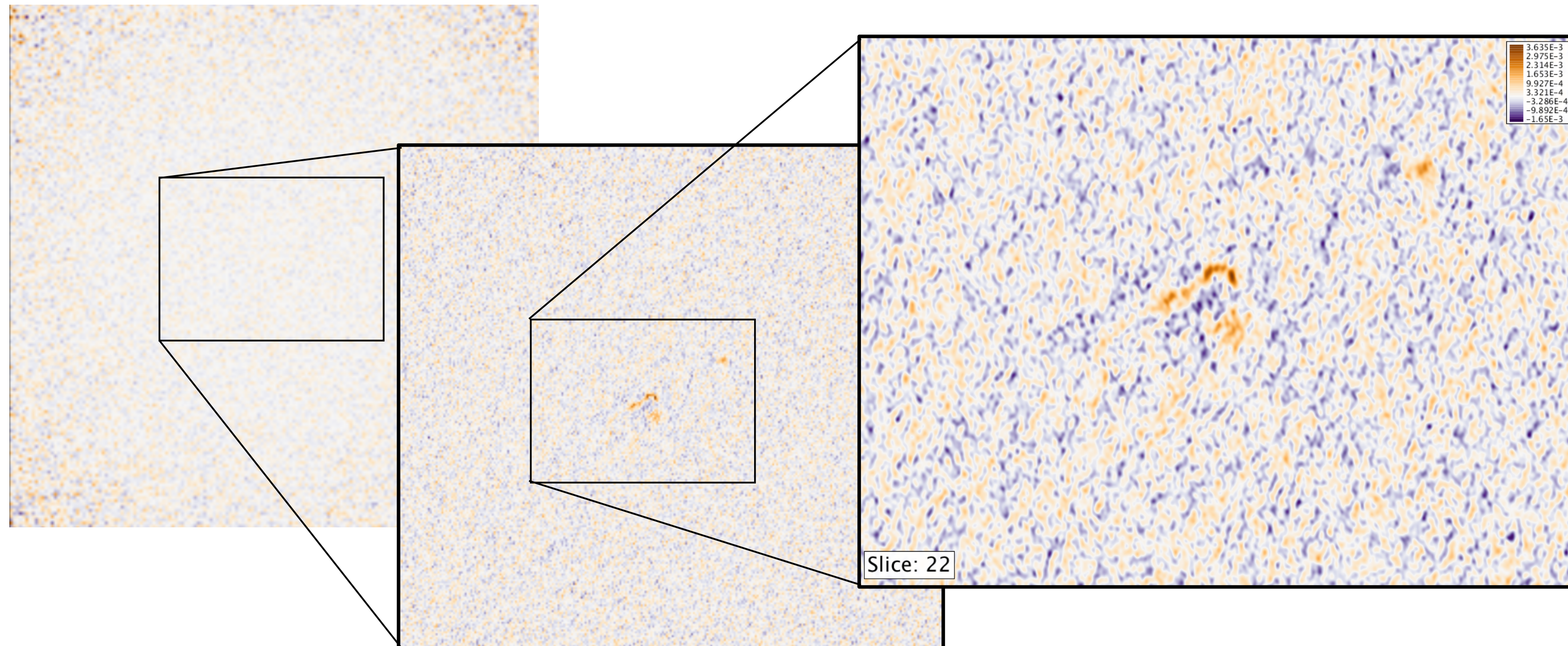


NGC 404: Mirach's Ghost Galaxy

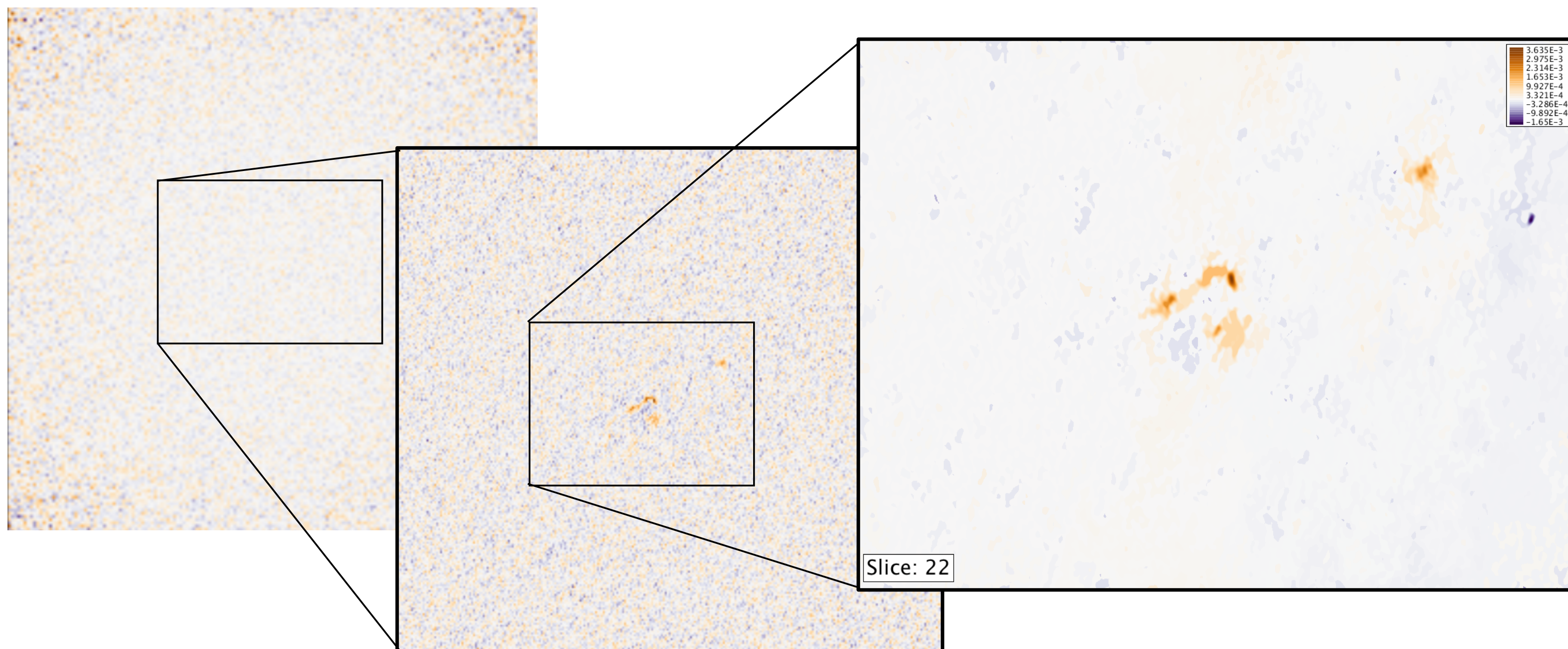


Feature Denoting and Source Finding

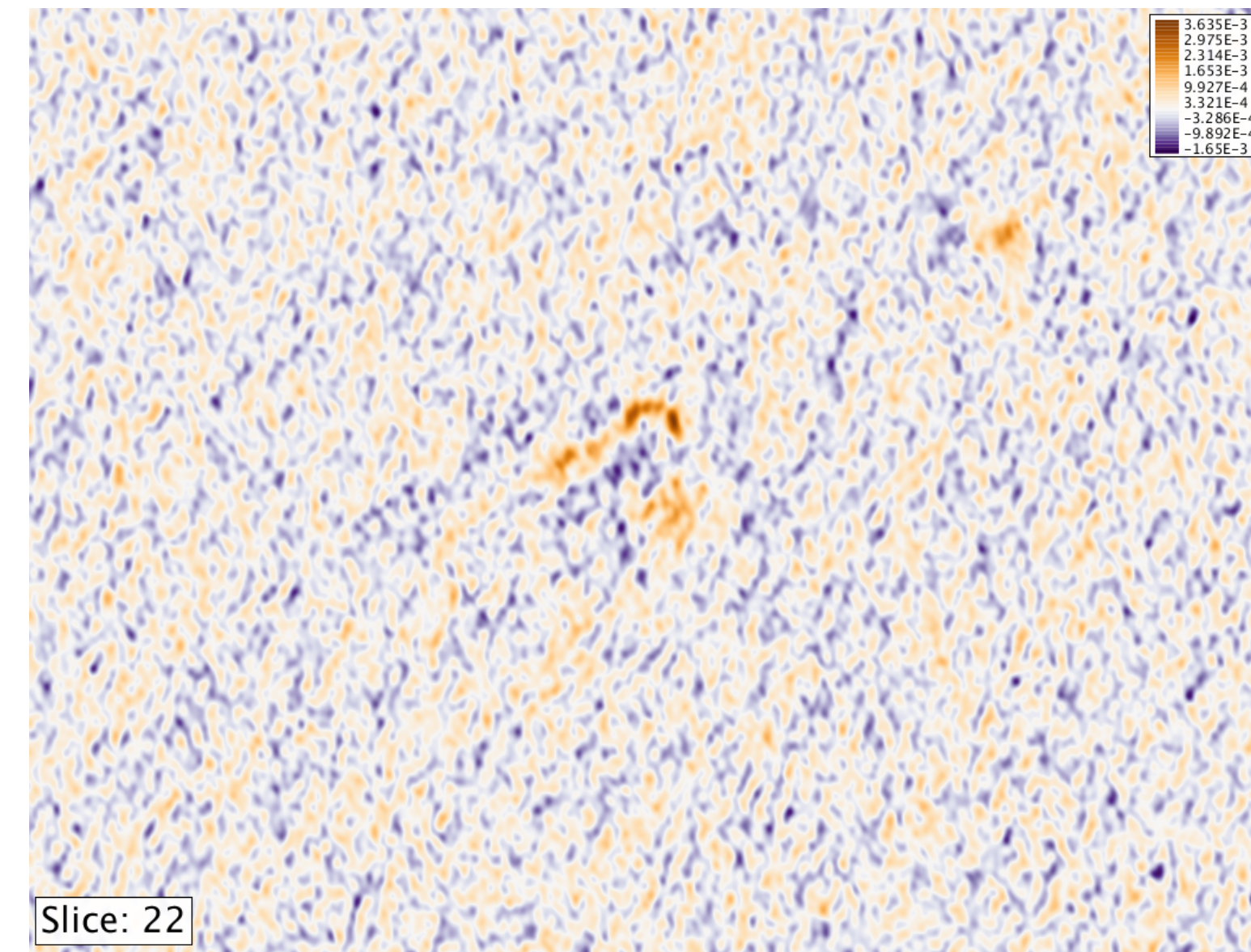
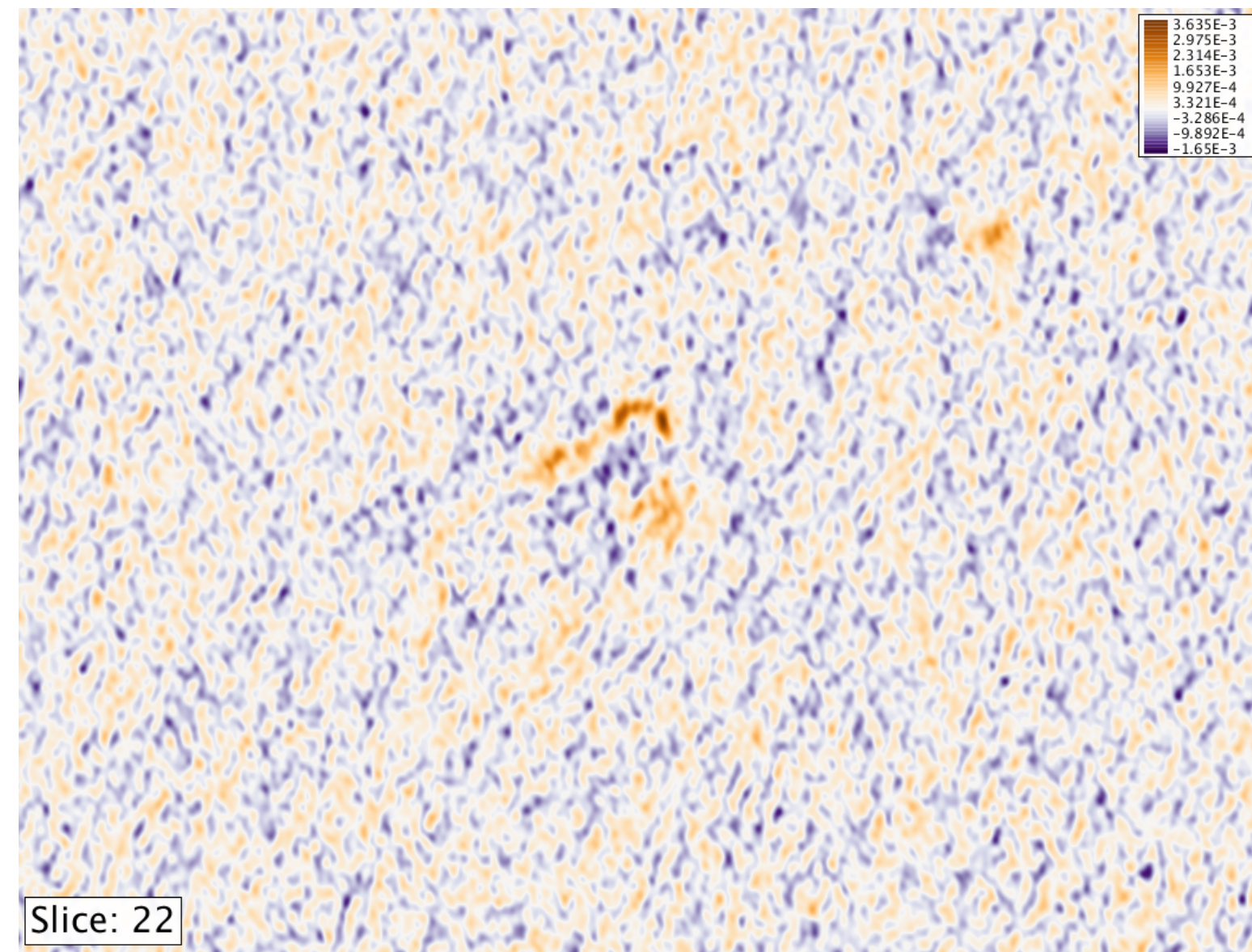
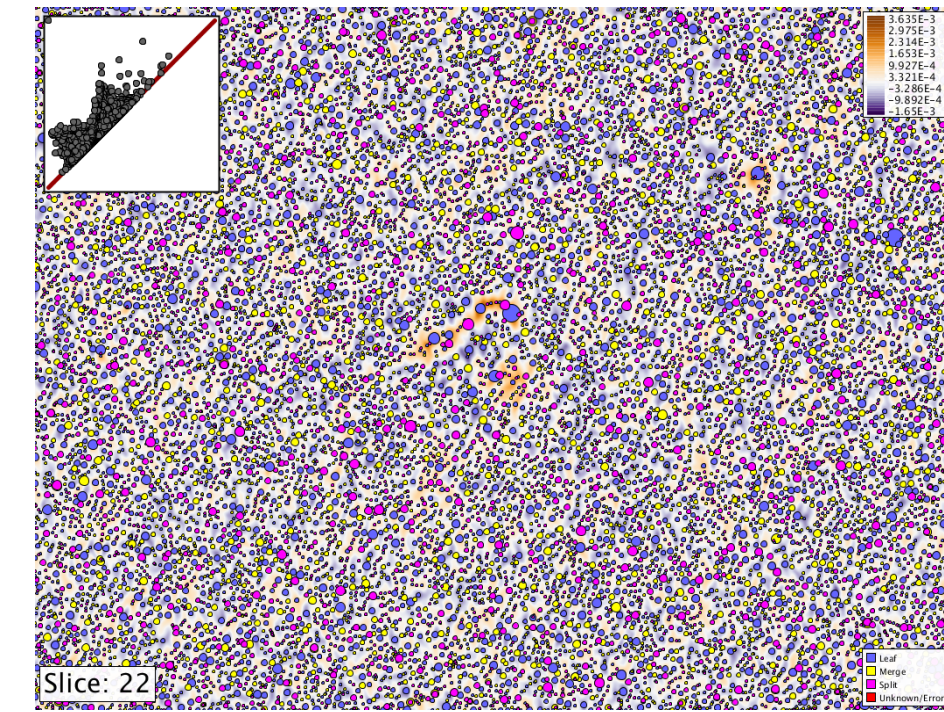
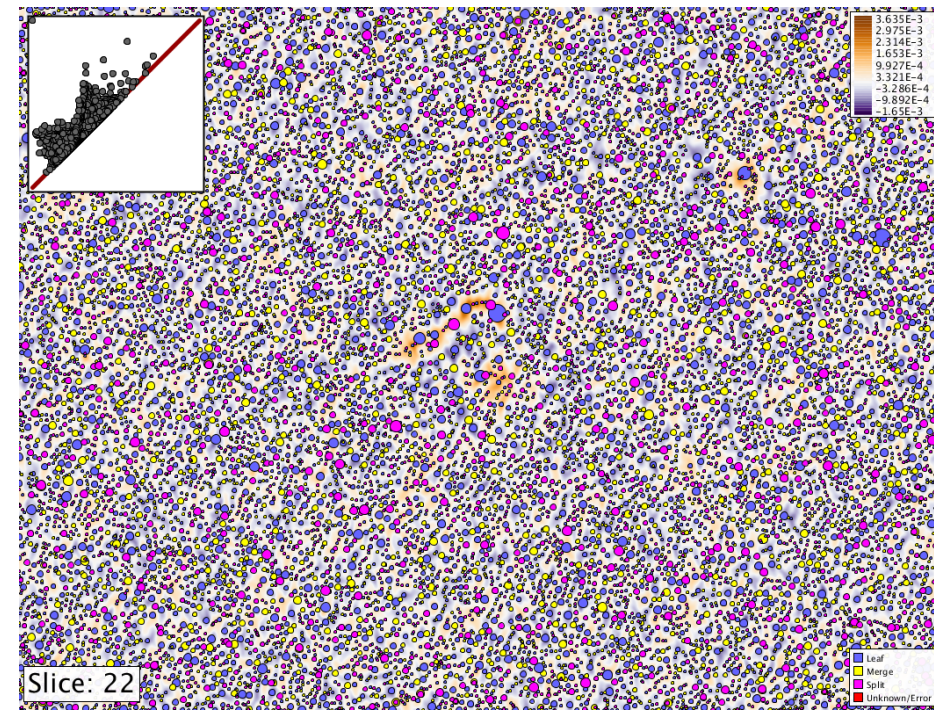


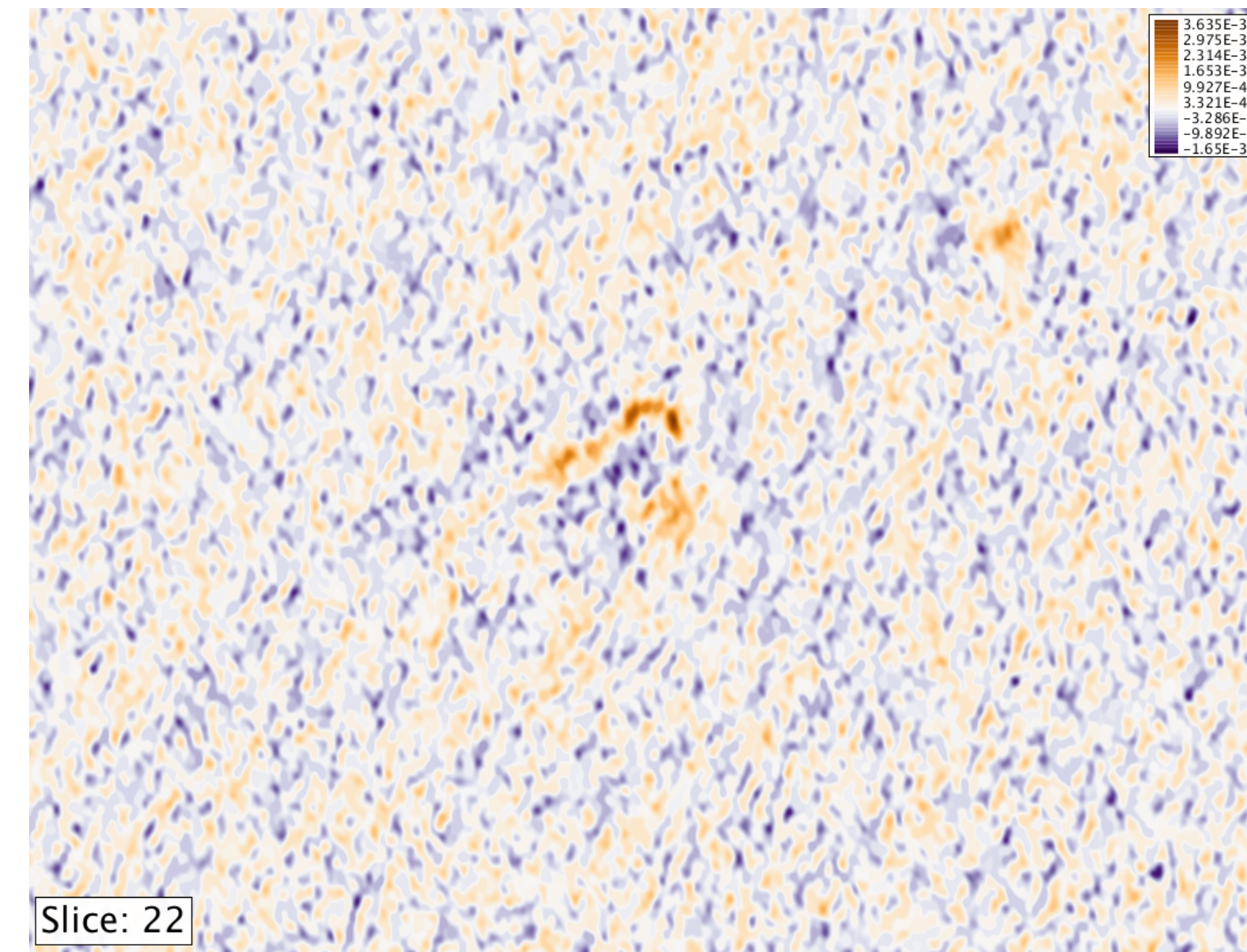
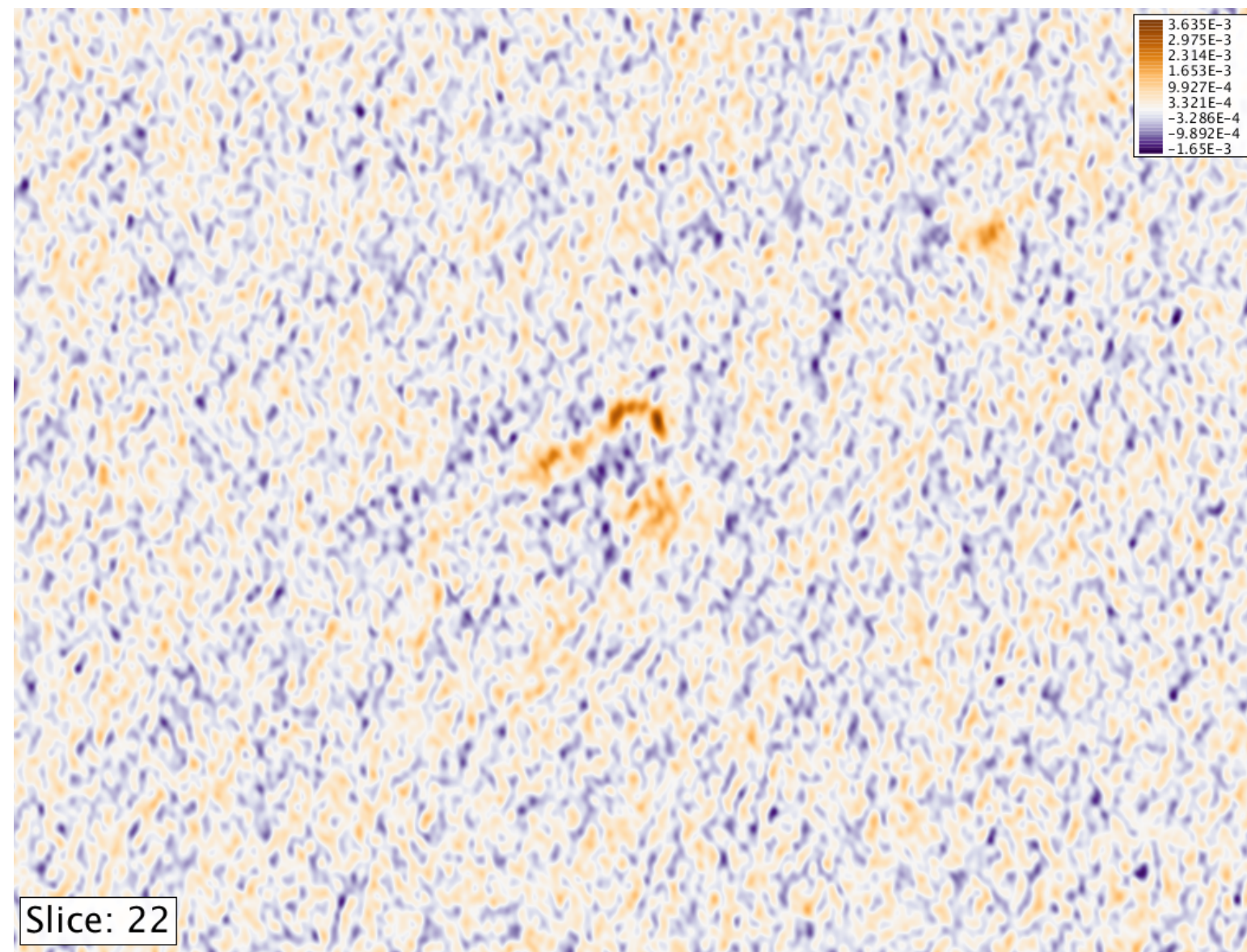
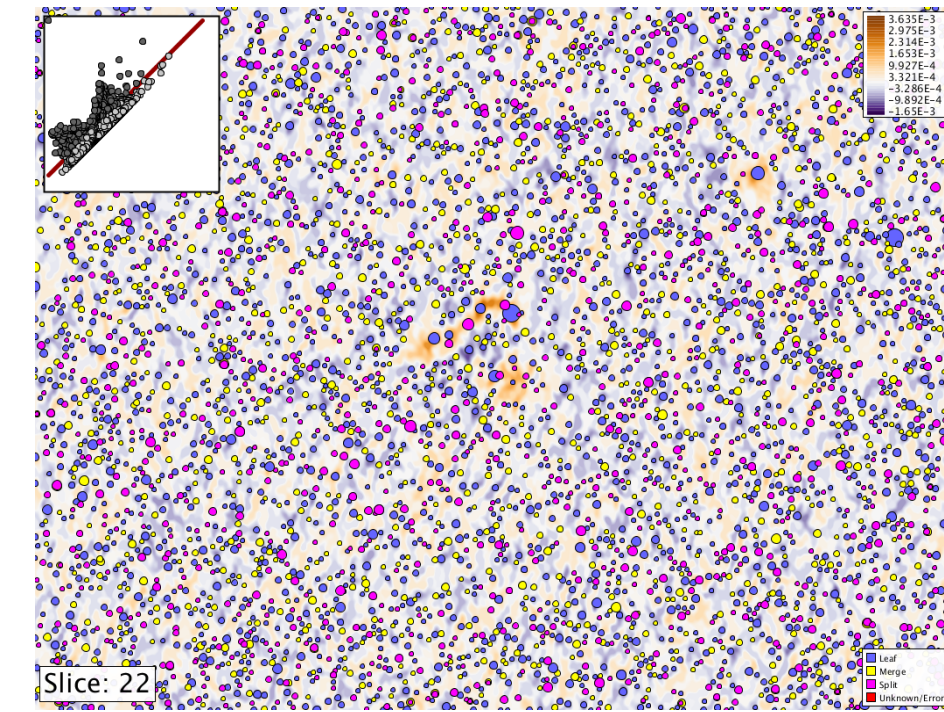
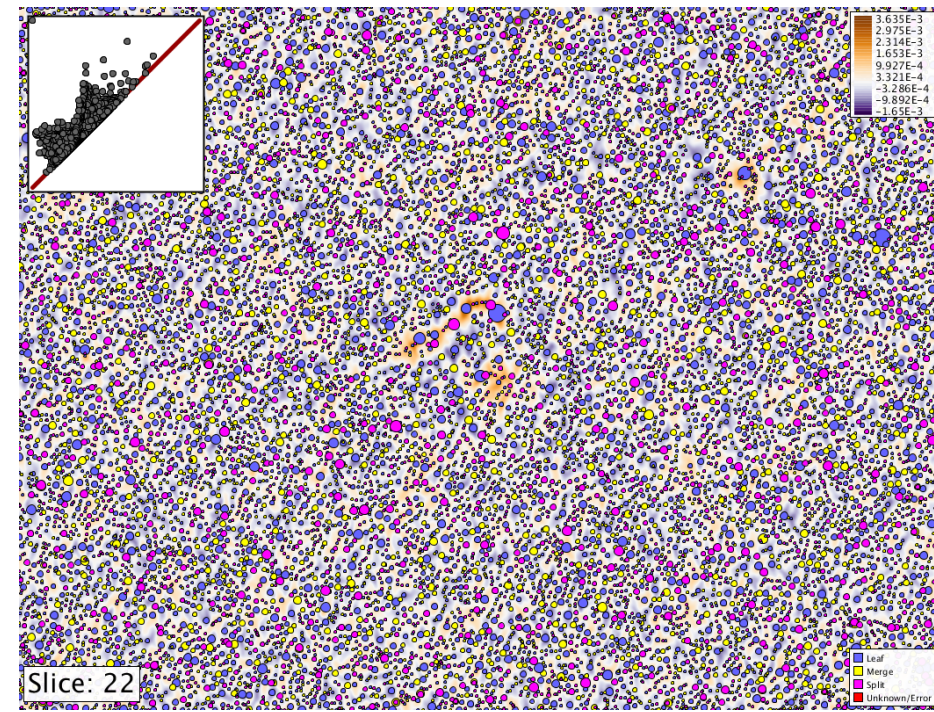


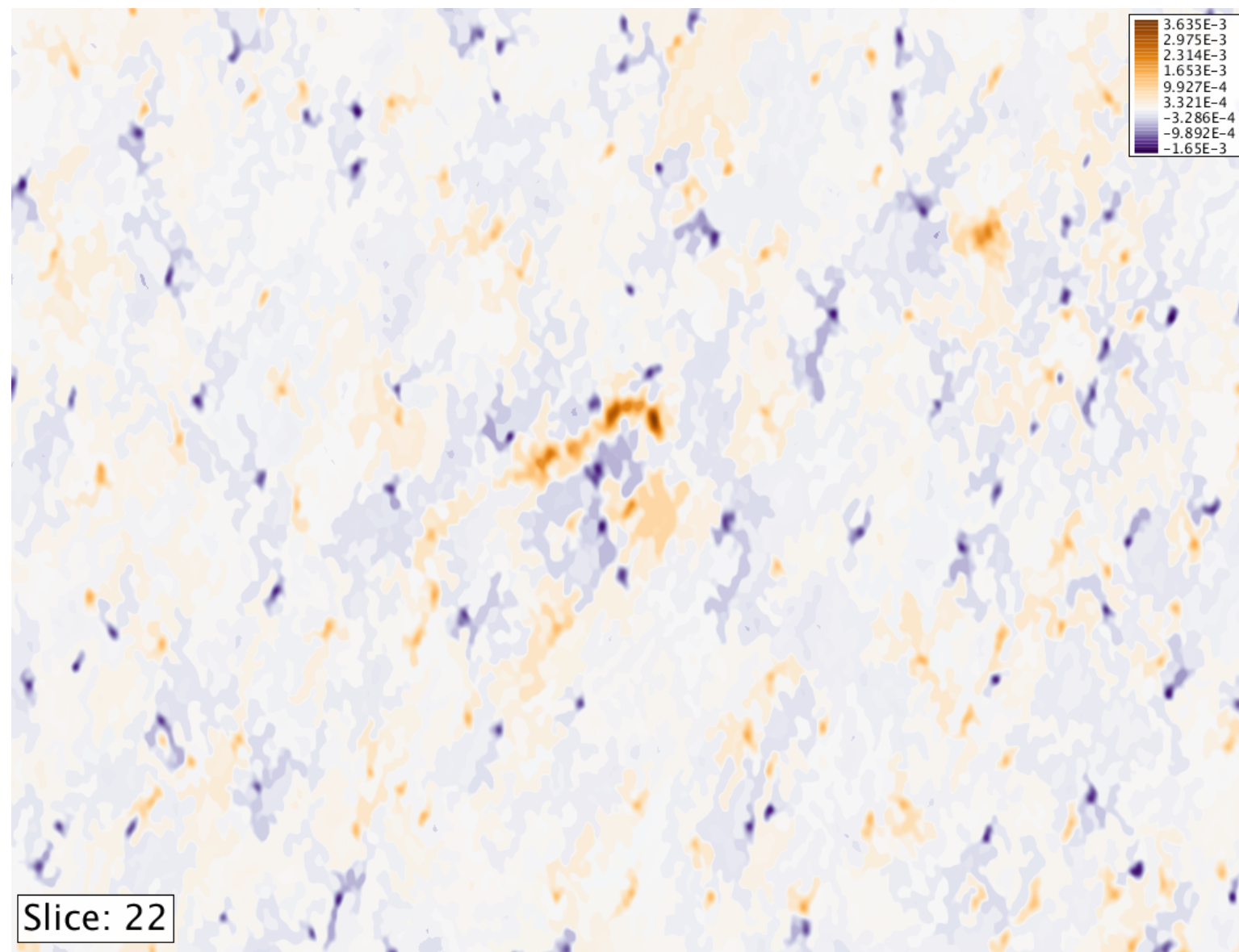
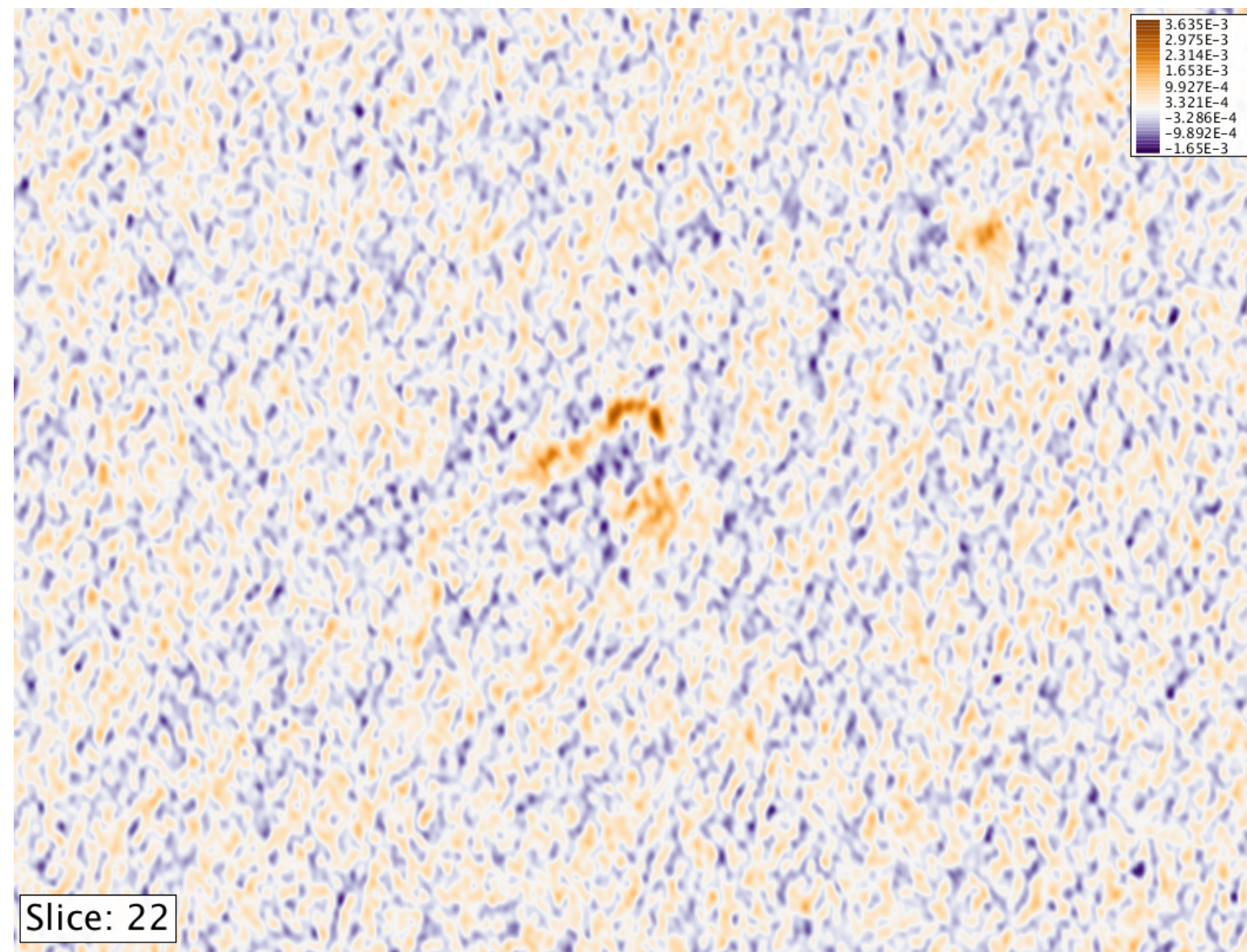
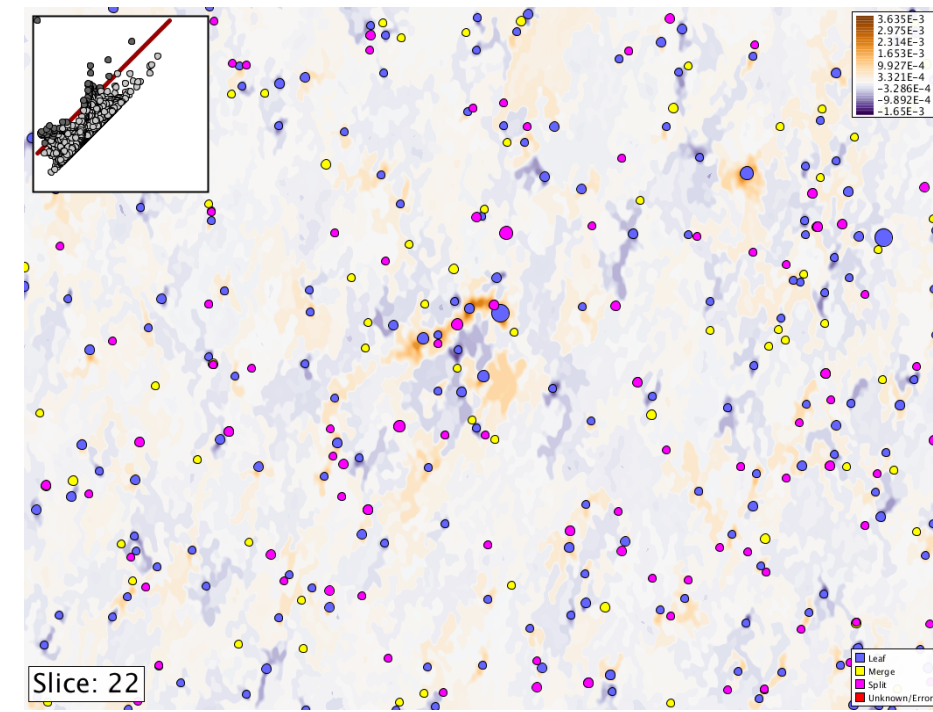
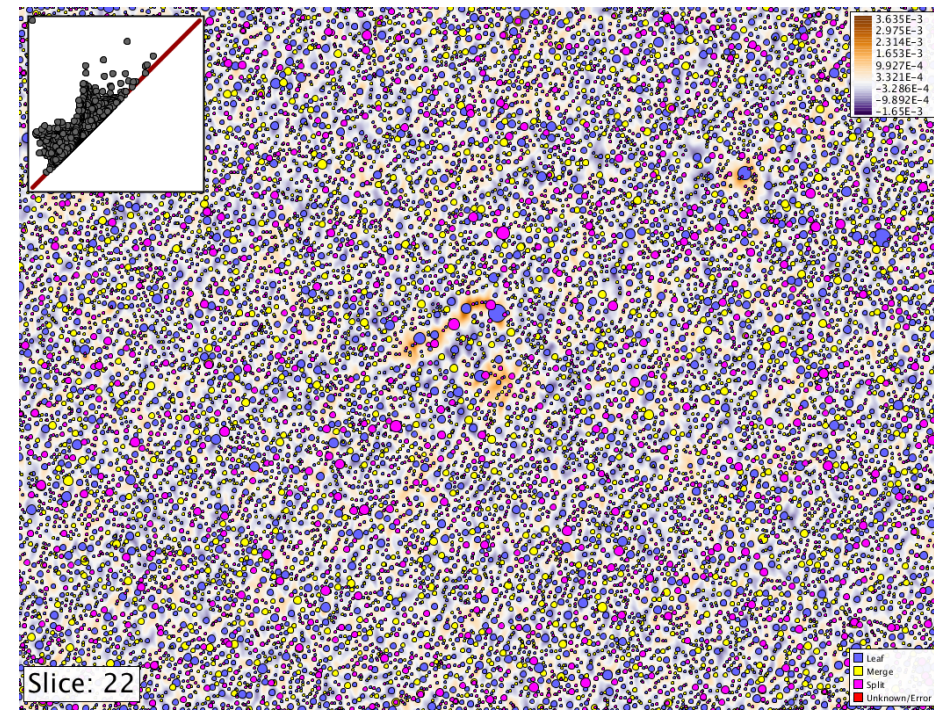
Paul Rosen, Bei Wang, Anil Seth, Betsy Mills, Adam Ginsburg, Julia Kamenetzky, Jeff Kern, Chris R. Johnson.
Manuscript, 2017.

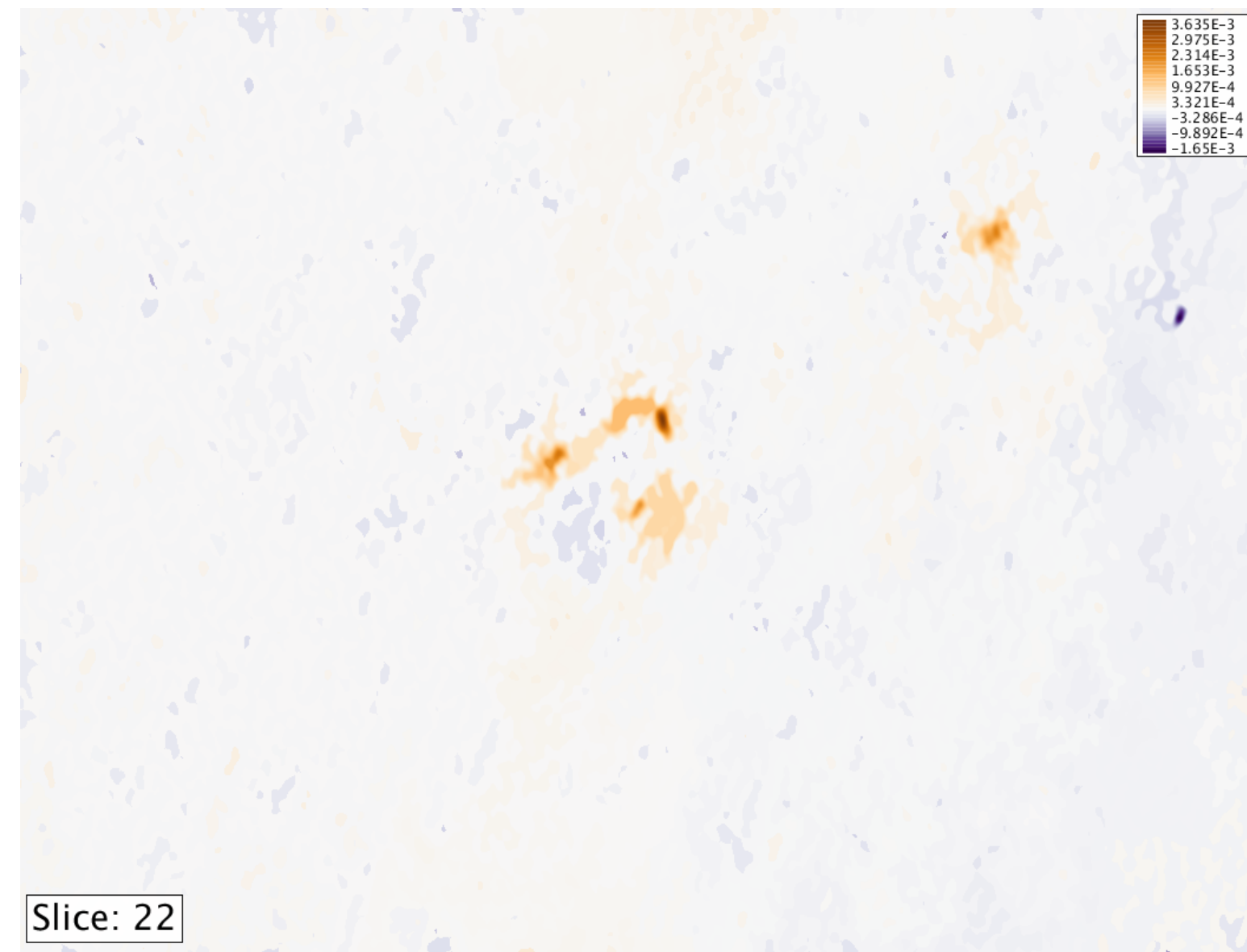
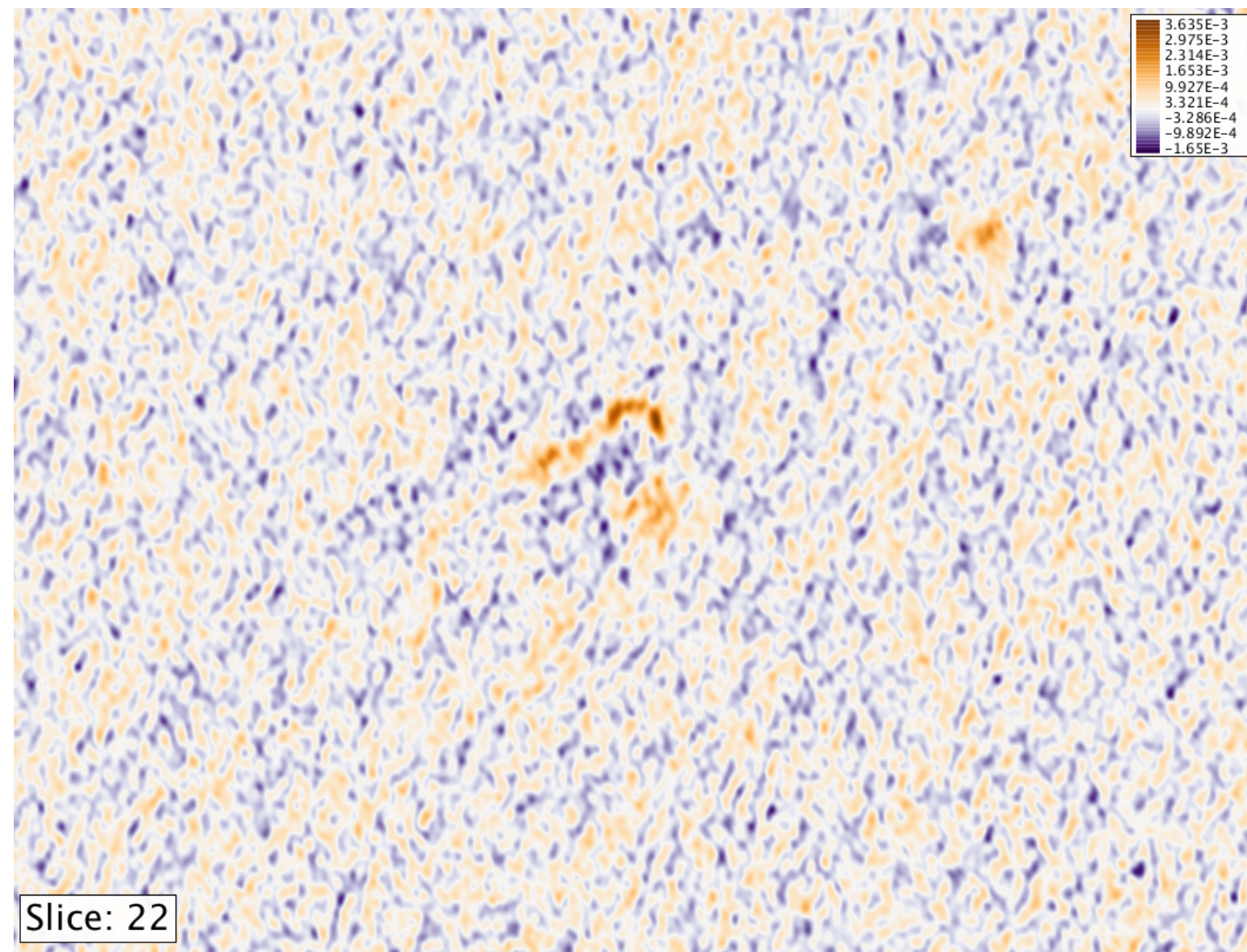
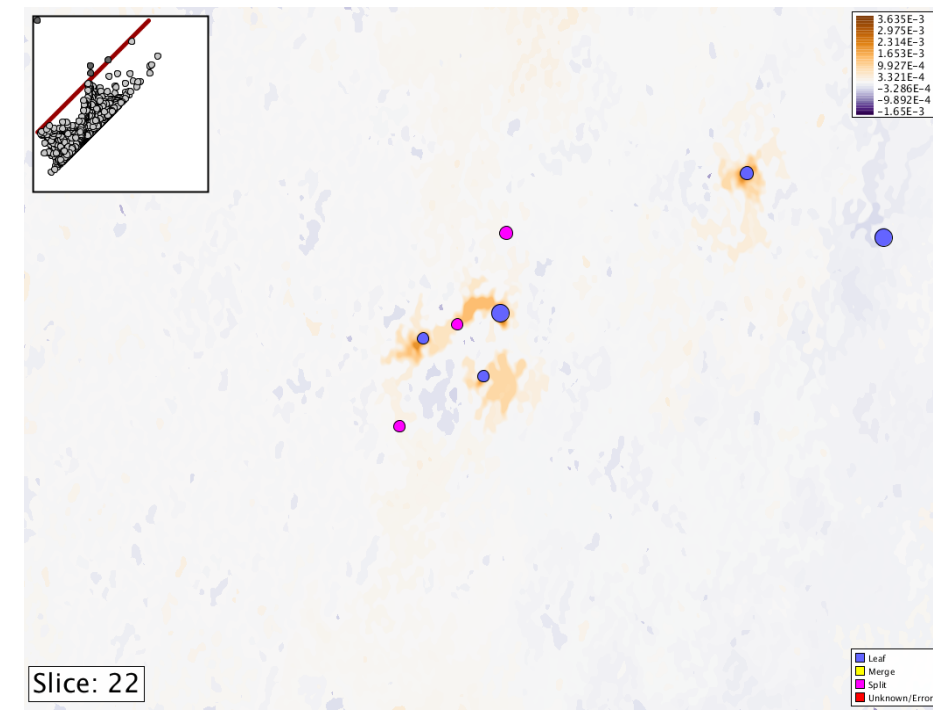
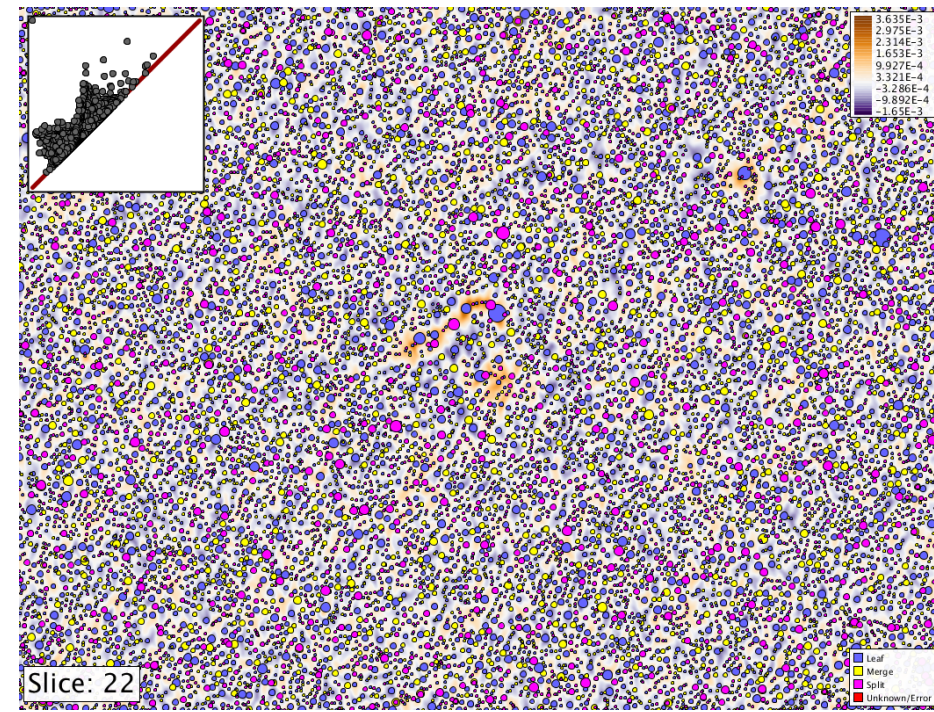


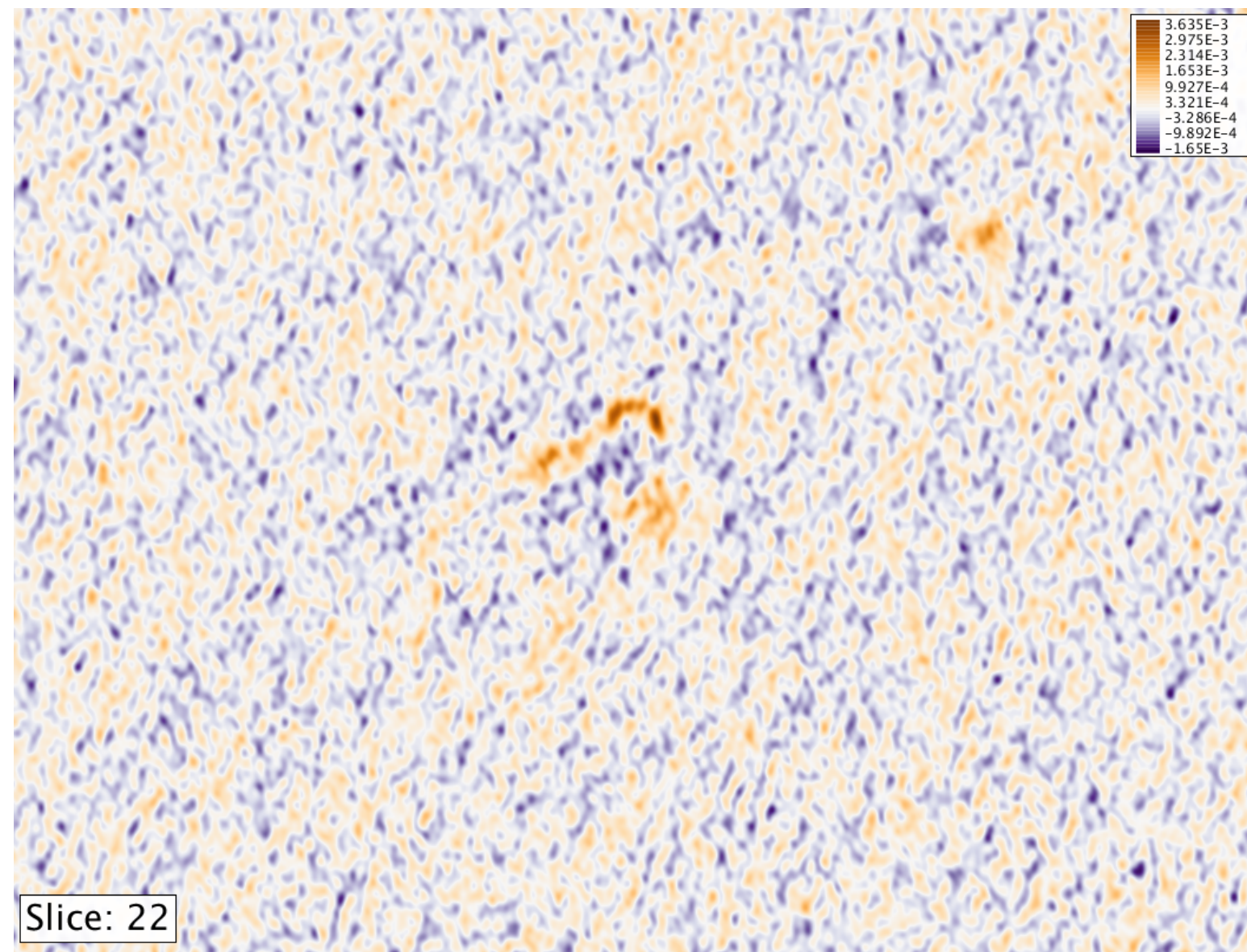
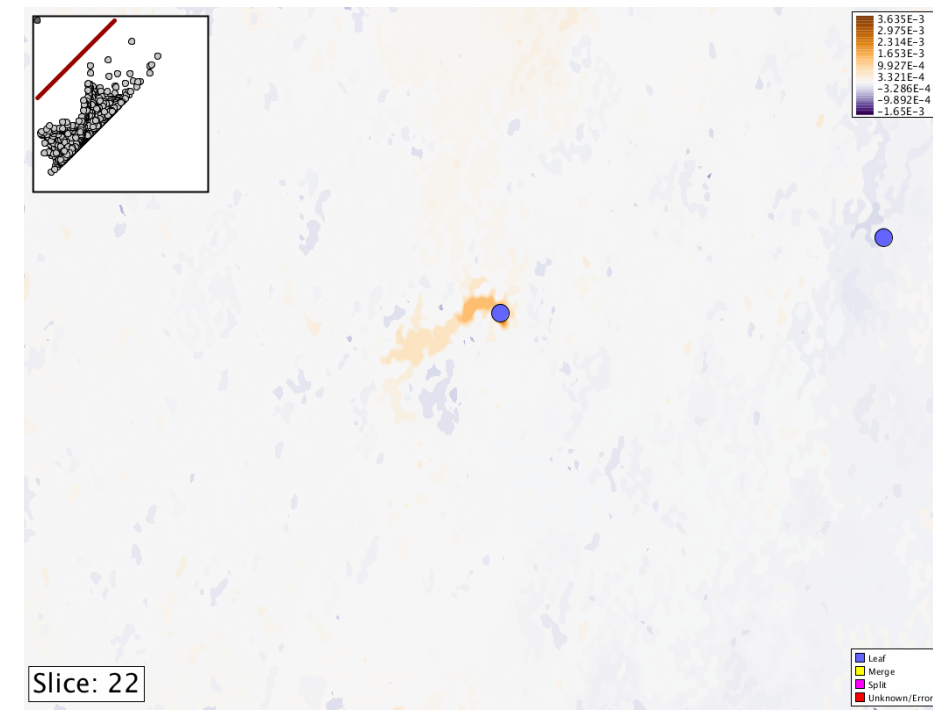
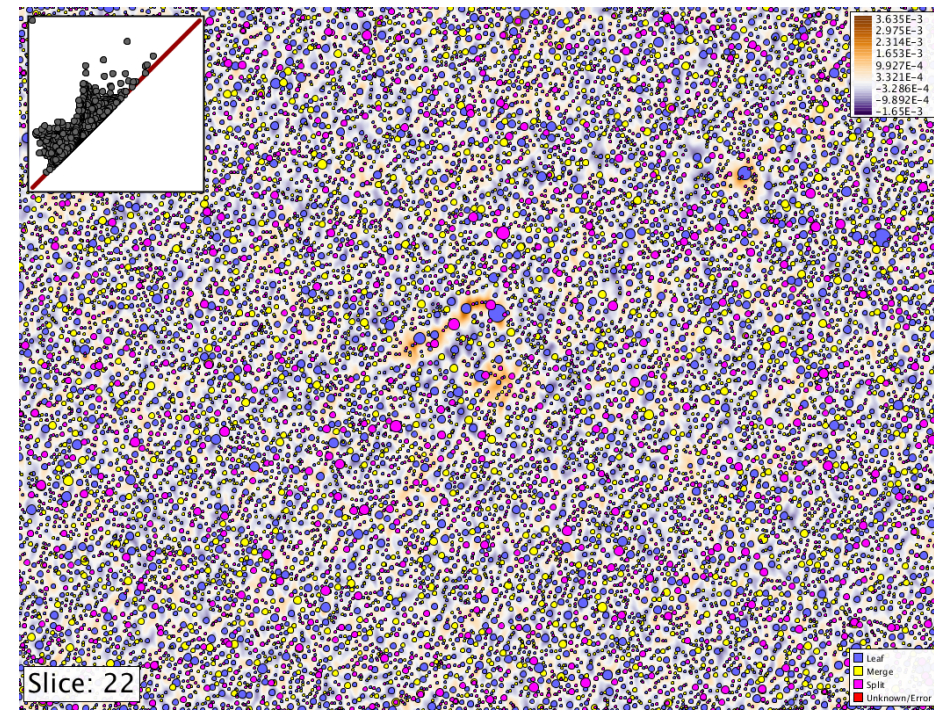
Denoising at Multi-scale and Source Finding



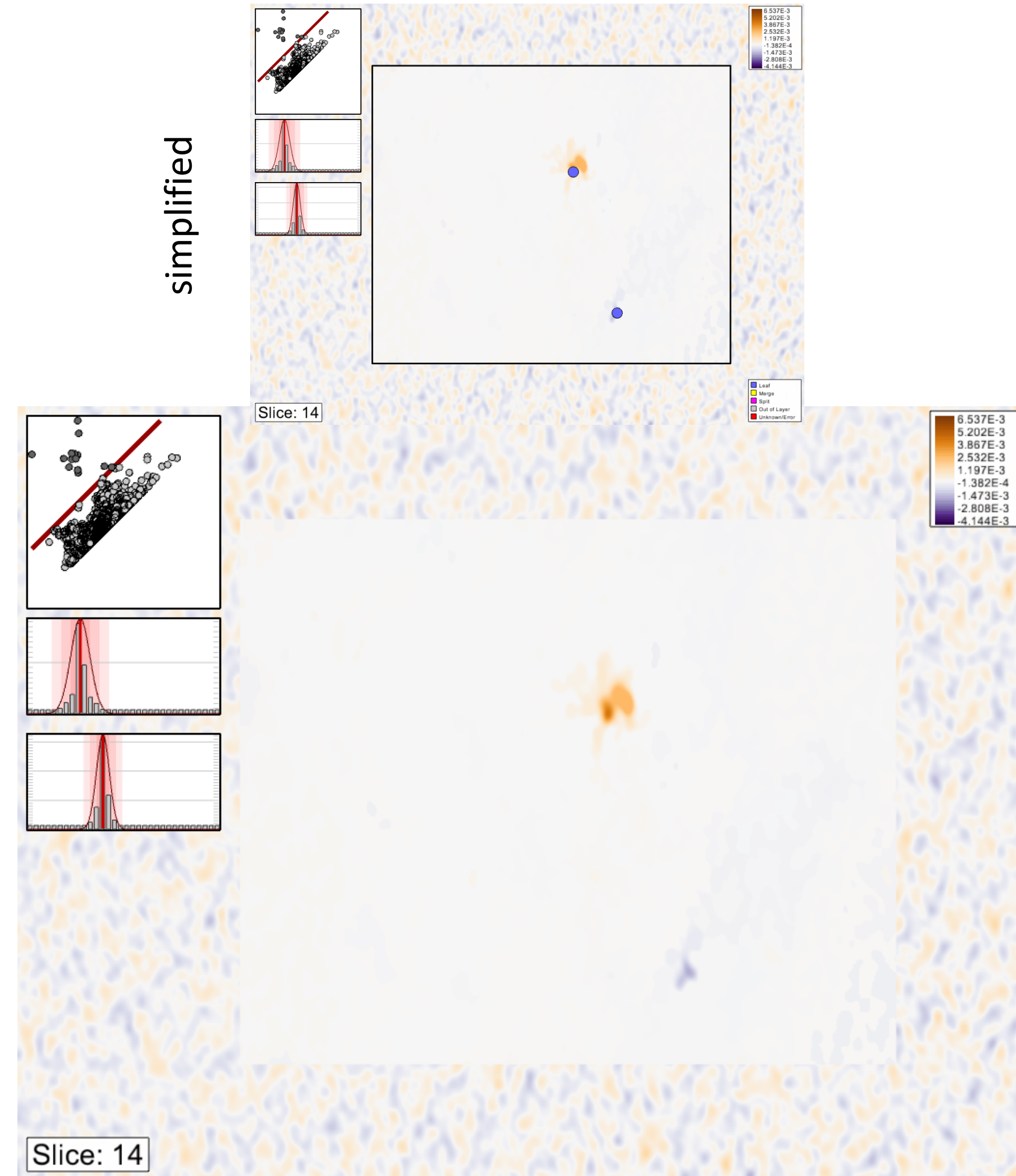
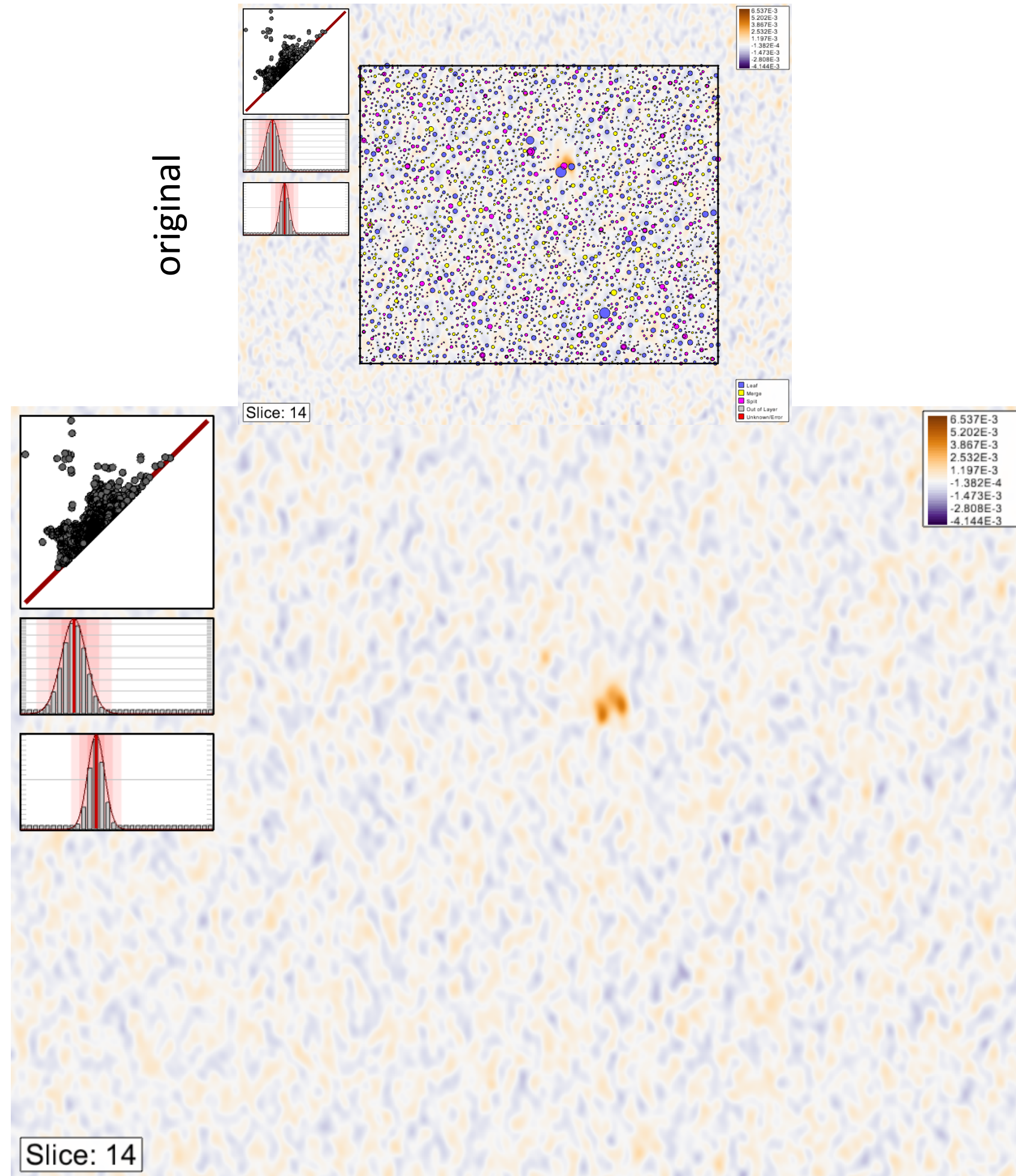


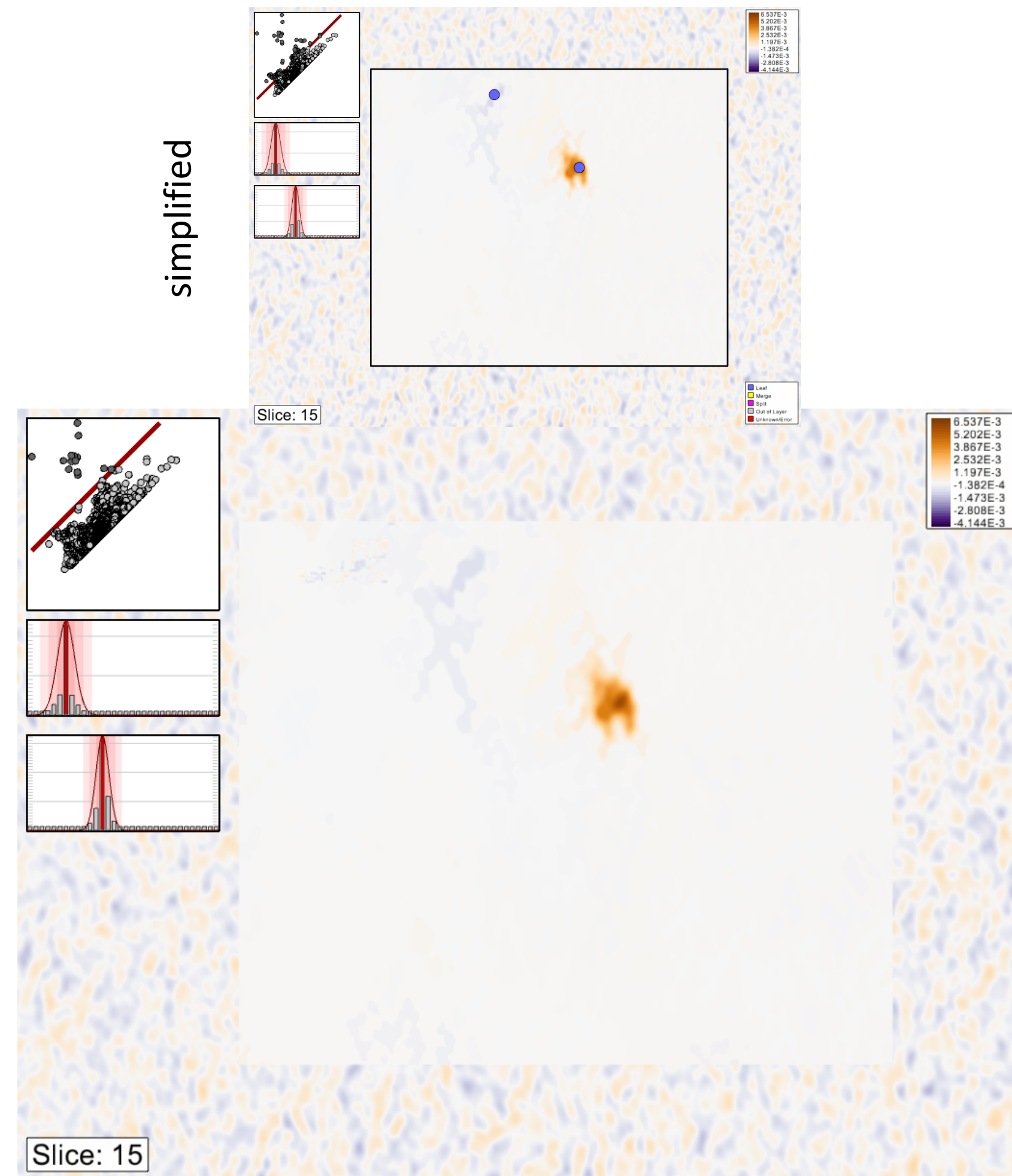
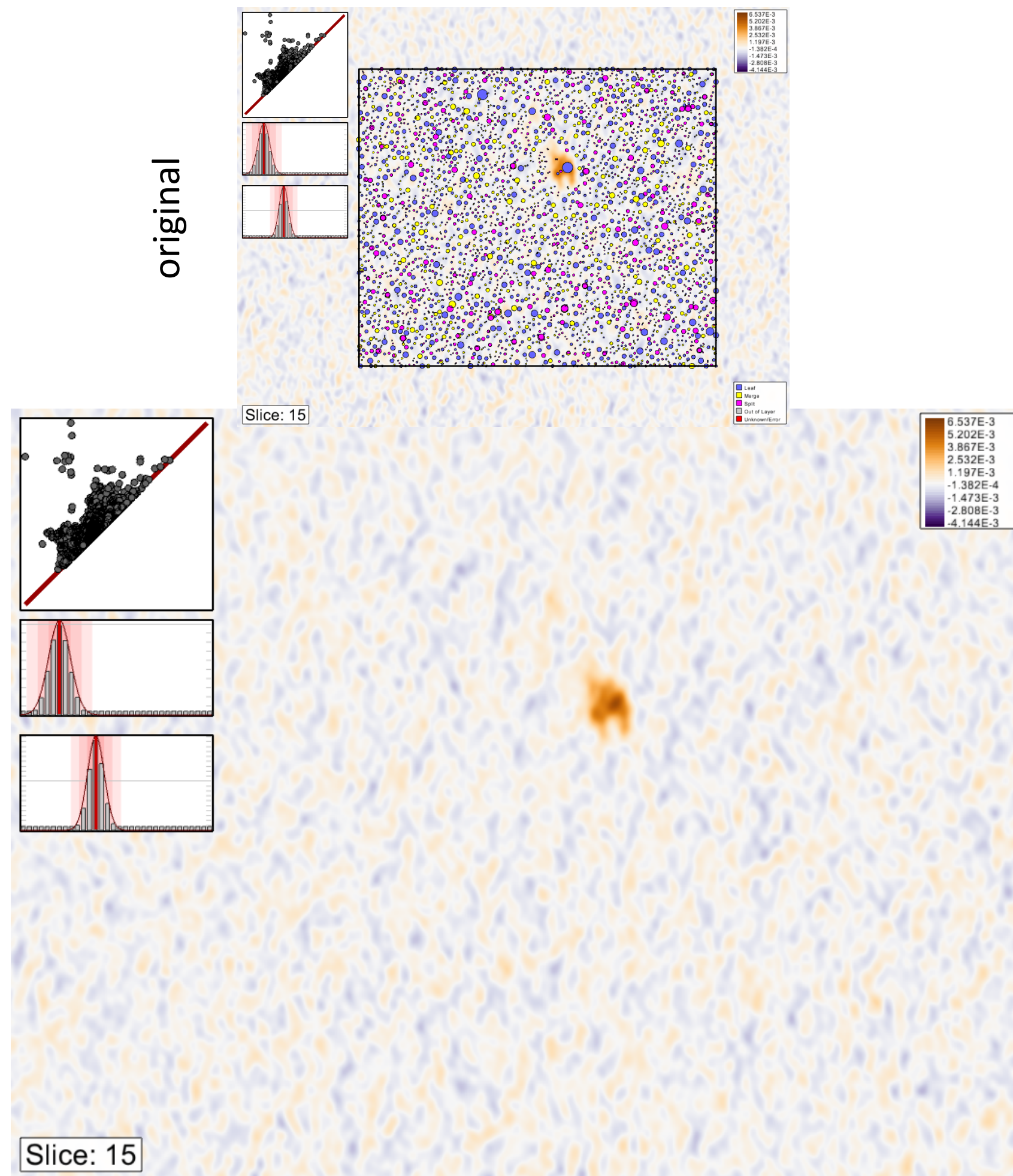


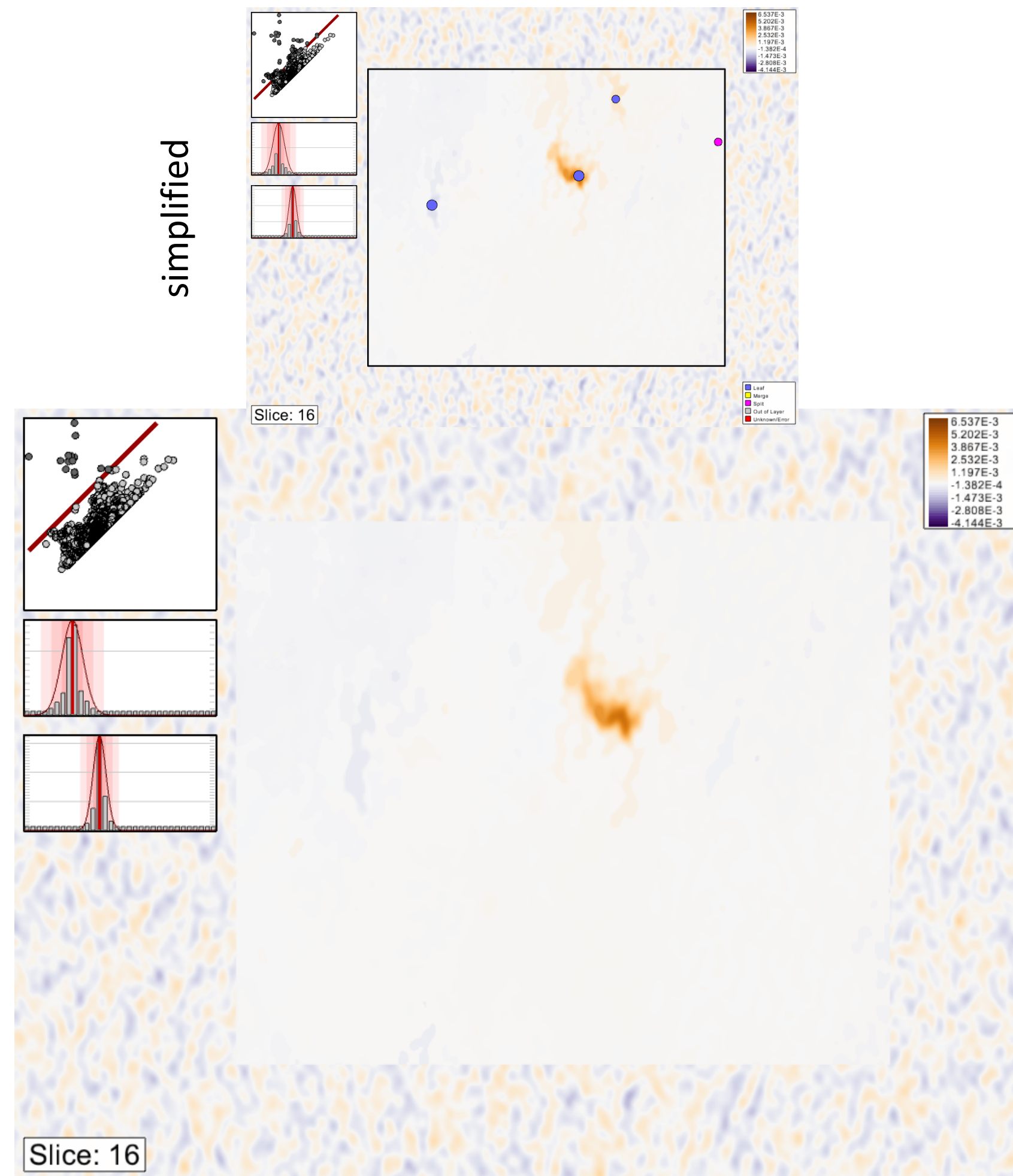
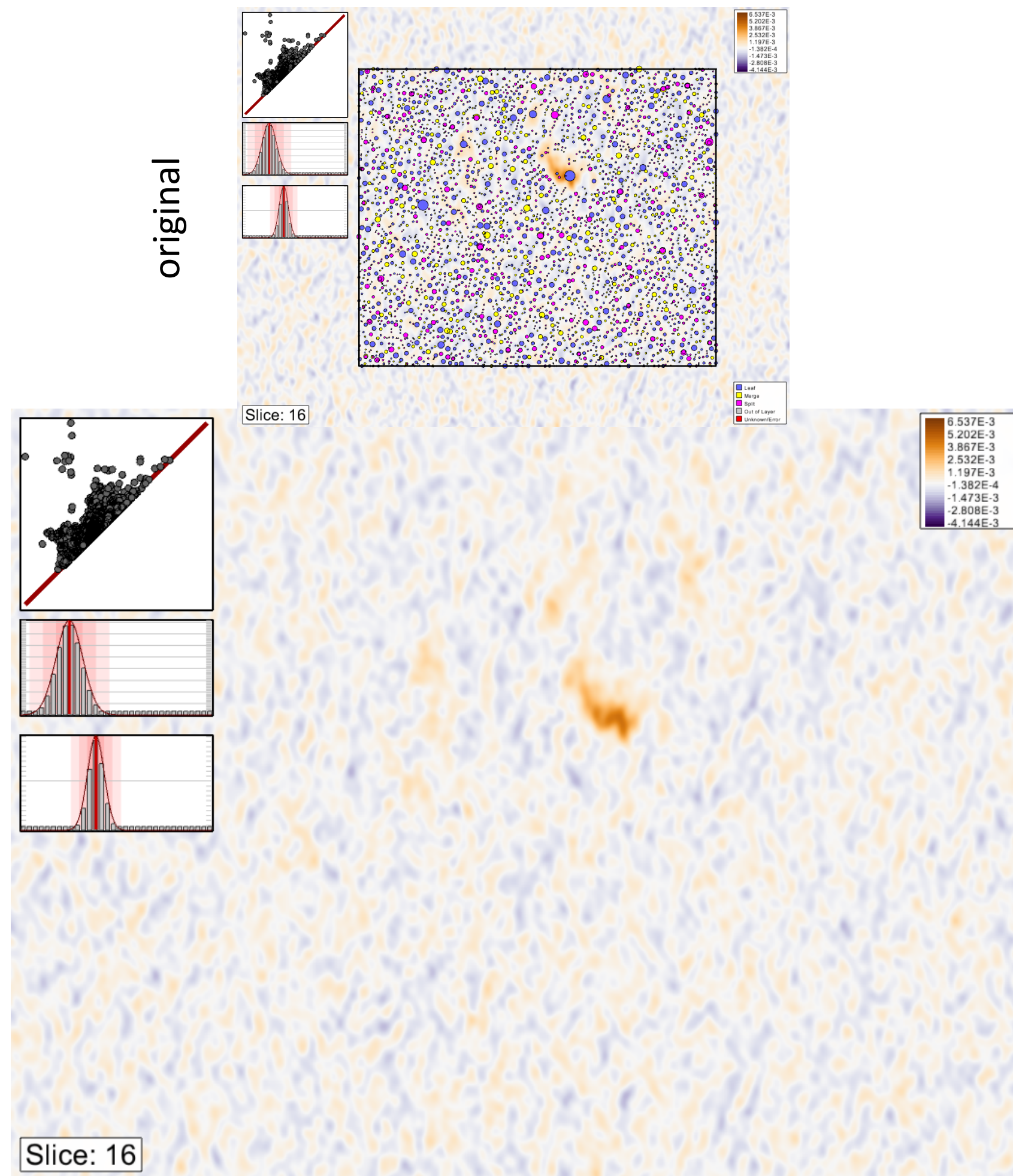


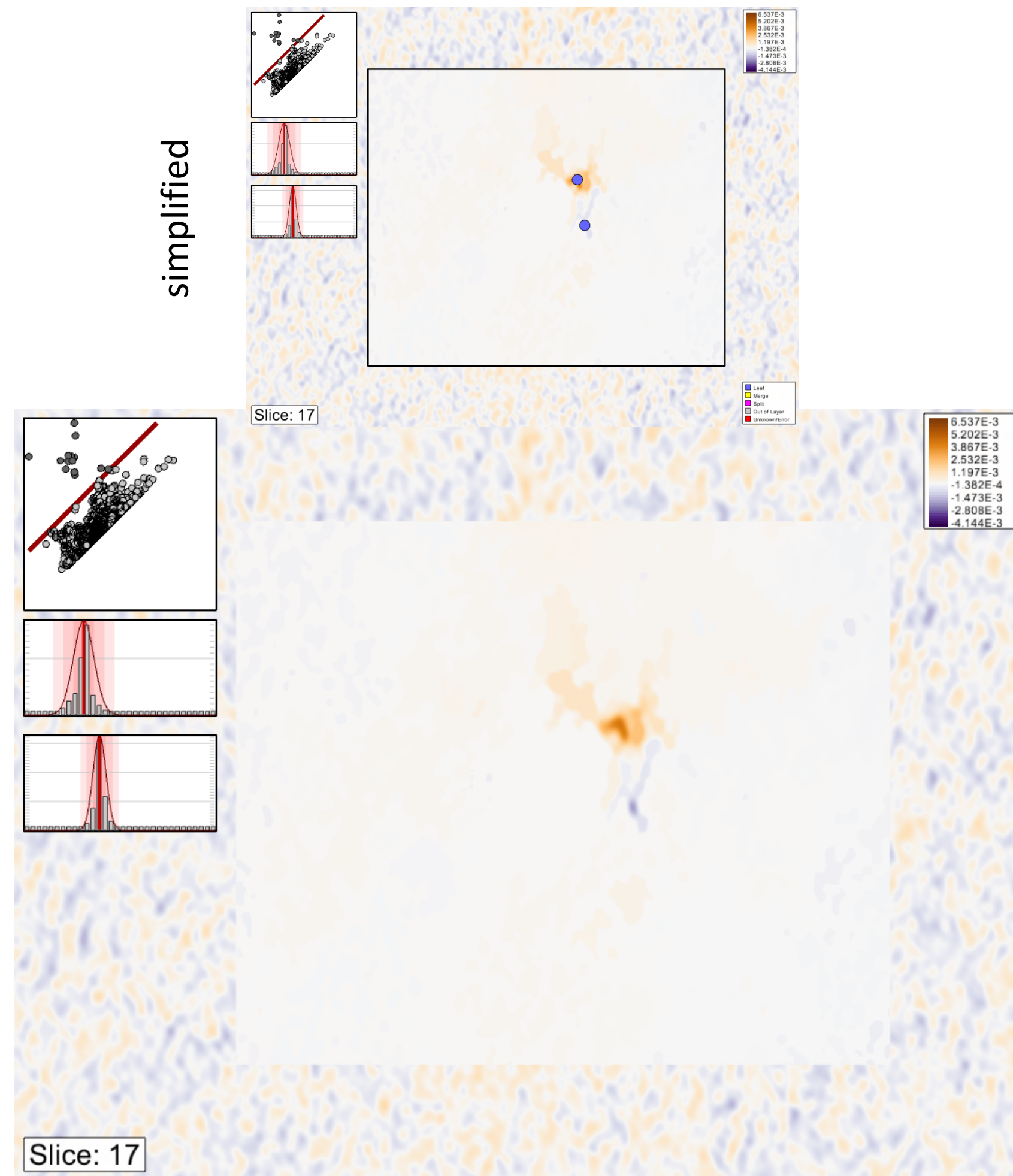
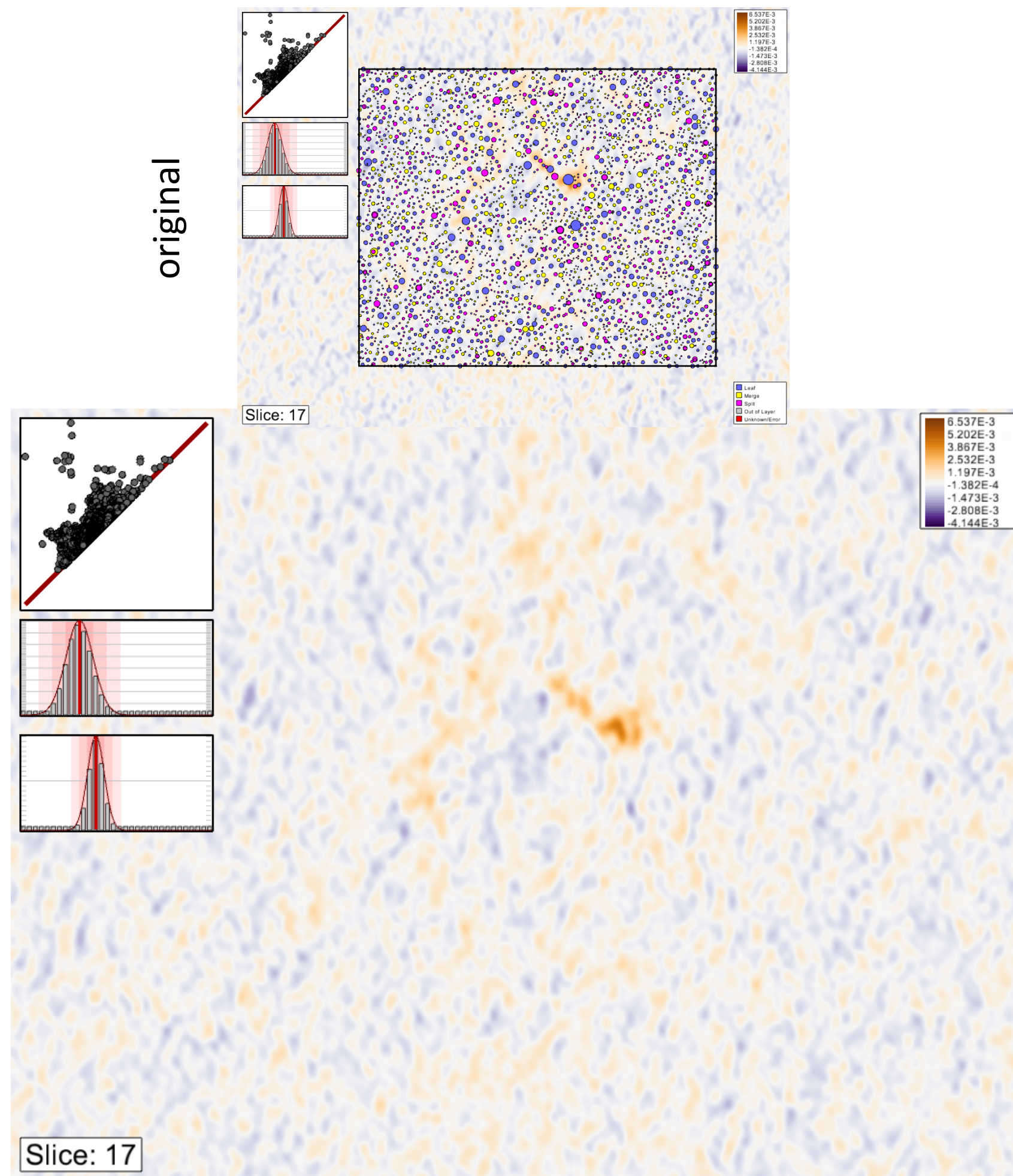


Stepping Through Slices

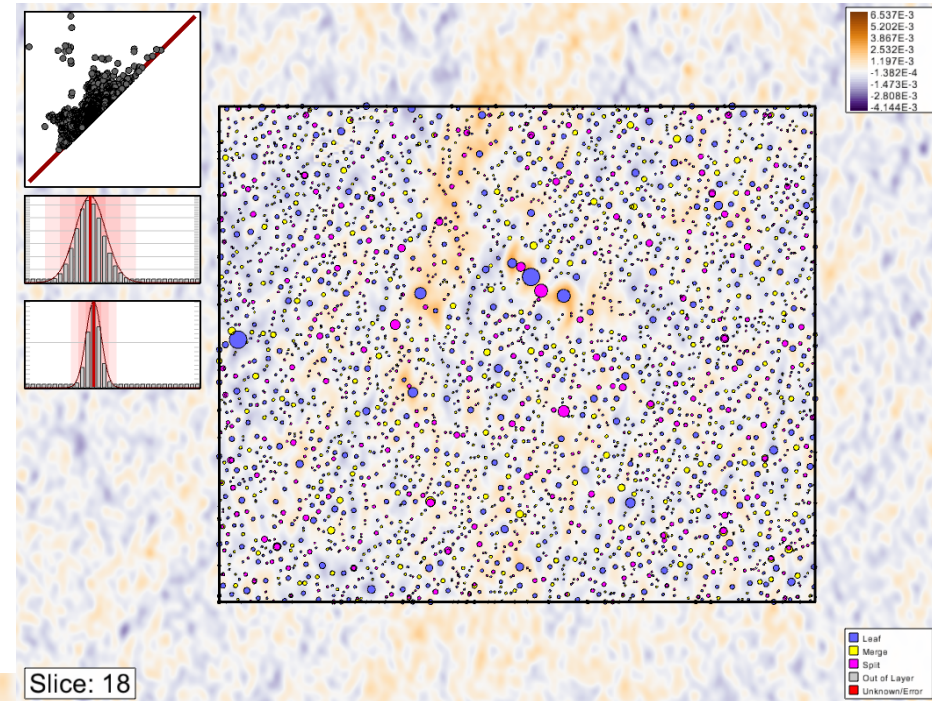




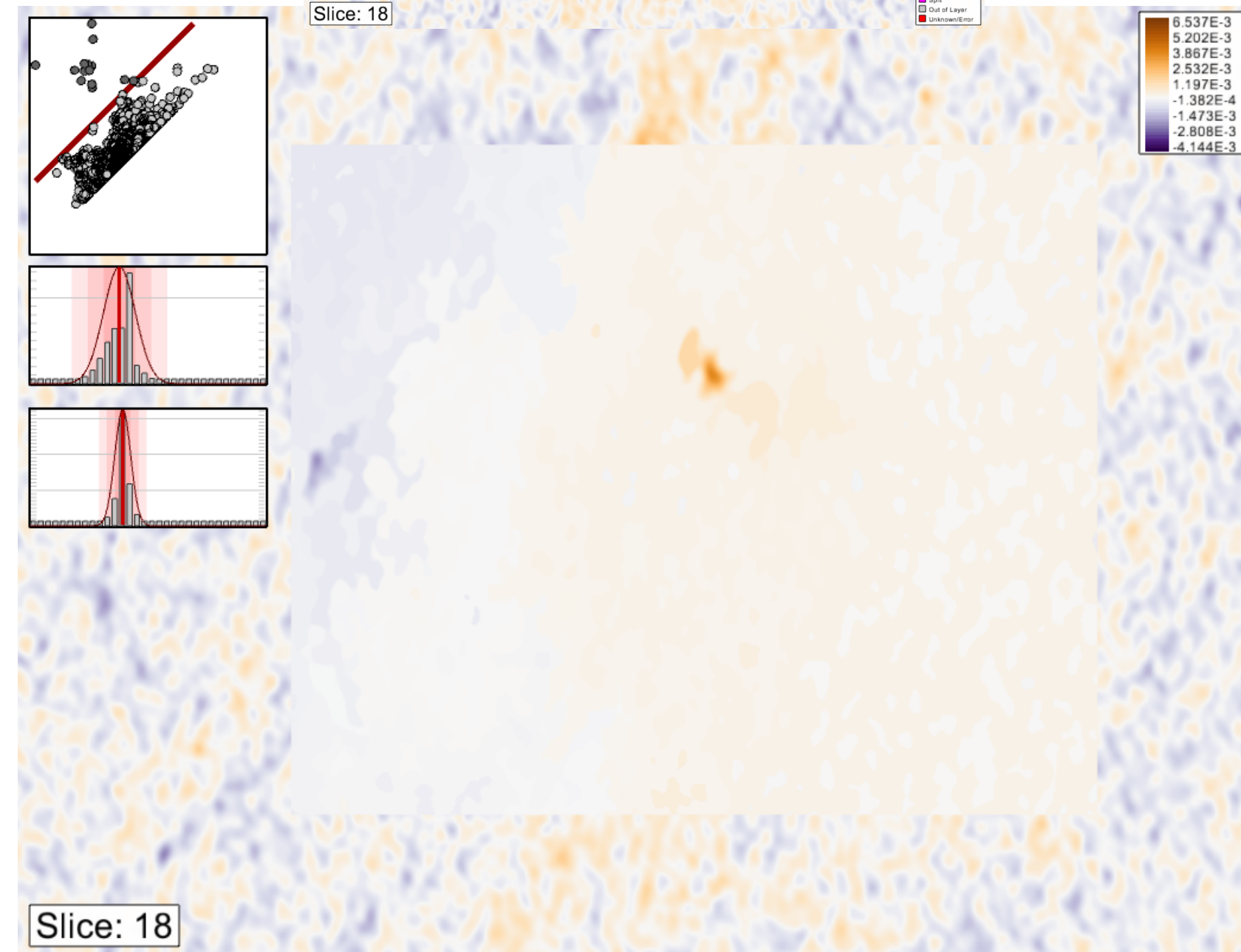
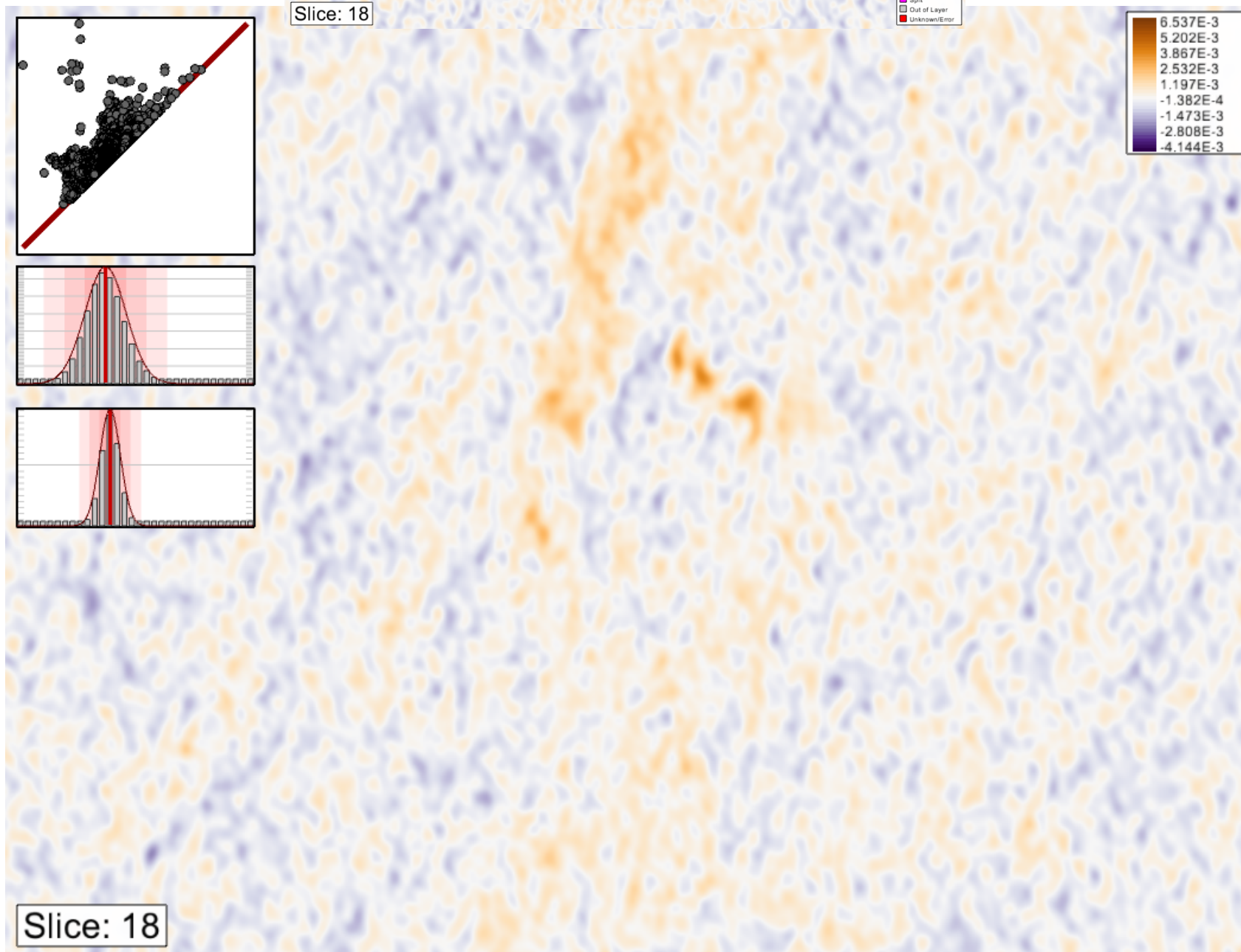
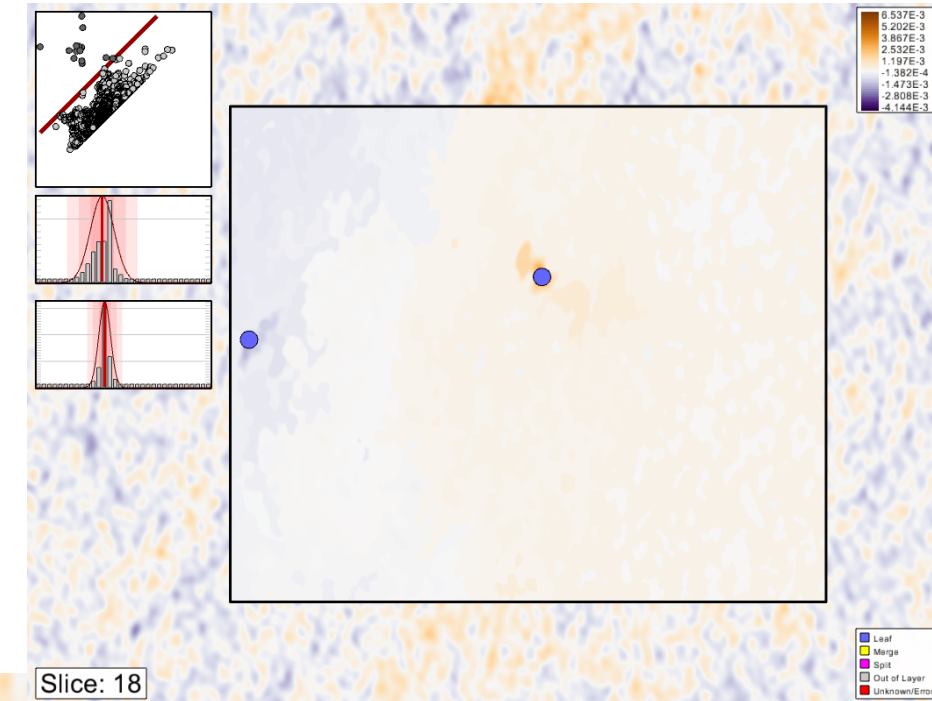




original

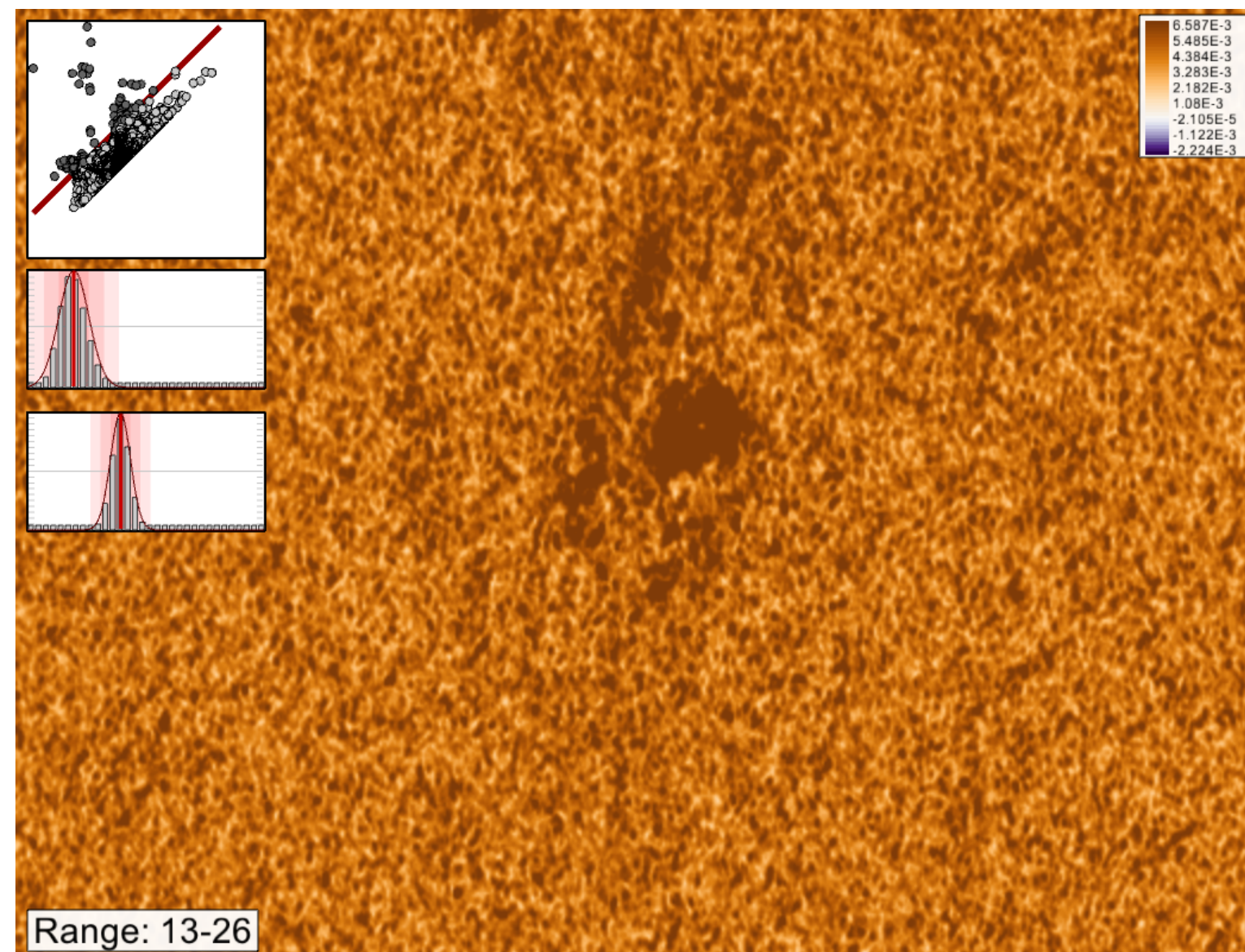


simplified

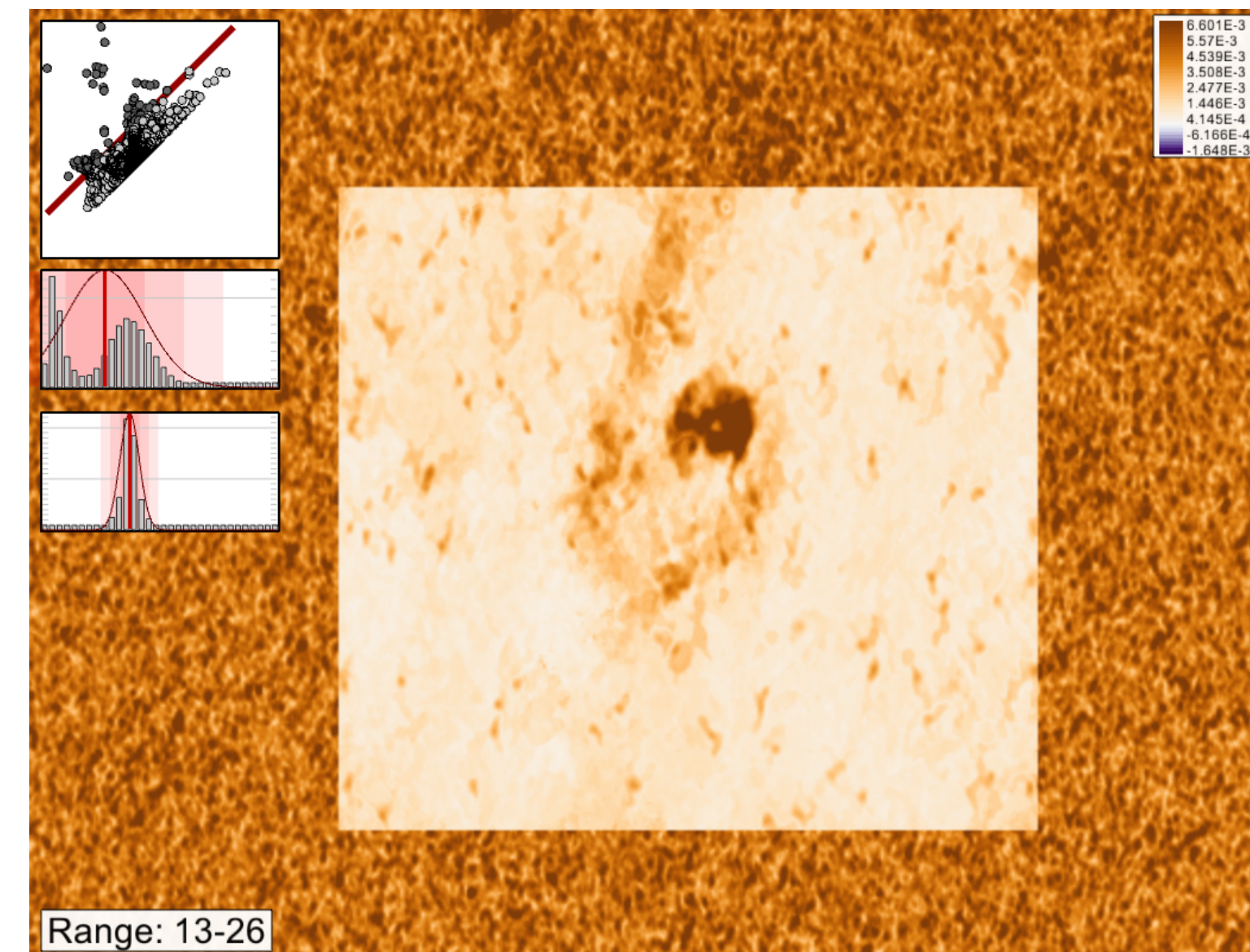


MOMENT 0 ANALYSIS

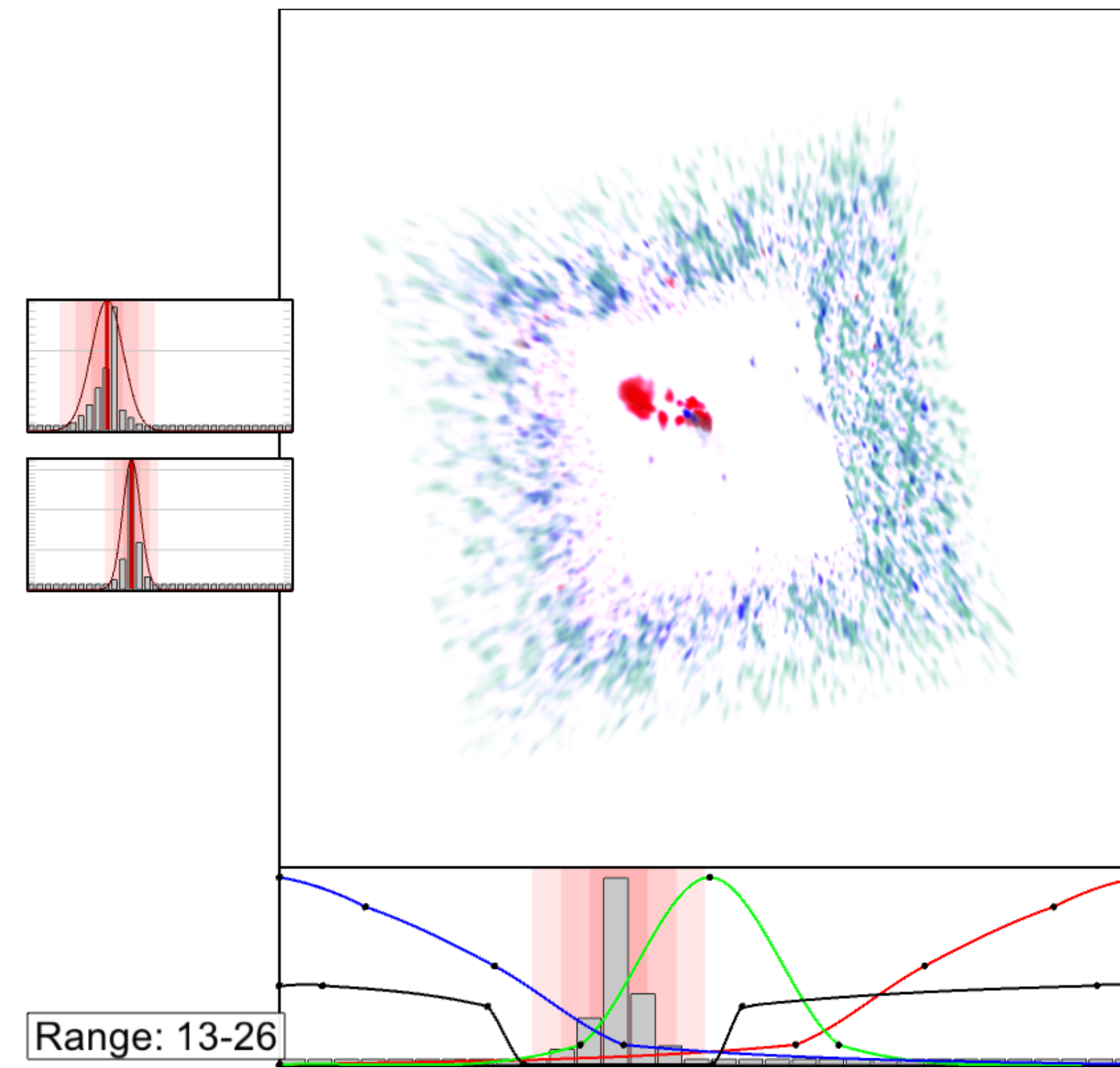
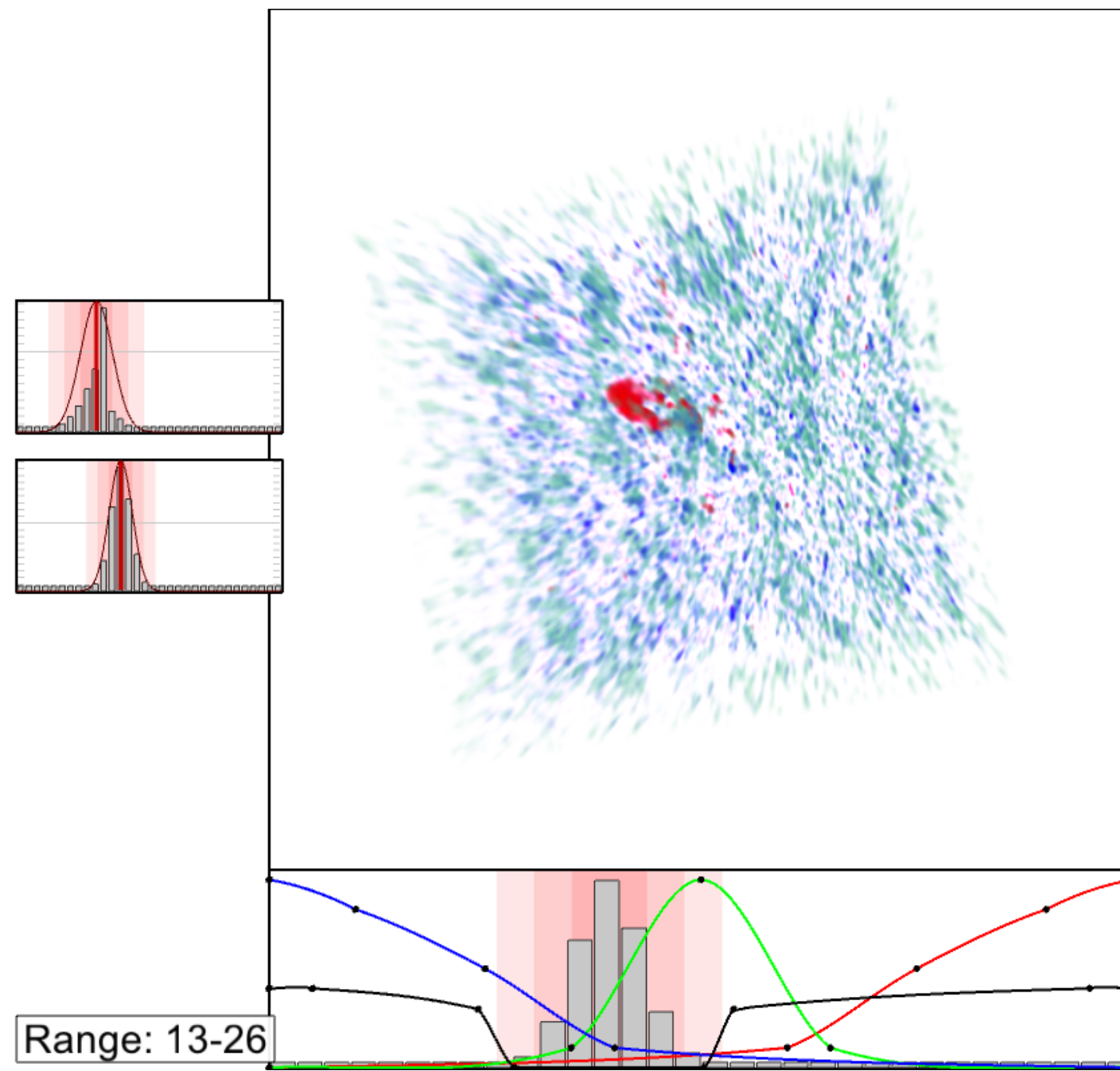
original



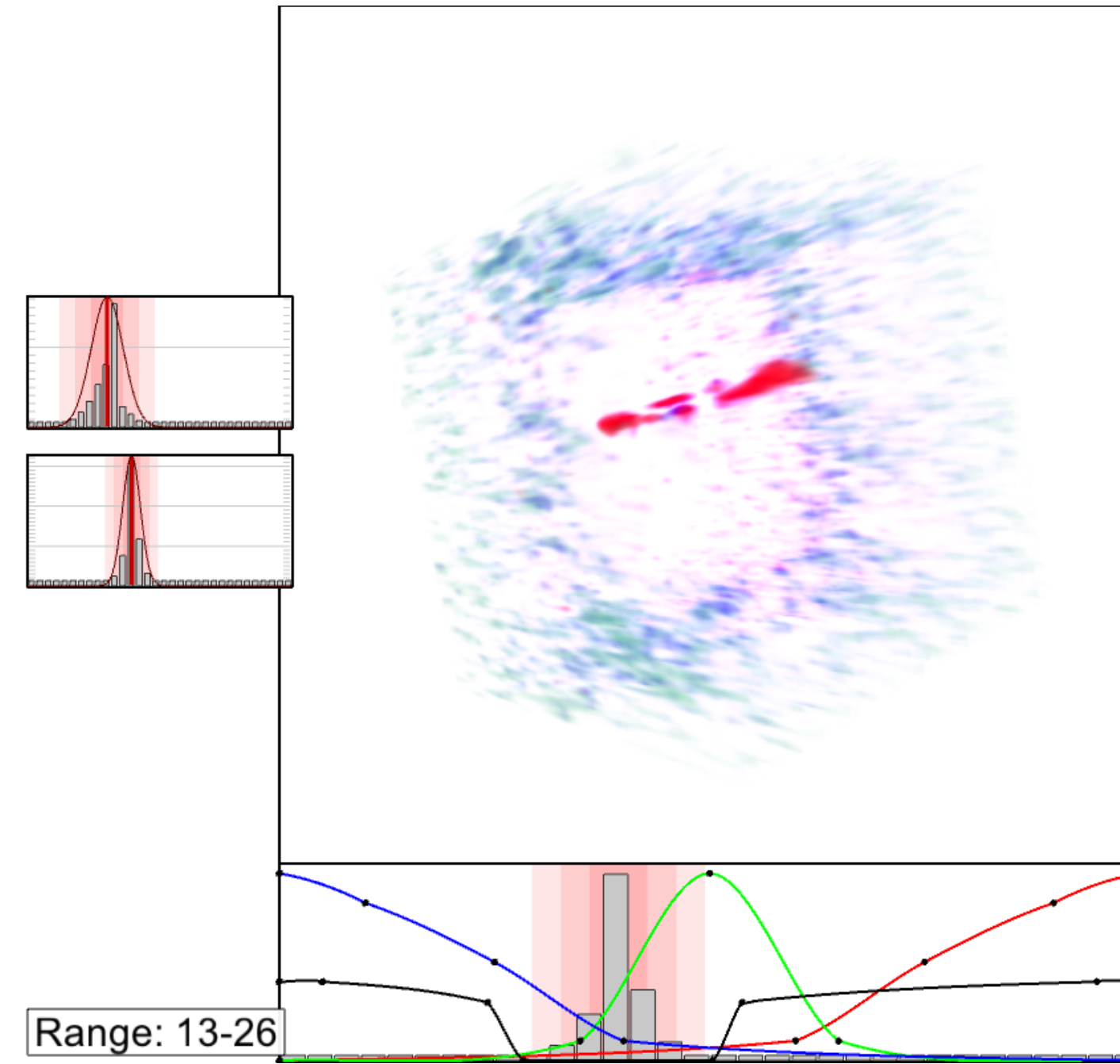
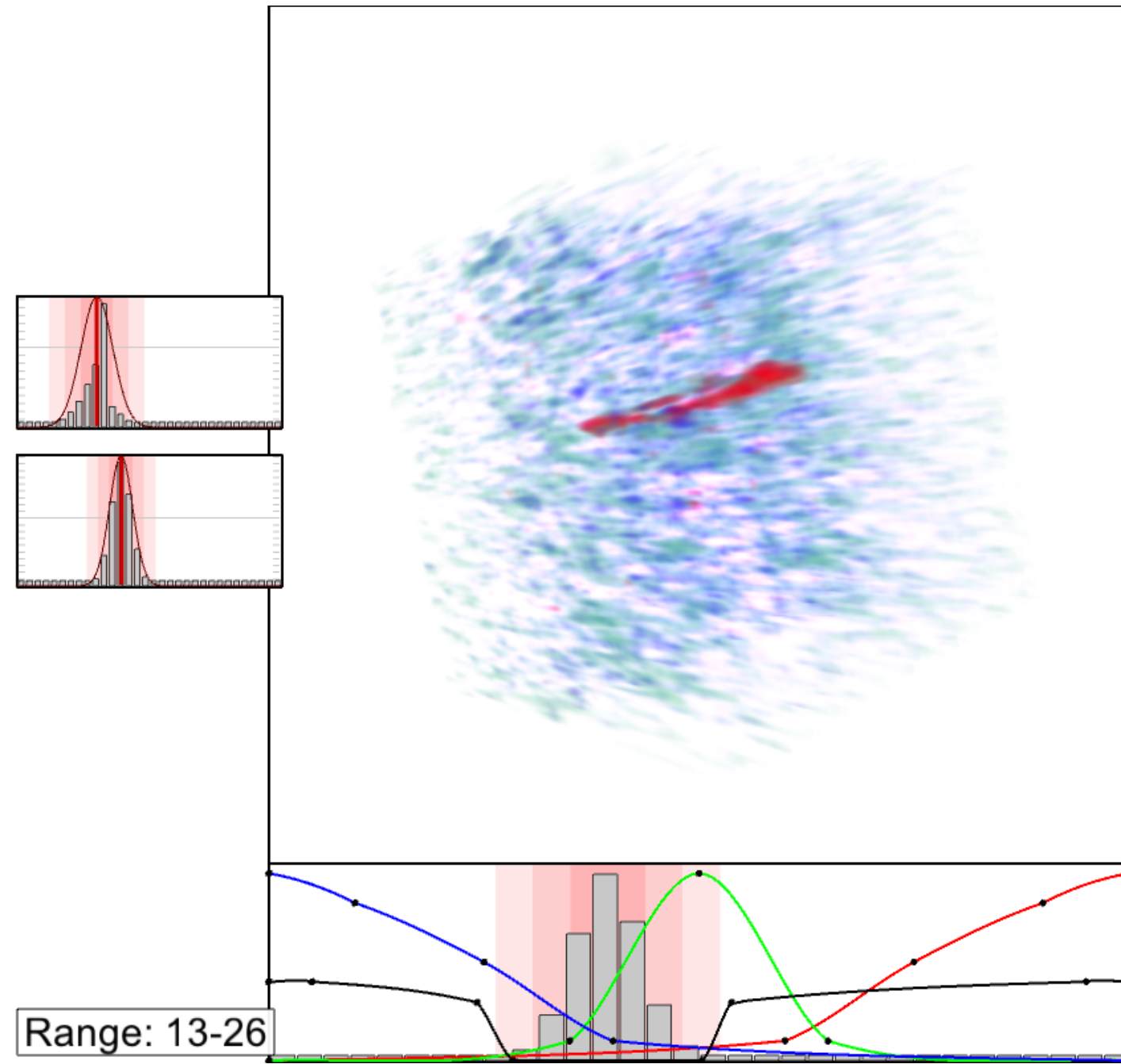
simplified



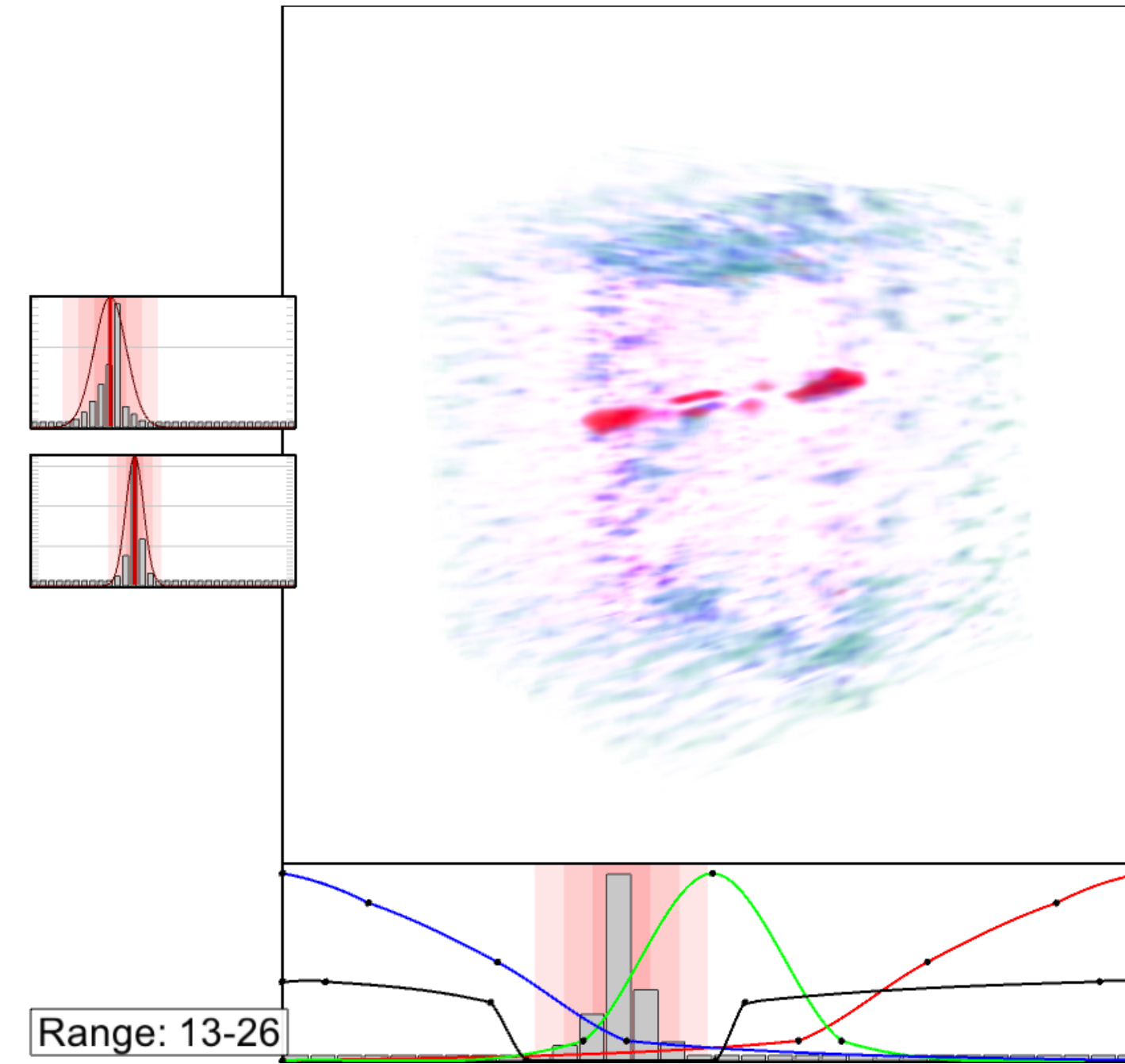
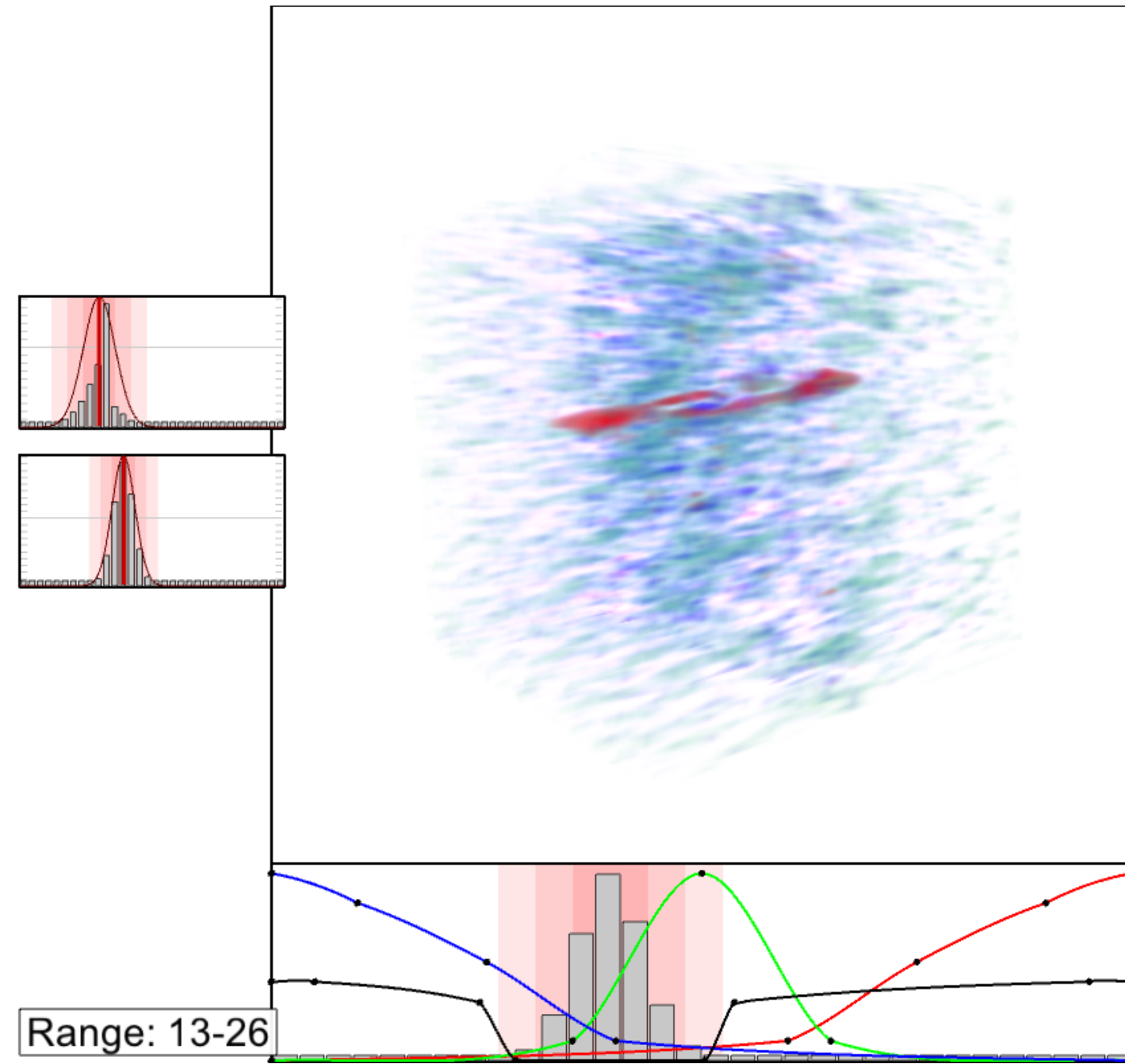
Observing the red shift



Observing the red shift



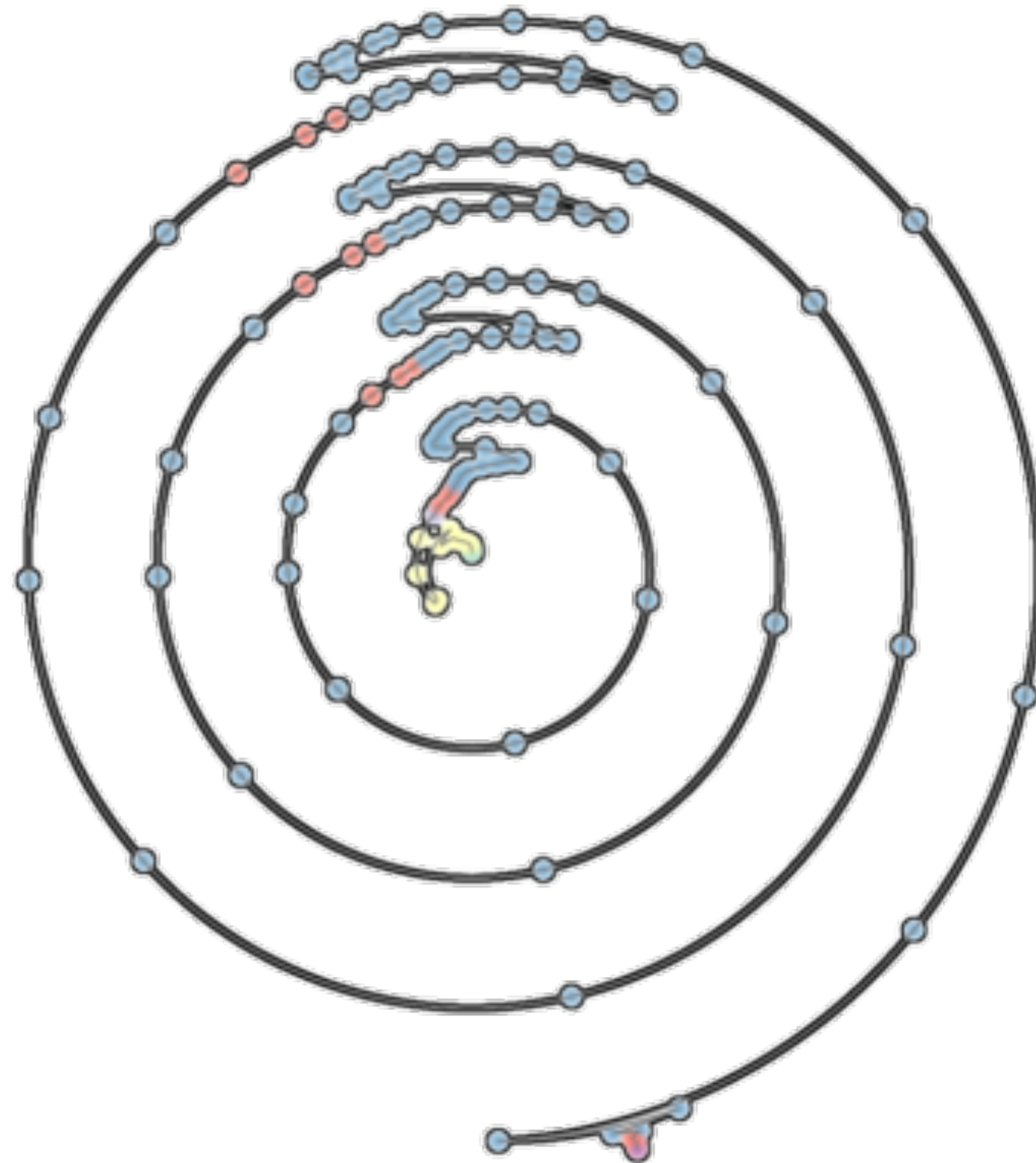
Observing the red shift



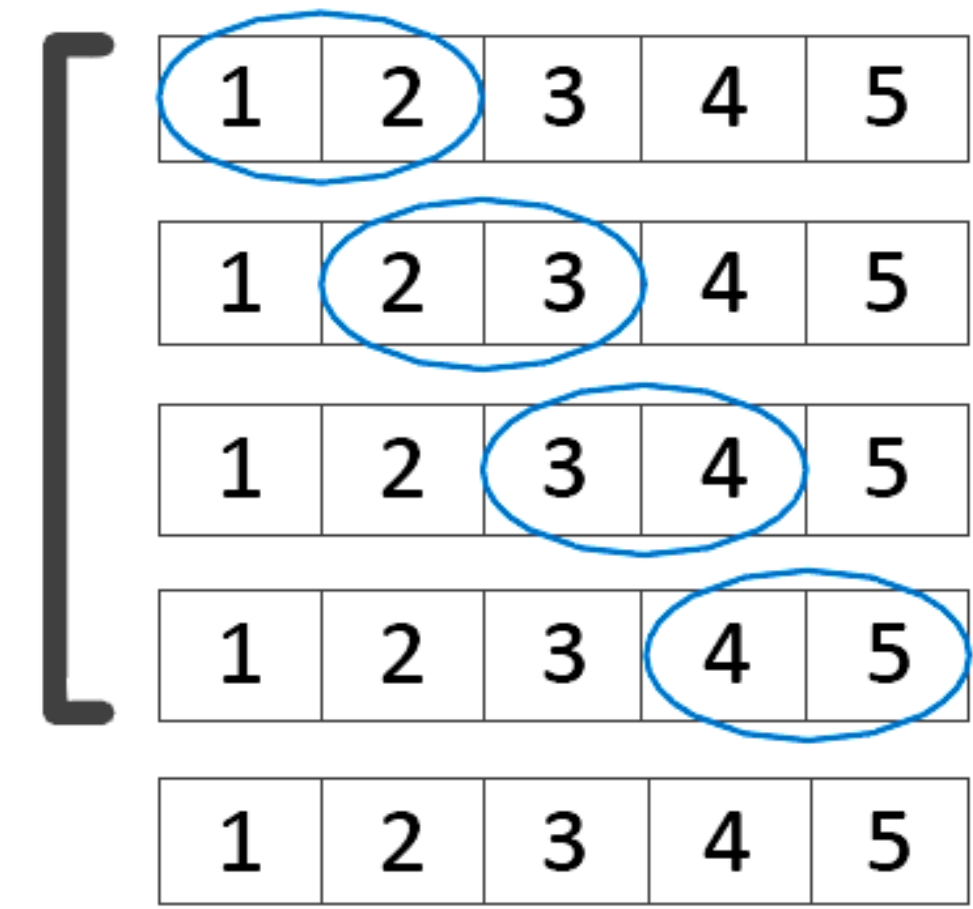
Software Visualization

Circular patterns in a program

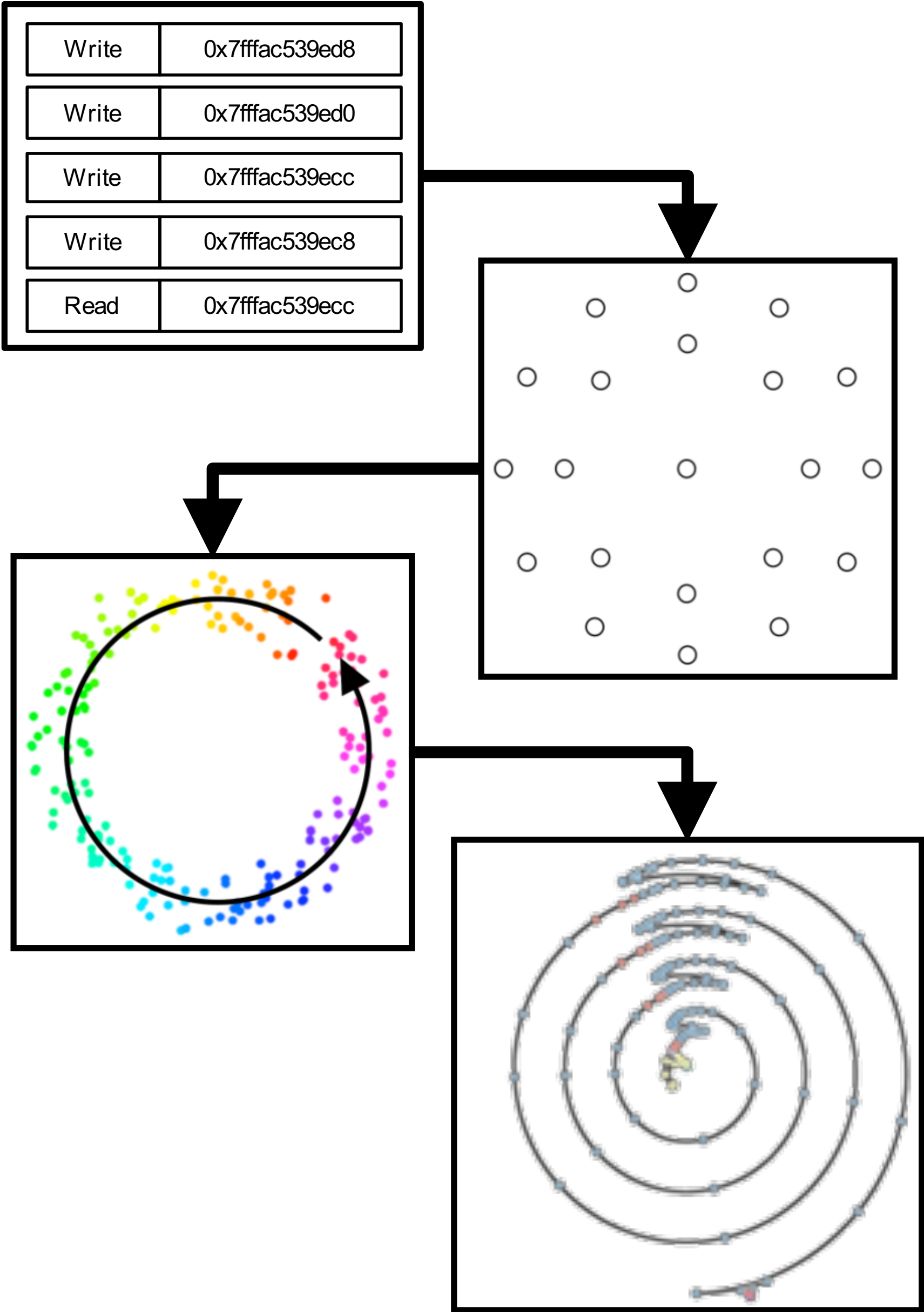
An example



```
File: sort.cpp
1: void bubblesort(std::vector<double>& v){
2:   for(unsigned end=v.size()-1; end >= 0; end--){
3:     bool swapped = false;
4:     for(unsigned i=0; i<end; i++){
5:       if(v[i] > v[i+1]){
6:         std::swap(v[i], v[i+1]);
7:         swapped = true;
8:       }
9:     }
10:    if(!swapped) break;
11:  }
12: }
```



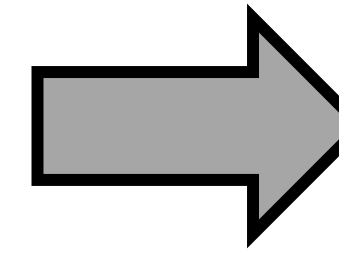
Convert memory reference traces to a point cloud



- Execute an application to capture memory reference trace
- Convert to high-dimensional point cloud
- Topological analysis identify cycles
- Visualize result

Capturing a memory reference trace

```
File: sort.cpp
1: void bubblesort(std::vector<double>& v){
2:   for(unsigned end=v.size()-1; end >= 0; end--){
3:     bool swapped = false;
4:     for(unsigned i=0; i<end; i++){
5:       if(v[i] > v[i+1]){
6:         std::swap(v[i], v[i+1]);
7:         swapped = true;
8:       }
9:     }
10:    if(!swapped) break;
11:  }
12: }
```



Write	0x7fffac539ed8
-------	----------------

Write	0x7fffac539ed0
-------	----------------

Write	0x7fffac539ecc
-------	----------------

Write	0x7fffac539ec8
-------	----------------

Read	0x7fffac539ecc
------	----------------

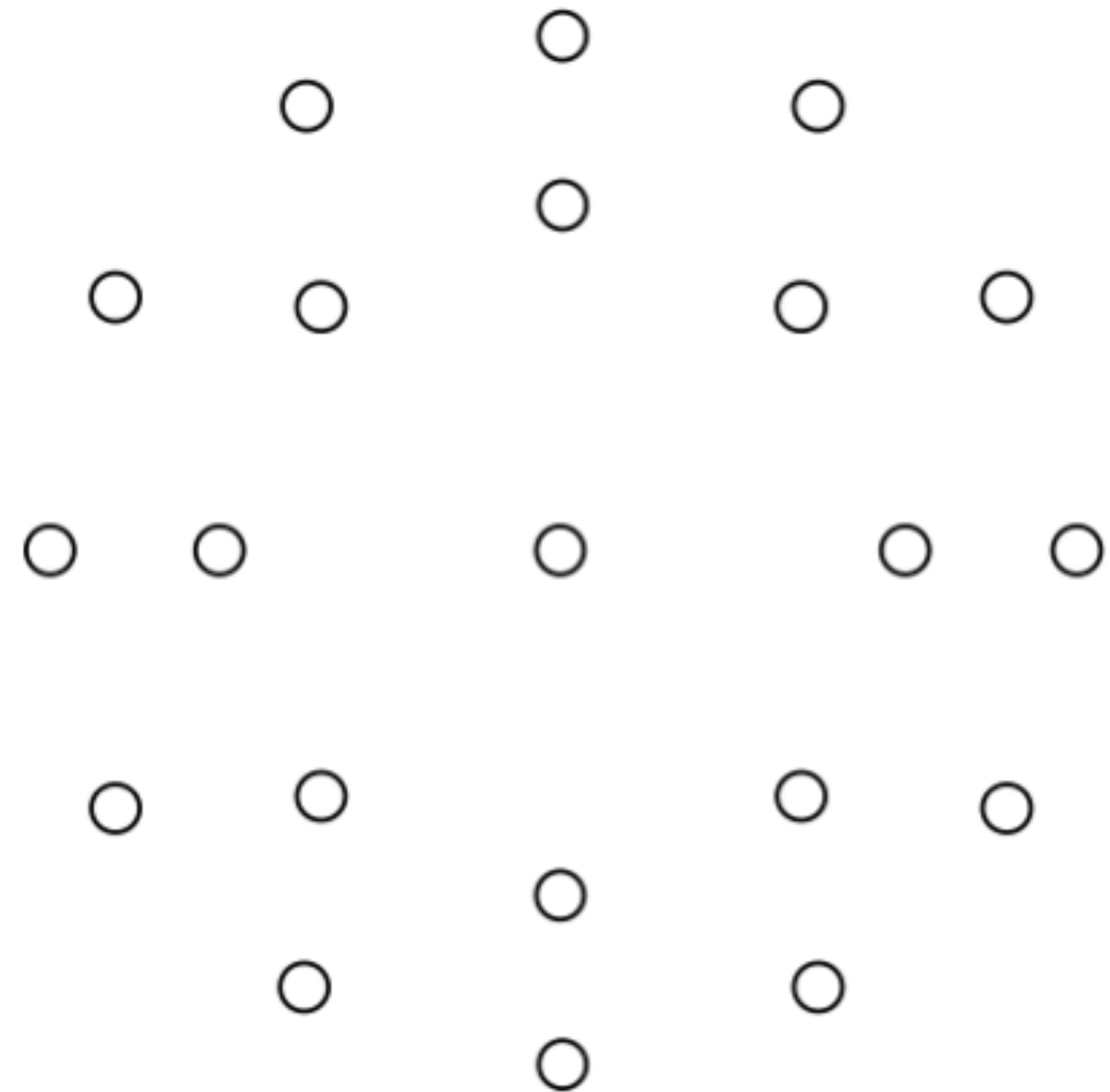
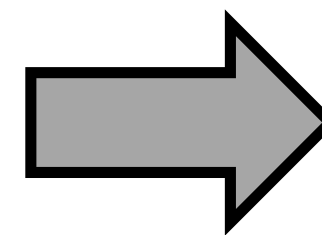
Read	0x7fffac539ec8
------	----------------

Write	0x7fffac539eb8
-------	----------------

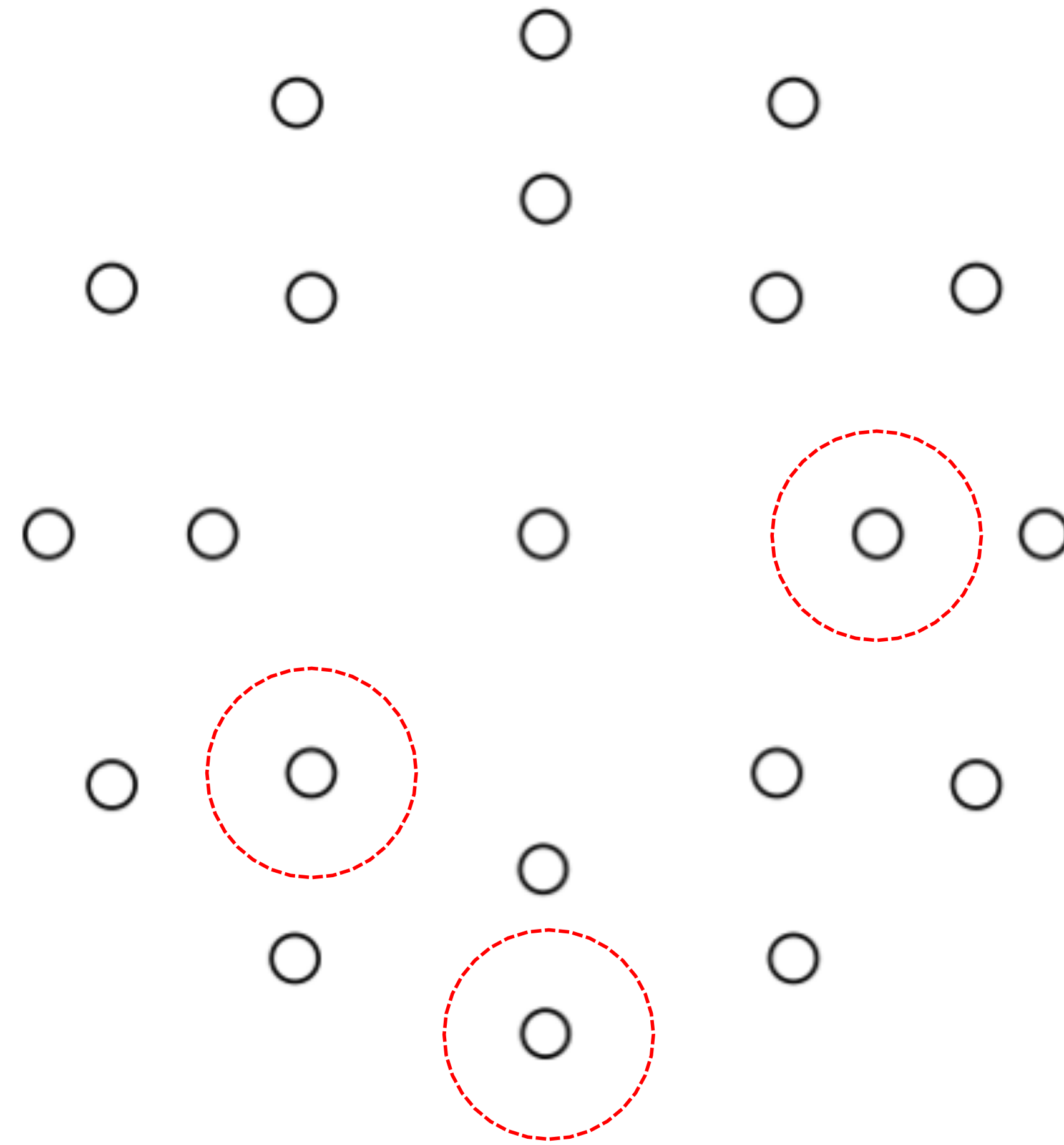
Write	0x7fffac539eb0
-------	----------------

Memory reference trace to point cloud

Write	0x7fffac539ed8
Write	0x7fffac539ed0
Write	0x7fffac539ecc
Write	0x7fffac539ec8
Read	0x7fffac539ecc
Read	0x7fffac539ec8
Write	0x7fffac539eb8
Write	0x7fffac539eb0



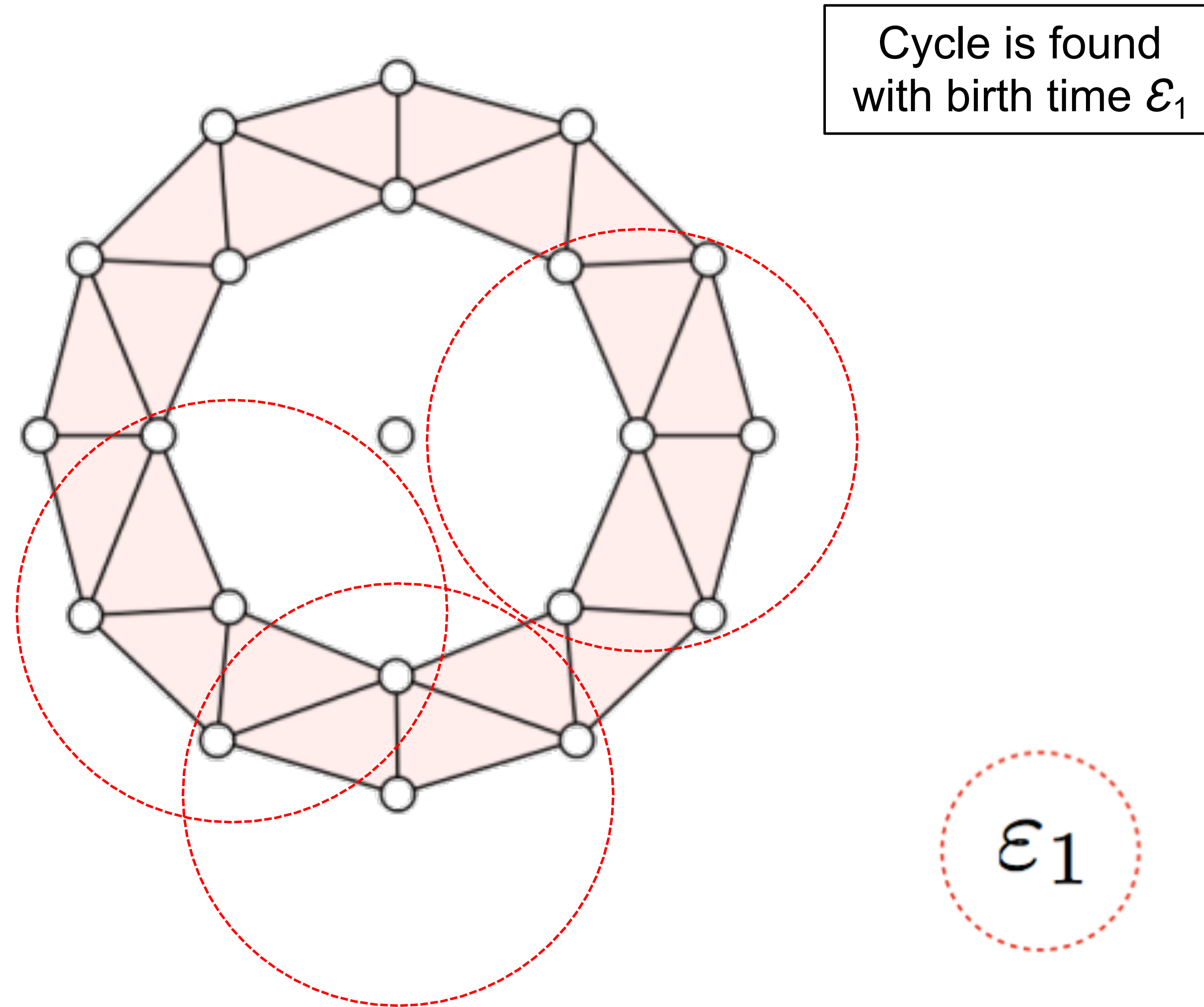
Topological data analysis to identify cycles



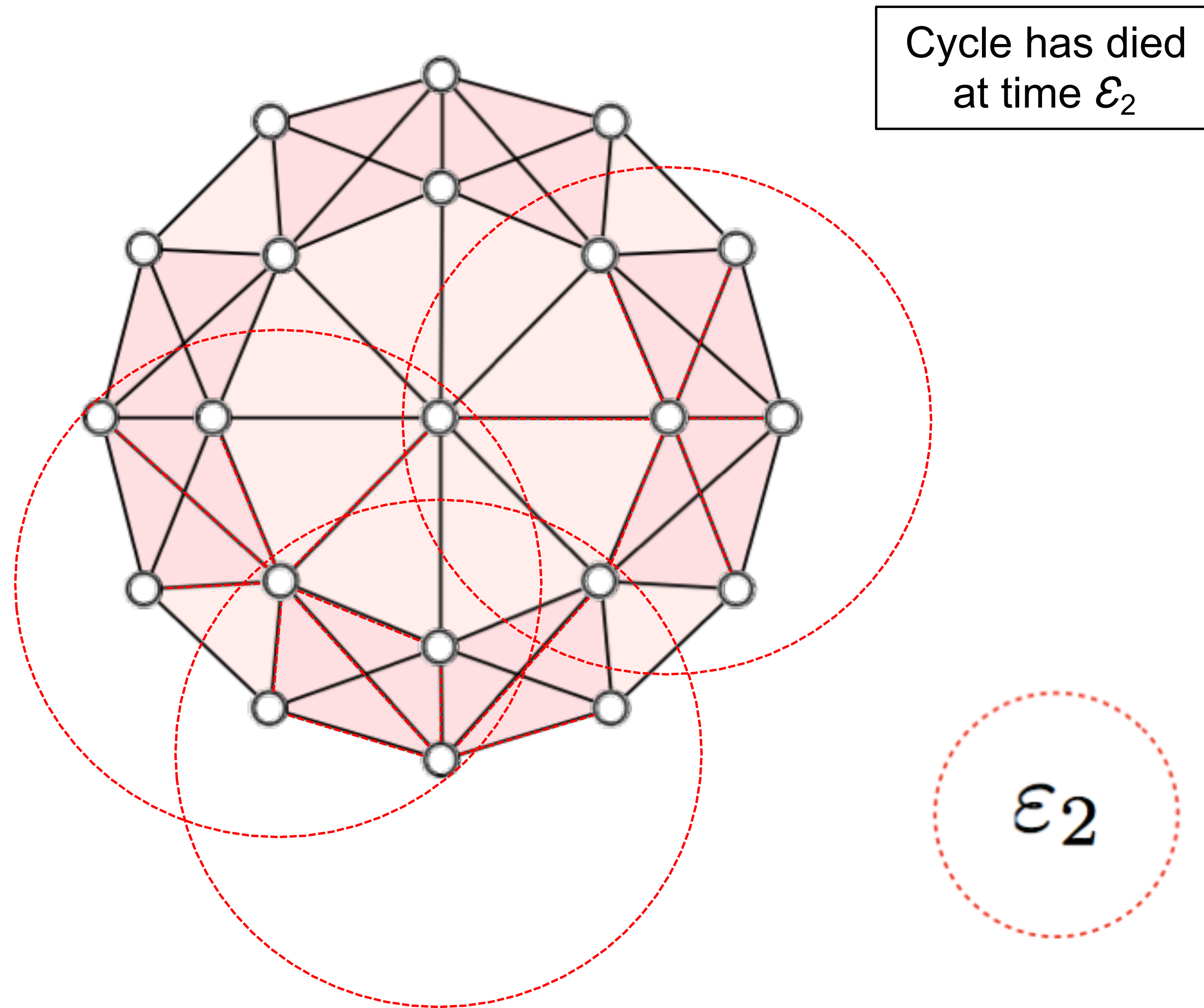
No connectivity
No cycles

ϵ_0

Topological data analysis to identify cycles

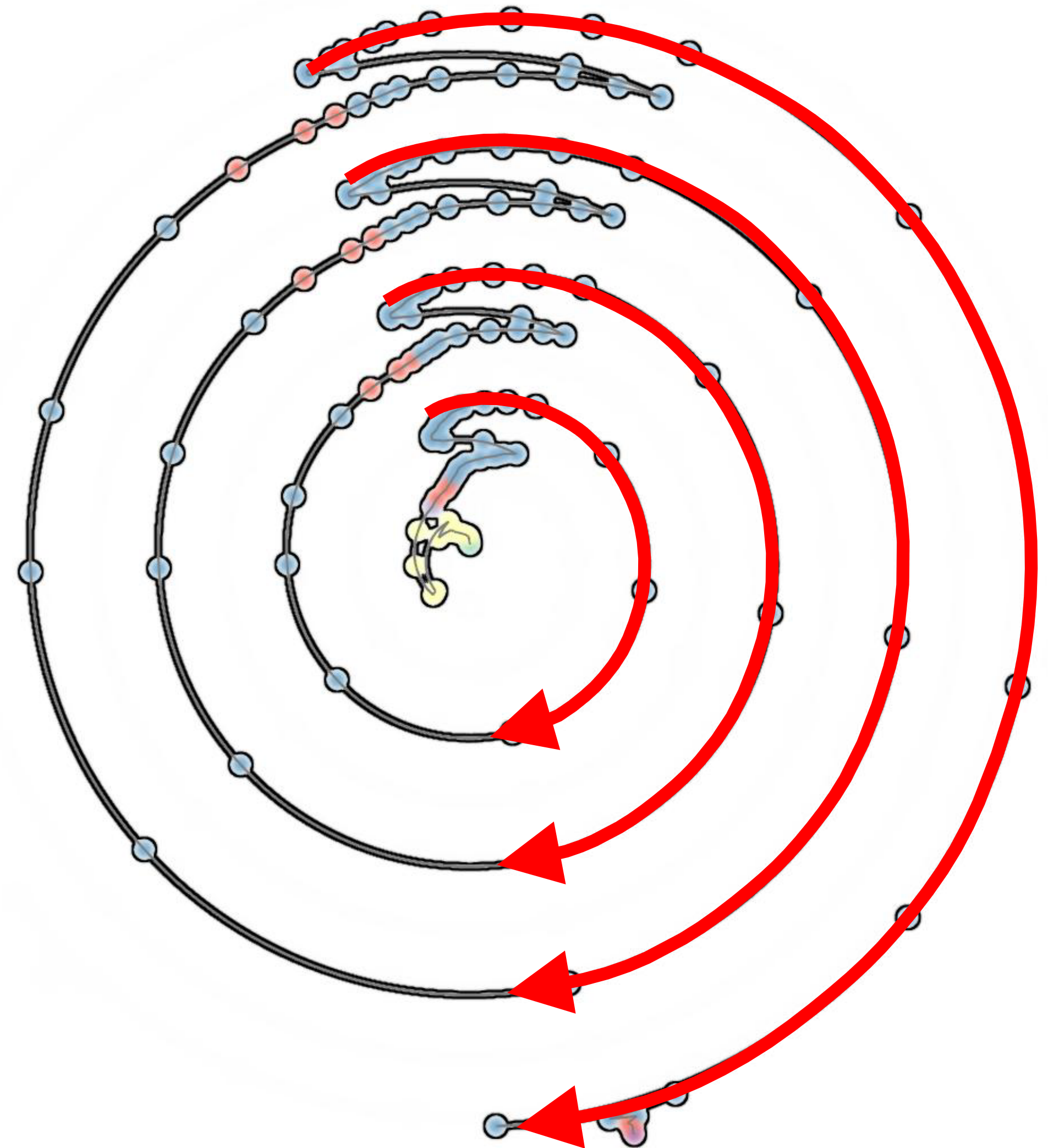
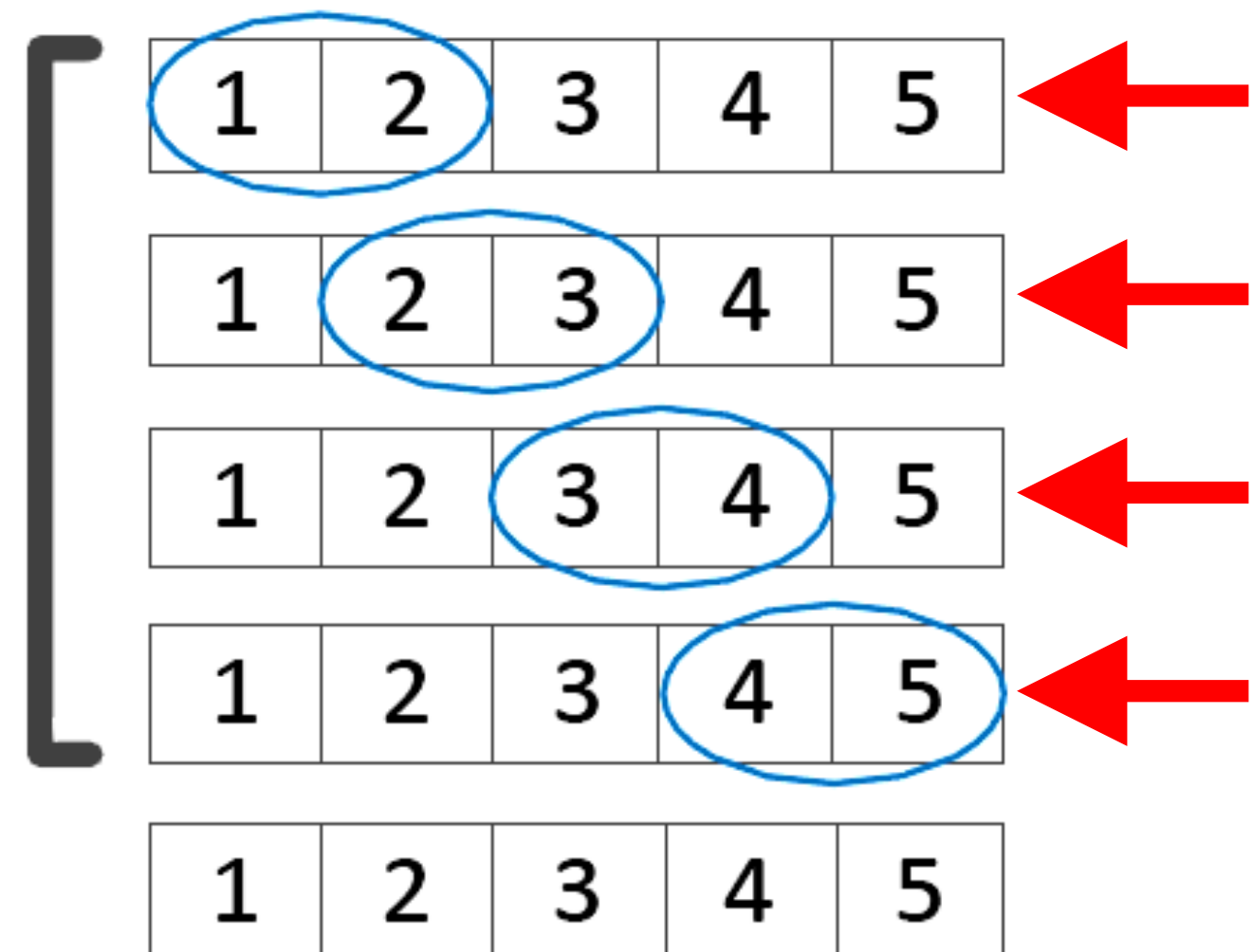


Topological data analysis to identify cycles



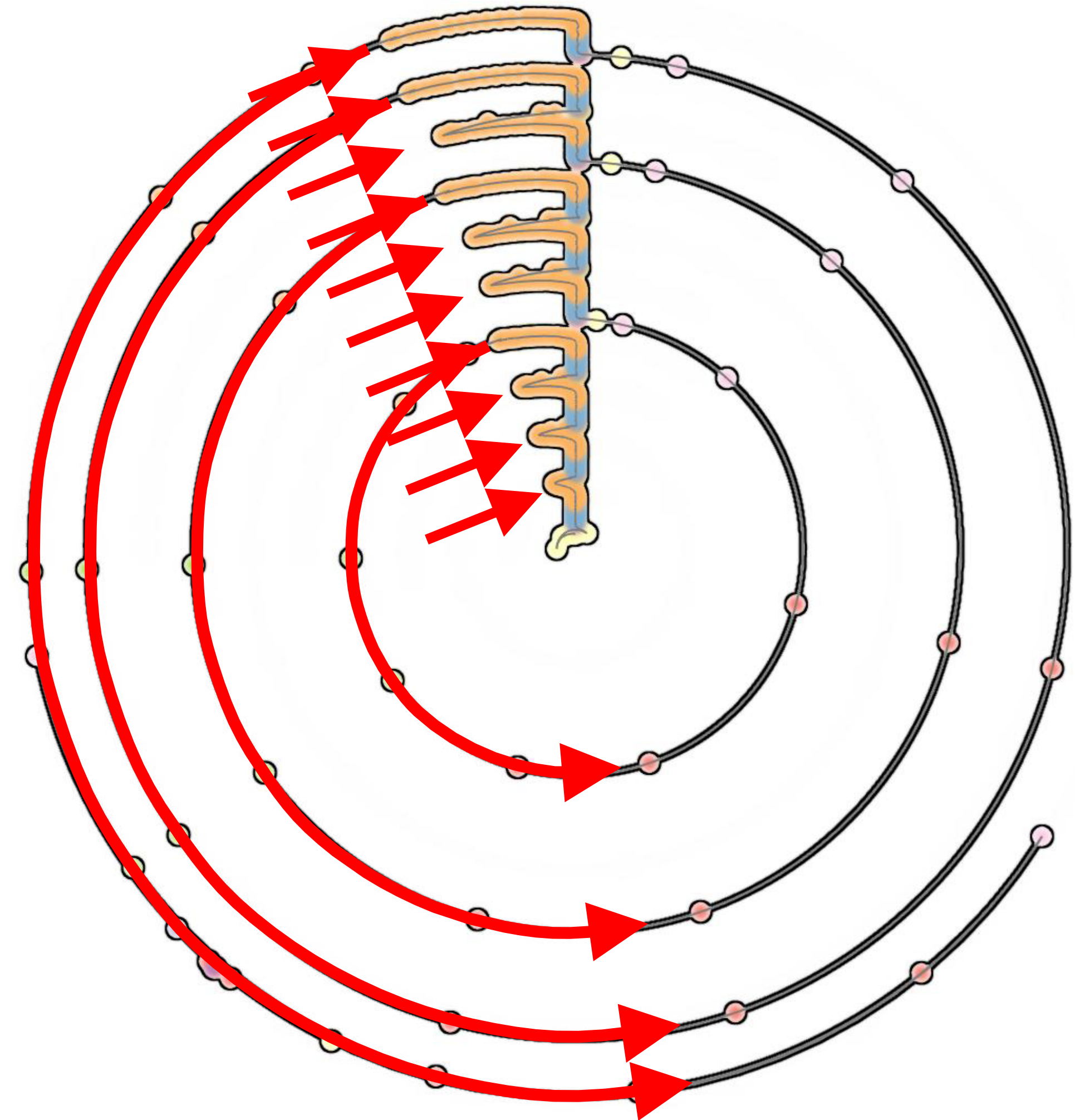
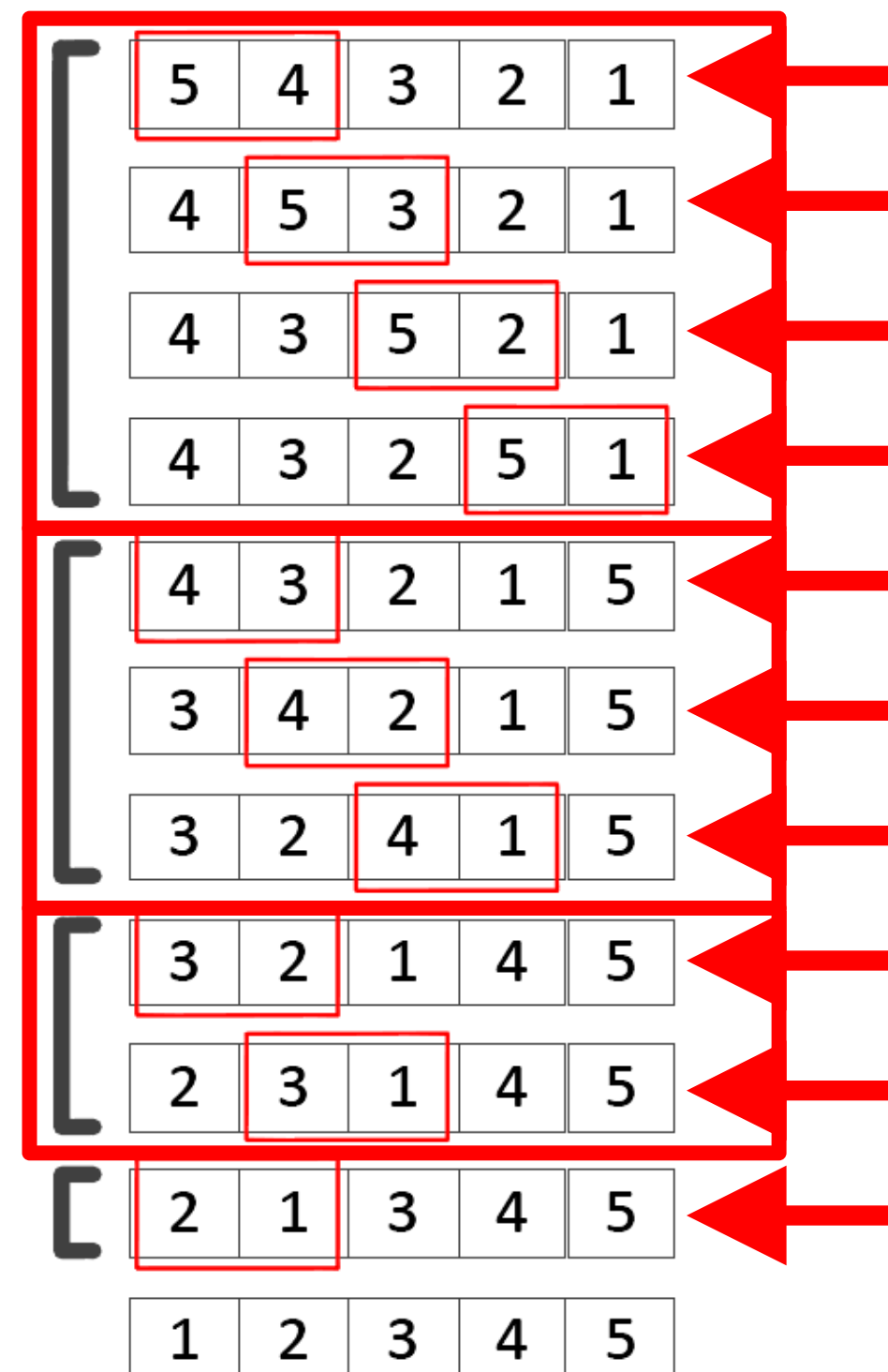
Data dependent structures

```
File: sort.cpp
1: void bubblesort(std::vector<double>& v){
2:   for(unsigned end=v.size()-1; end >= 0; end--){
3:     bool swapped = false;
4:     for(unsigned i=0; i<end; i++){
5:       if(v[i] > v[i+1]){
6:         std::swap(v[i], v[i+1]);
7:         swapped = true;
8:       }
9:     }
10:    if(!swapped) break;
11:  }
12: }
```



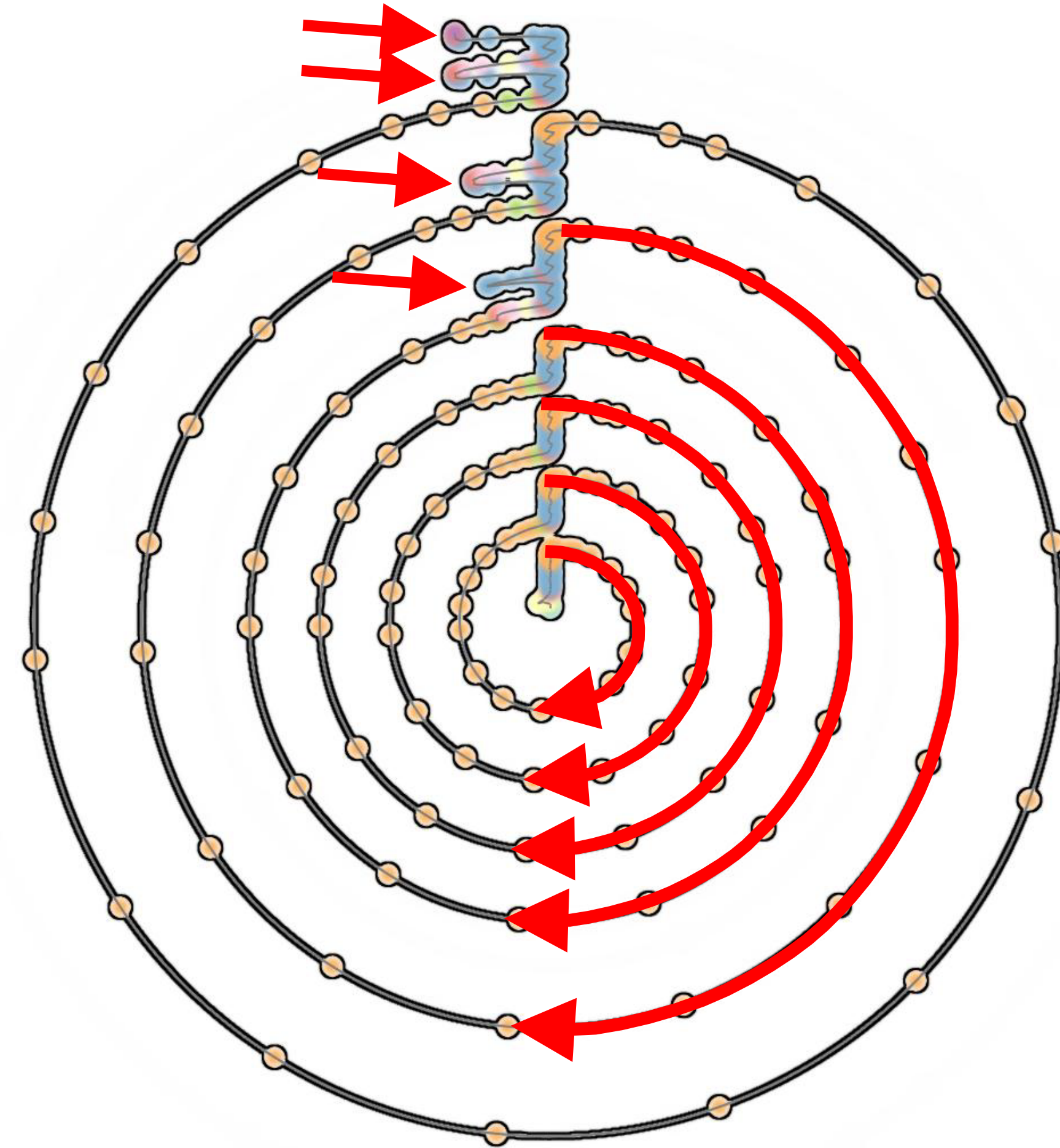
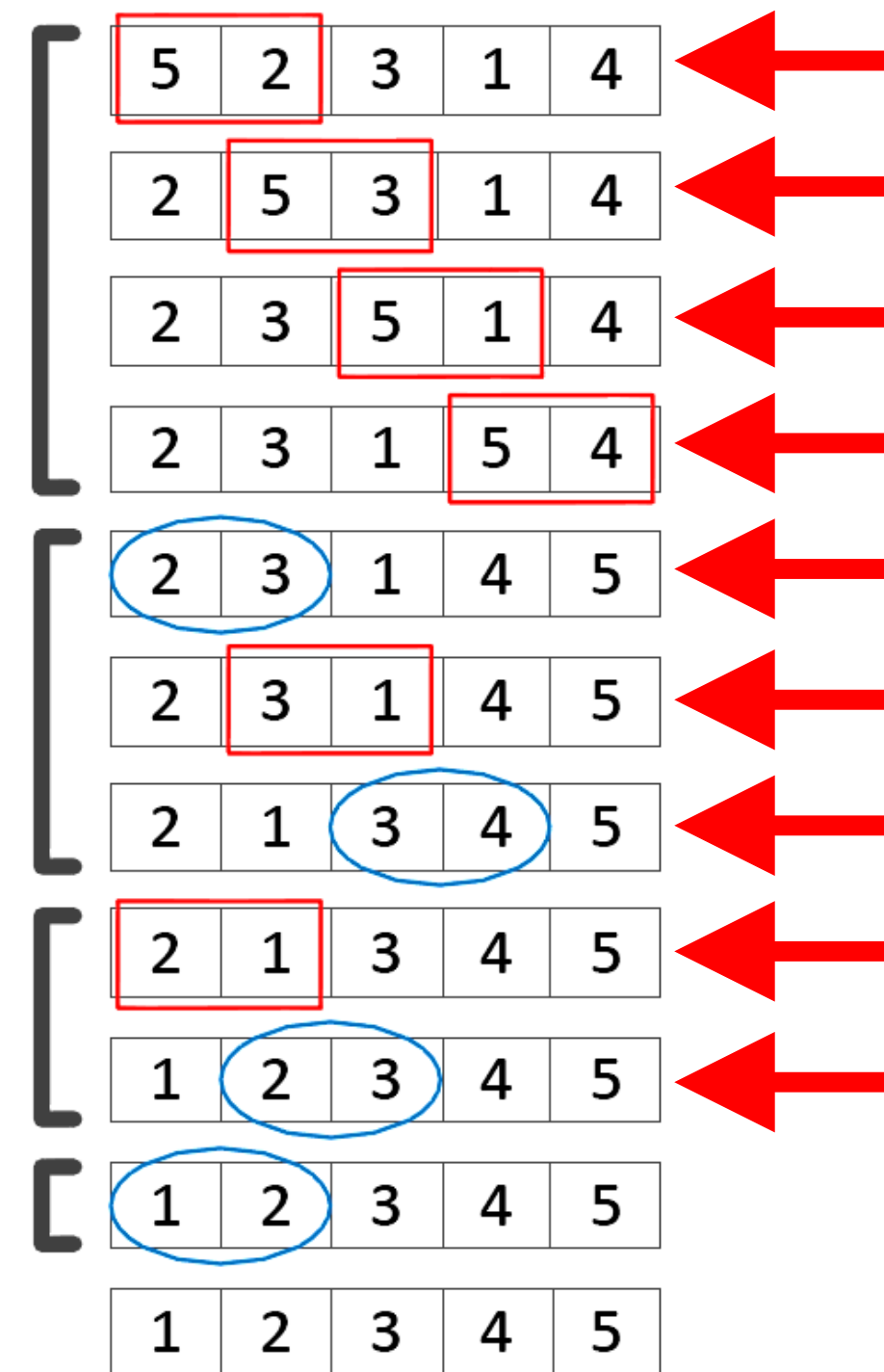
Data dependent structures

```
File: sort.cpp
1: void bubblesort(std::vector<double>& v){
2:   for(unsigned end=v.size()-1; end >= 0; end--){
3:     bool swapped = false;
4:     for(unsigned i=0; i<end; i++){
5:       if(v[i] > v[i+1]){
6:         std::swap(v[i], v[i+1]);
7:         swapped = true;
8:       }
9:     }
10:    if(!swapped) break;
11:  }
12: }
```



Data dependent structures

```
File: sort.cpp
1: void bubblesort(std::vector<double>& v){
2:   for(unsigned end=v.size()-1; end >= 0; end--){
3:     bool swapped = false;
4:     for(unsigned i=0; i<end; i++){
5:       if(v[i] > v[i+1]){
6:         std::swap(v[i], v[i+1]);
7:         swapped = true;
8:       }
9:     }
10:    if(!swapped) break;
11:  }
12: }
```

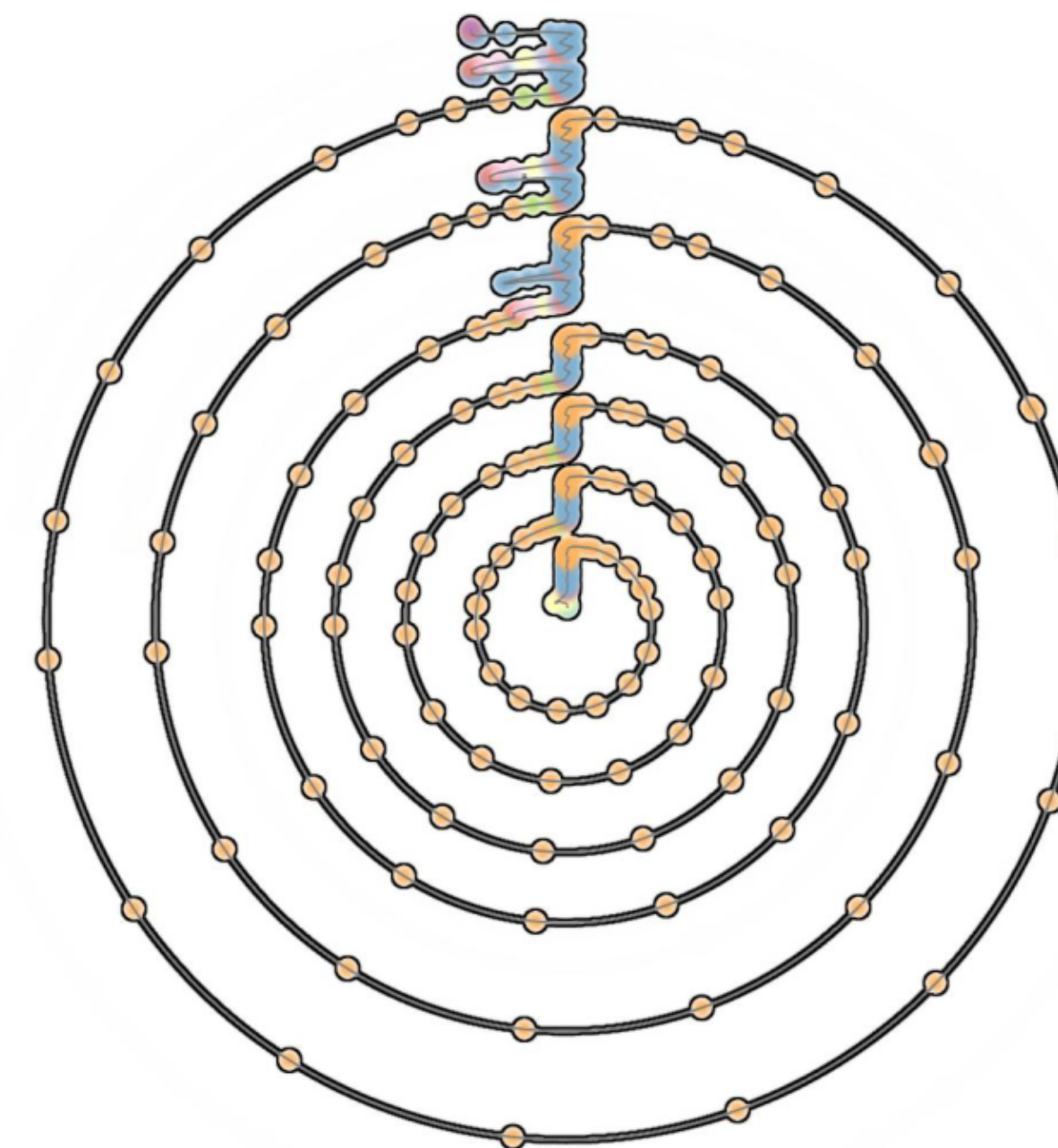
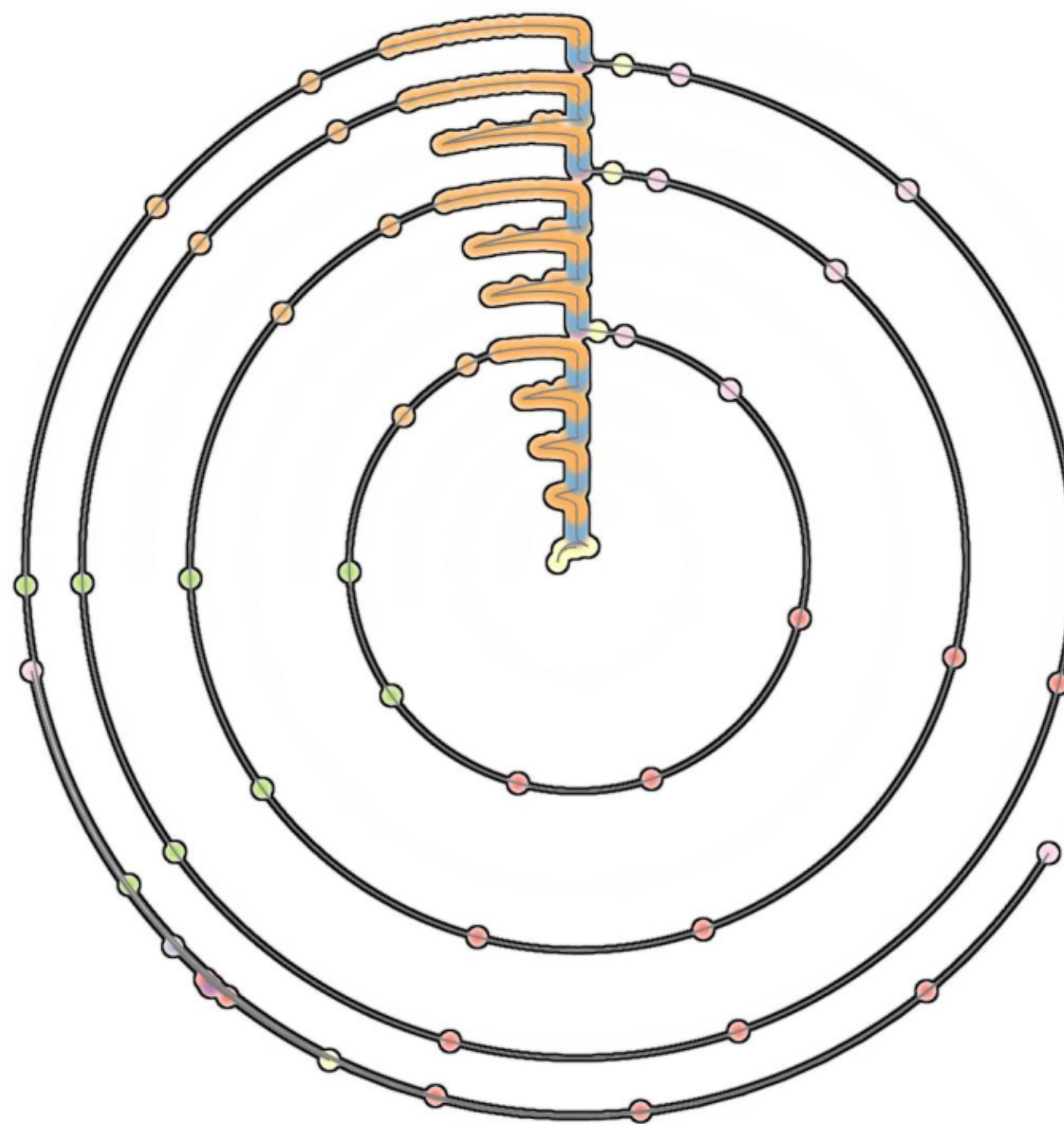
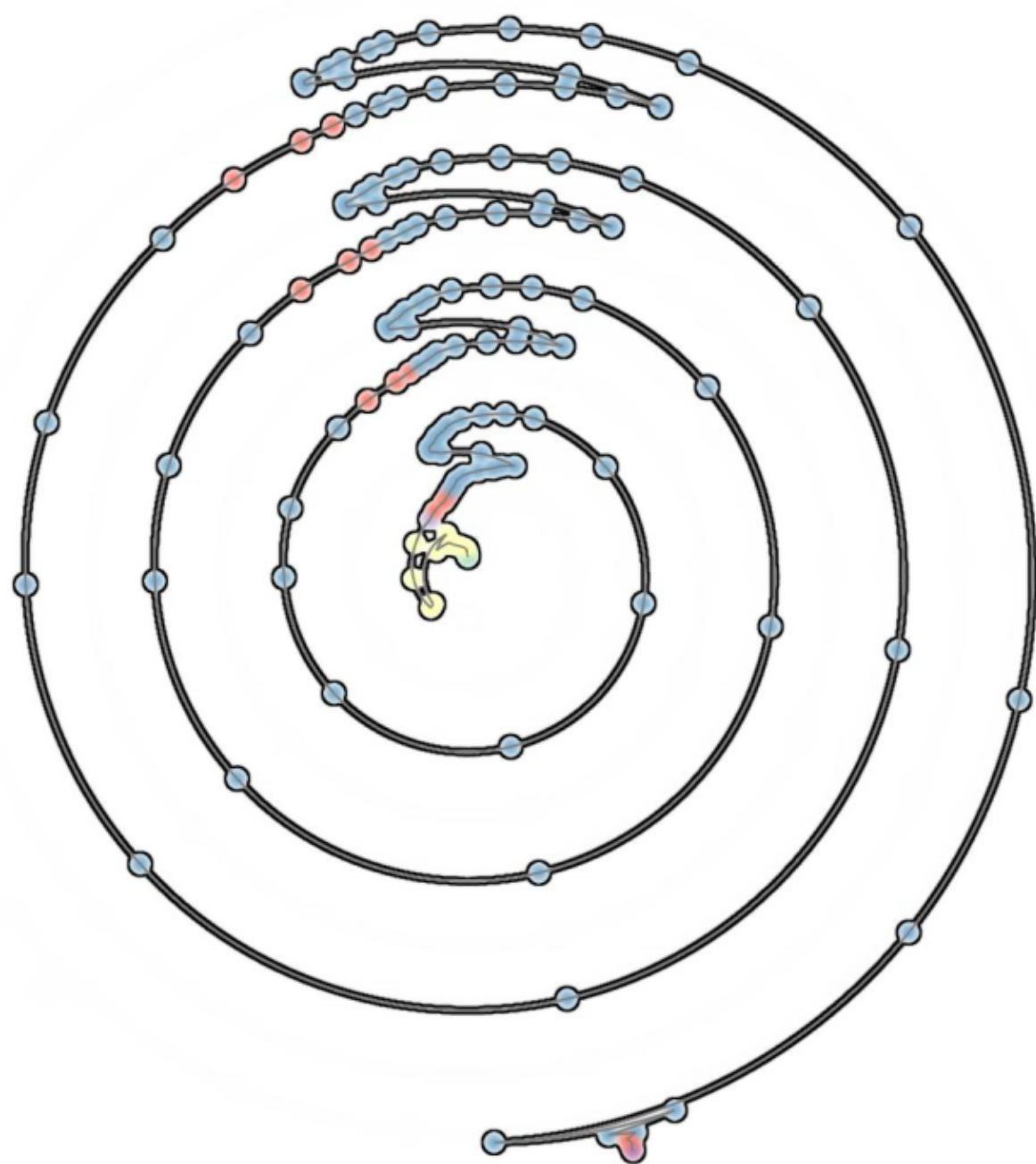


Data dependent structures

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

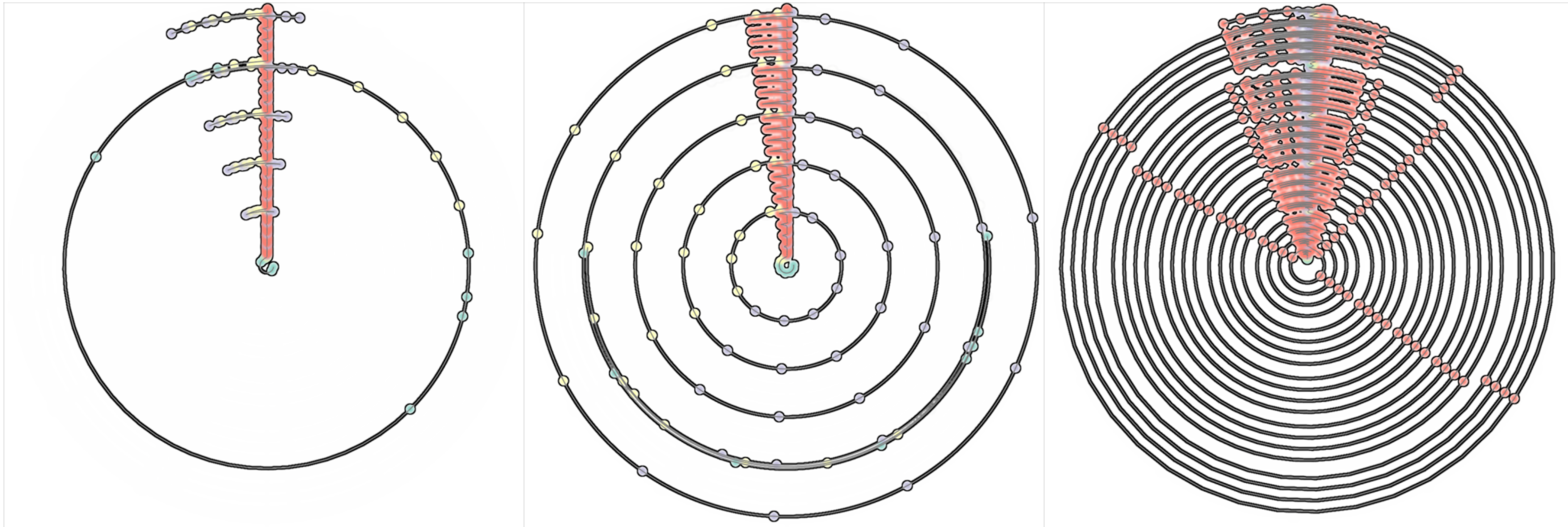
5	4	3	2	1
4	5	3	2	1
4	3	5	2	1
4	3	2	5	1
4	3	2	1	5
3	4	2	1	5
3	2	4	1	5
3	2	1	4	5
2	3	1	4	5
2	1	3	4	5
1	2	3	4	5

5	2	3	1	4
2	5	3	1	4
2	3	5	1	4
2	3	1	5	4
2	3	1	4	5
2	3	1	4	5
2	1	3	4	5
2	1	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5



Algorithm dependent structures

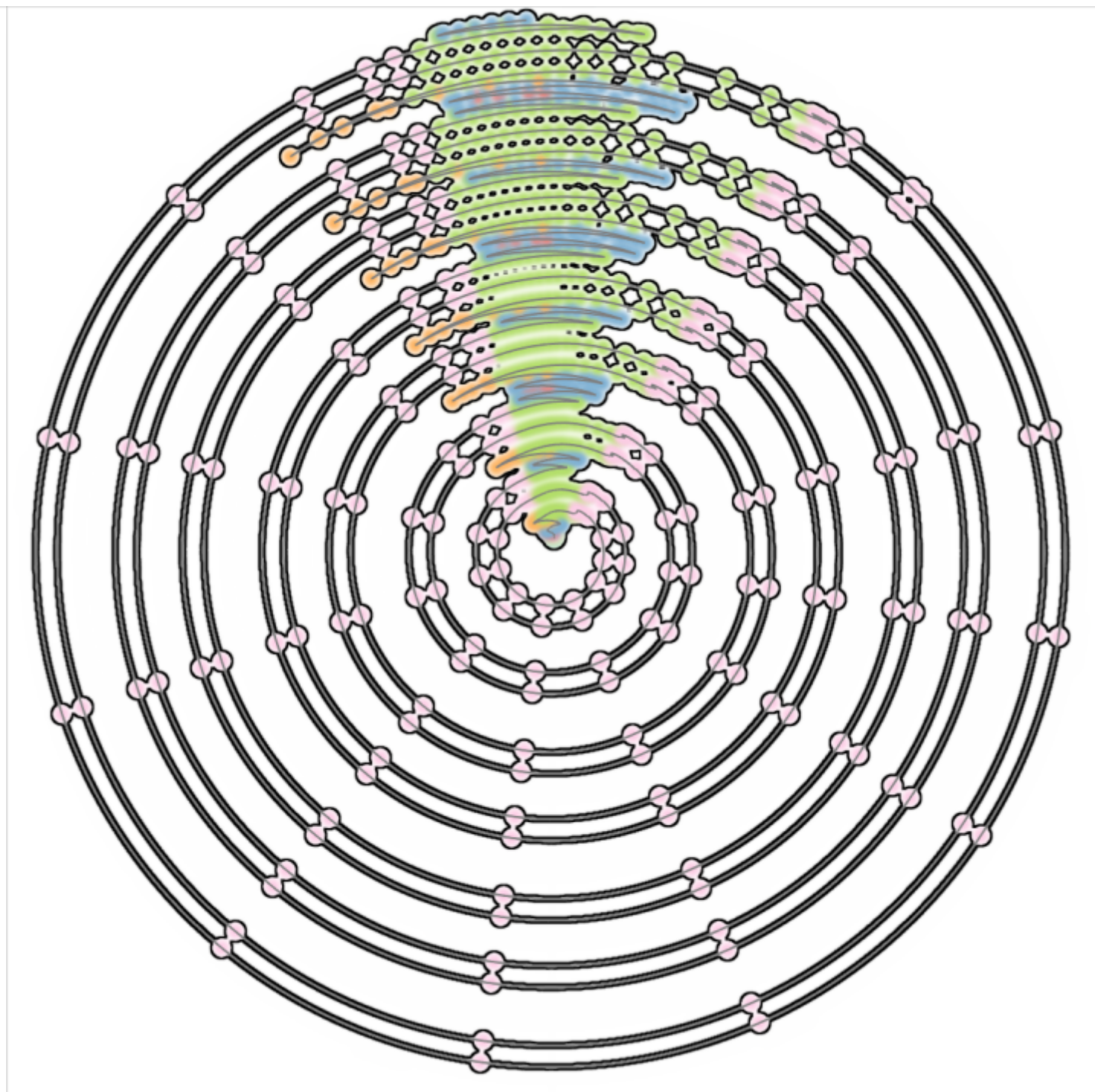
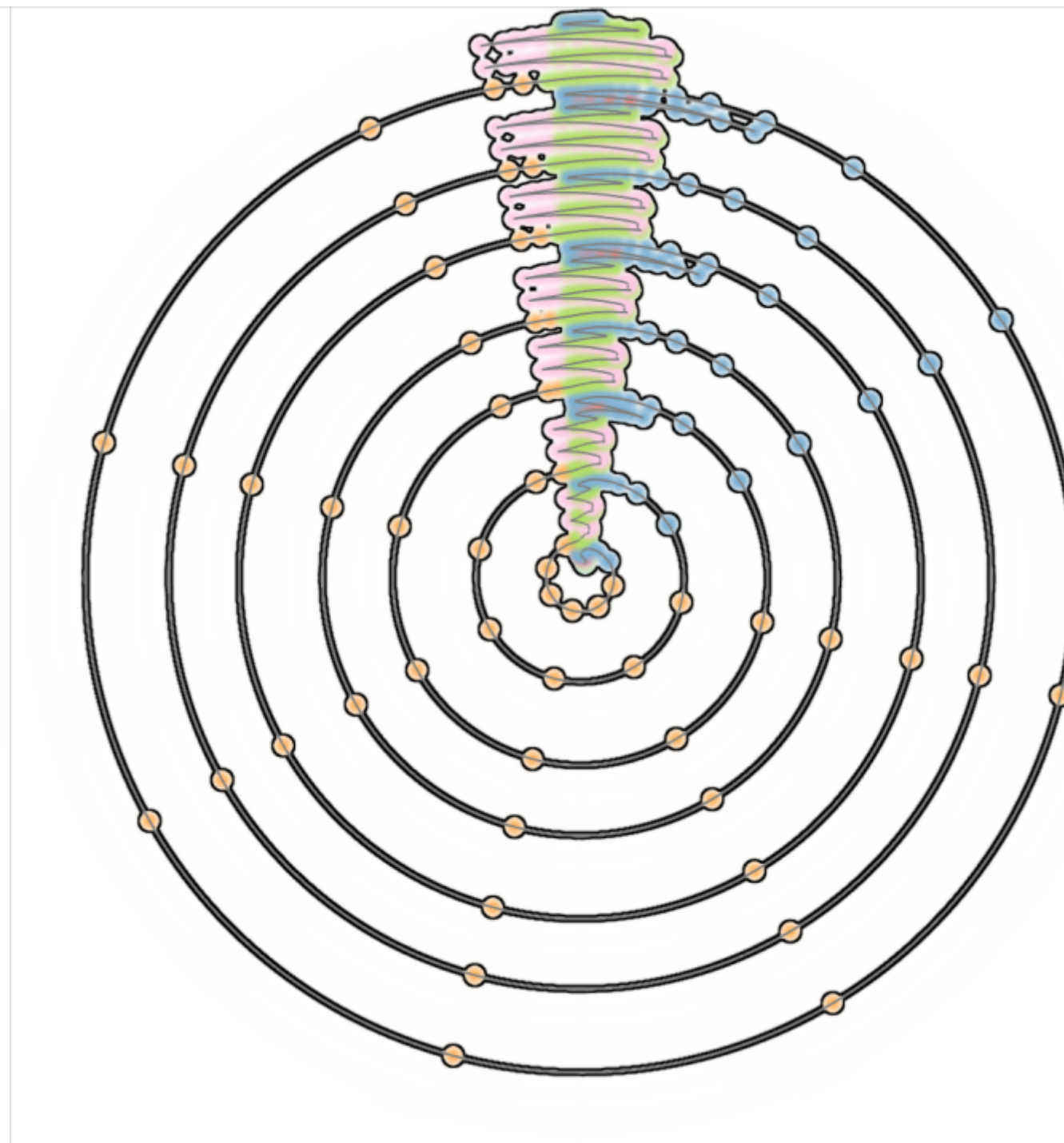
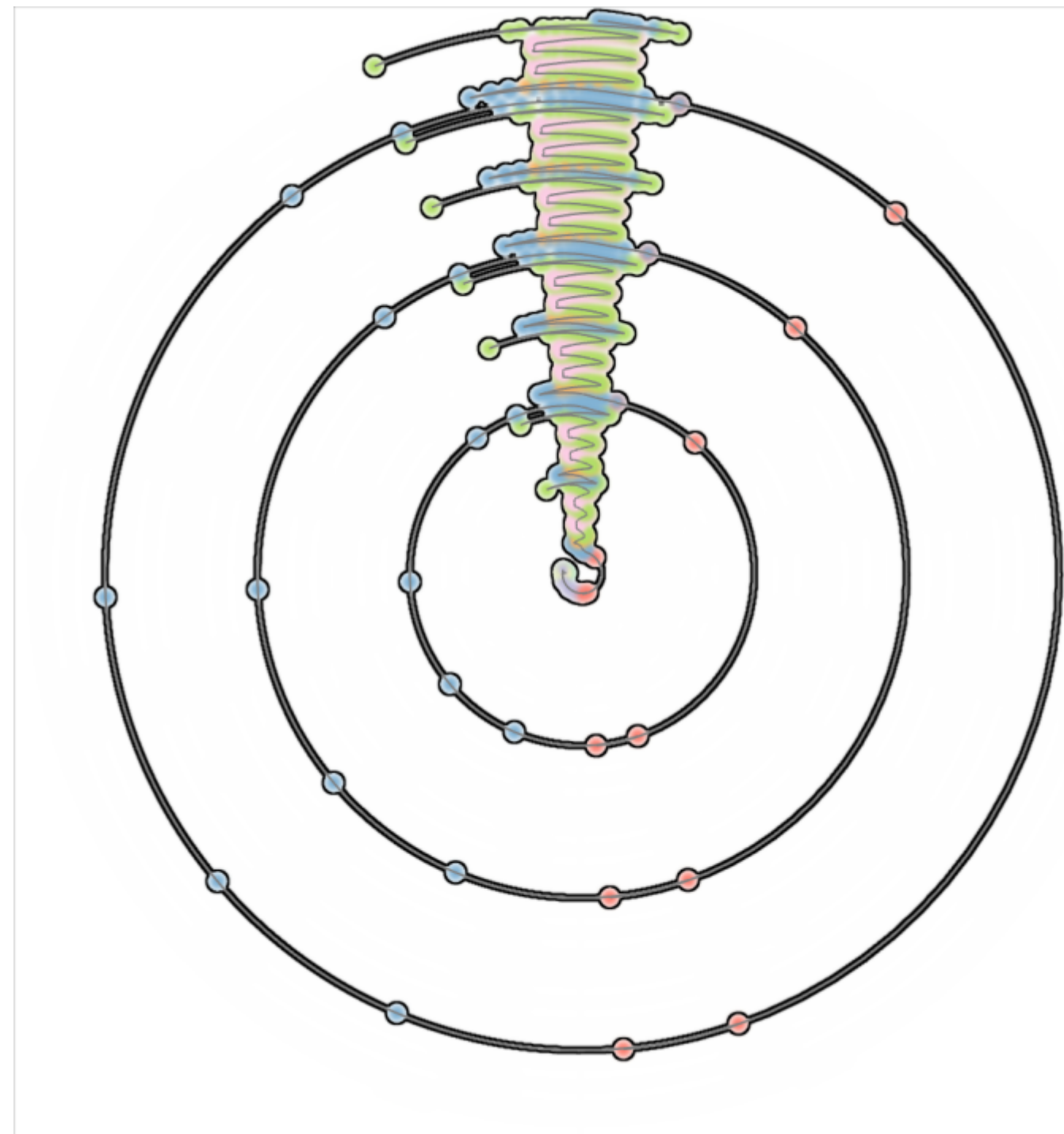
```
File: matmult.cpp
1: unsigned int i, j, k;
2: for (i = 0; i < N; i++)
3:   for (j = 0; j < N; j++)
4:     for (k = 0; k < N; k++)
5:       linC[i*N + j] += linA[i*N + k] * linB[k*N + j];
```



Algorithm dependent structures

File: blocked-matmult.cpp

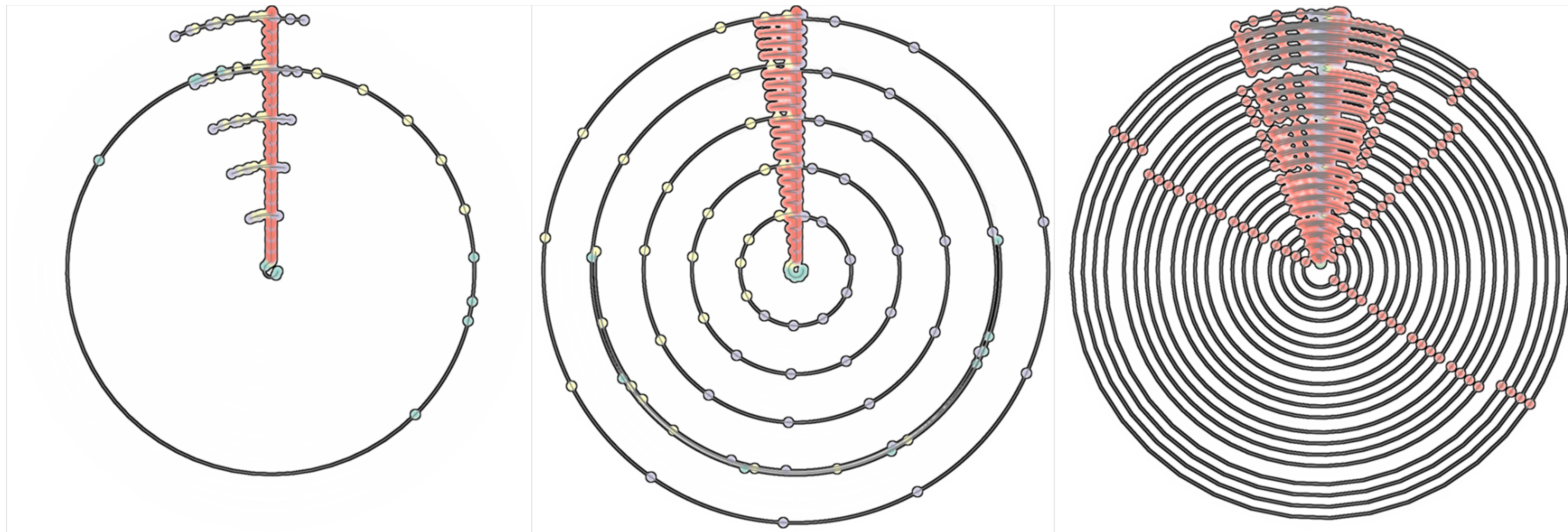
```
1: unsigned int i, j, k, j0, k0;  
2: for (k0 = 0; k0 < N; k0 += b)  
3:   for (j0 = 0; j0 < N; j0 += b)  
4:     for (i = 0; i < N; i++)  
5:       for (k = k0; k < min(k0 + b, N); k++) {  
6:         r = linA[i*N + k];  
7:         for (j = j0; j < min(j0 + b, N); j++)  
8:           linC[i*N + j] += r*linB[k*N + j];  
9:       }
```



Algorithm dependent structures

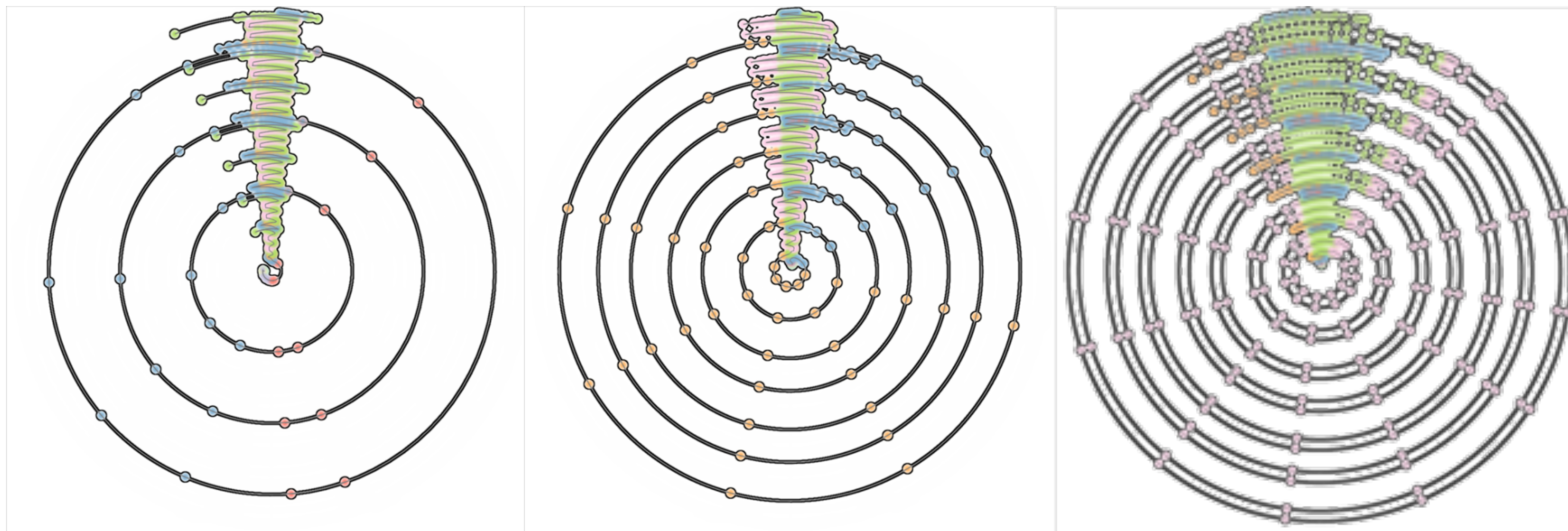
Naïve Matrix

Multiply



Blocked Matrix

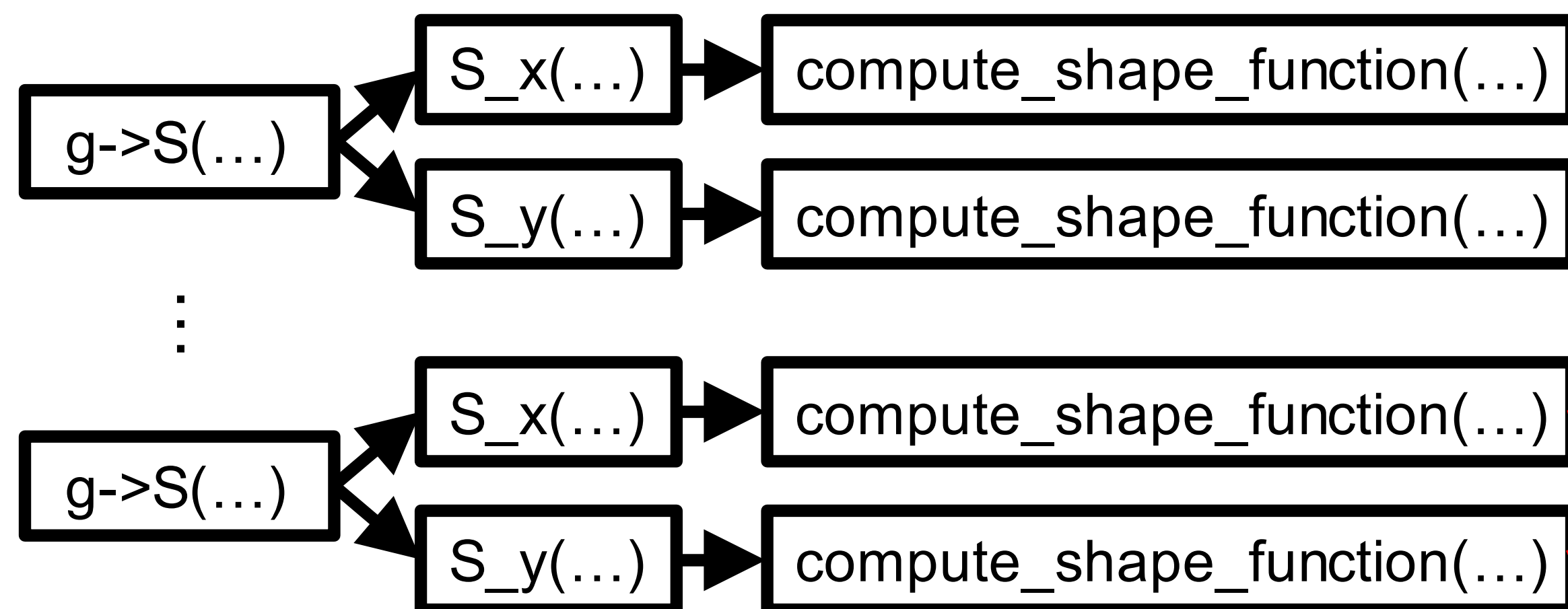
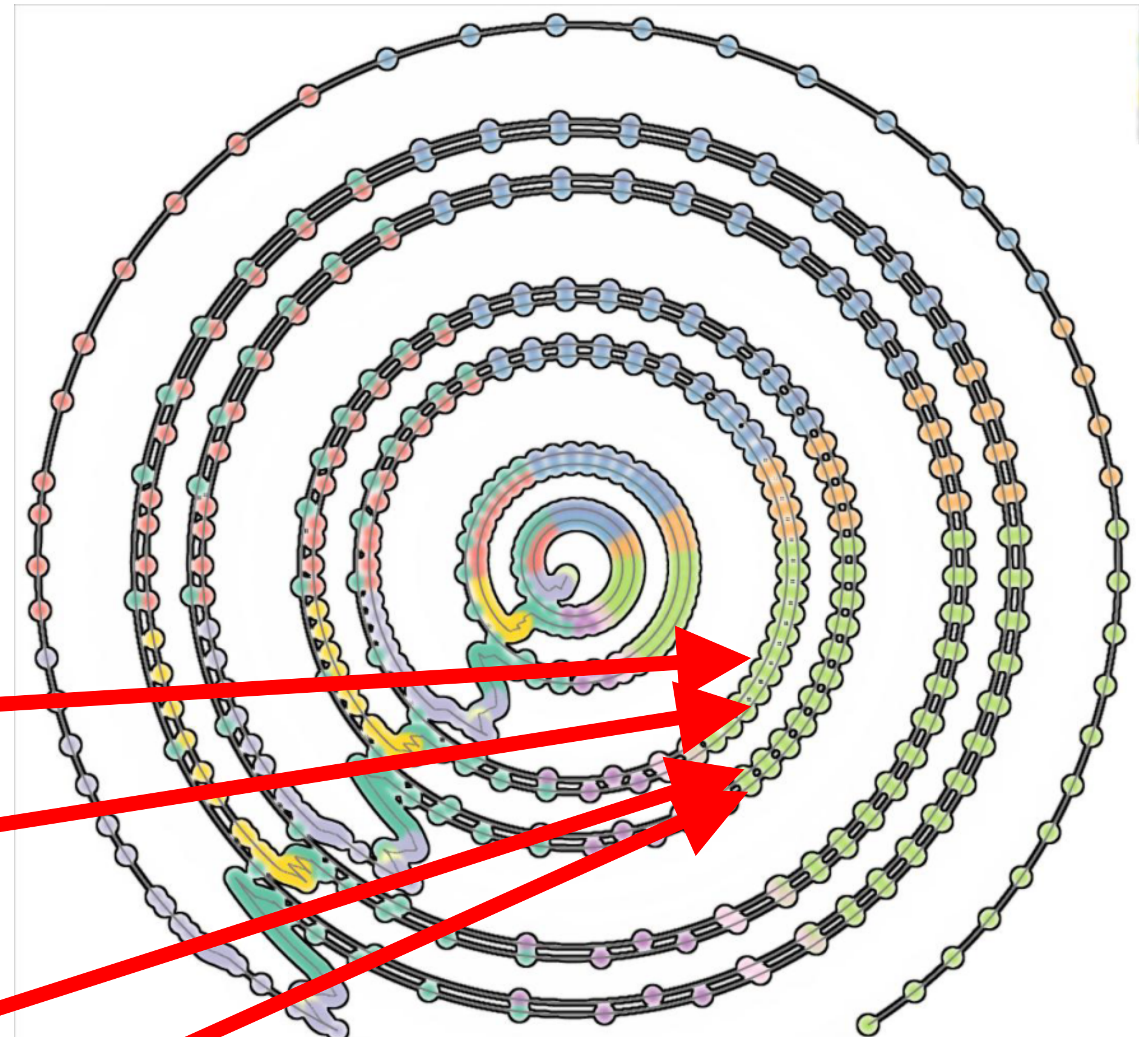
Multiply



Non-loop based structures

```
File: MPM.cpp
1: for(unsigned ii=i; ii<=i+1; ii++){
2:   for(unsigned jj=j; jj<=j+1; jj++){
3:     g->mass(ii,jj) += g->S(ii, jj, mp->position(p)) * mp->mass(p);
4:     g->momentum(ii,jj) += g->S(ii, jj, mp->position(p)) * mp->mass(p) * mp->velocity(p);
5:   }
6: }

File: Grid.h
1: double S(int i, int j, const Point& p){ ... }
2: unsigned indexify(unsigned i, unsigned j) const { ... }
3: double S_x(int i, double x){ ... }
4: double S_y(int j, double y){ ... }
5: static double compute_shape_function(int cell, double position, double cell_size){
6:   // This is the distance of "position" from the position of "cell".
7:   const double cell_distance = std::abs((position - cell_size*cell) / cell_size);
8:   // Perform case analysis.
9:   if(cell_distance >= 1.0){
10:    return 0.0;
11:   }
12:   else{
13:    return 1.0 - cell_distance;
14:   }
15: }
```



Topology Tool Kit

TTK

Installation Demo

TTK installation tutorial

```
ttk@ttk-VirtualBox: ~/code/ttk-0.9/build
-- Installing: /usr/local/lib/paraview-5.2/plugins/libUncertainDataEstimator.so
-- Set runtime path of "/usr/local/lib/paraview-5.2/plugins/libUncertainDataEstimator.so" to "/usr/local/bin/.."
ttk@ttk-VirtualBox:~/code/ttk-0.9/build$ ls /usr/local/bin/*Cnd
/usr/local/bin/blankCnd
/usr/local/bin/continuousScatterplotCnd
/usr/local/bin/contourForestsCnd
/usr/local/bin/geometrySmootherCnd
/usr/local/bin/jacobiSetCnd
/usr/local/bin/pointDataToCellDataCnd
```

TTK v0.9 (c) 2017

Getting Started Guide Example Visualizations

9:44 / 10:40

CC HD YouTube

<https://youtu.be/etAe13KEWsk>

Dragon Demo (contour tree)

TTK usage tutorial - Dragon demo

TTK usage logs (repeated):

```
[Common] Welcome!  
[vtkPersistenceDiagram] Starting computation on field 'Elevation'...  
[ContourForests] Initialization : 3.60812e-05  
[ContourForests] Sort scalars (-mirror) : 0.00673103  
[ContourForests] Interface and overlap init. : 0.00521493  
[ContourForests] Parallel allocations : 0.00123596  
[ParallelBuild] Merge Tree 0 constructed in : 0.0298059  
[ContourForests] ParallelBuild : 0.039411  
[ContourForests] Stitch : 9.53674e-07  
[ContourForests] Create Contour tree : 0.00135207  
[ContourForests] TOTAL : 0.0481222  
Tree computed ...  
[vtkPersistenceDiagram] Memory usage: 1.59375 MB.  
[Common] Welcome!  
[vtkSphereFromPoint] Spheres computed in 0.0111189 s.  
[vtkSphereFromPoint] Memory usage: 0.148438 MB.  
[vtkSphereFromPoint] Spheres computed in 0.0102599 s.  
[vtkSphereFromPoint] Memory usage: 0.40332 MB.  
[vtkSphereFromPoint] Spheres computed in 0.00801682 s.  
[vtkSphereFromPoint] Memory usage: 0.130659 MB.  
[vtkSphereFromPoint] Spheres computed in 0.00624895 s.  
[vtkSphereFromPoint] Memory usage: 0 MB.  
[vtkSphereFromPoint] Spheres computed in 0.00735998 s.  
[vtkSphereFromPoint] Memory usage: 0 MB.  
[vtkSphereFromPoint] Spheres computed in 0.00740409 s.  
[vtkSphereFromPoint] Memory usage: 0 MB.  
[vtkSphereFromPoint] Spheres computed in 0.0150149 s.  
[vtkSphereFromPoint] Memory usage: 0 MB.  
[Common] Welcome!  
[vtkTopologicalSimplification] Scalar field simplified in 0.0003881 s. (24 threads)  
[vtkTopologicalSimplification] Memory usage: 0 MB.  
[Common] Welcome!  
[vtkScalarFieldCriticalPoints] Starting computation on field 'Elevation'...  
[ScalarFieldCriticalPoints] 5 maxima.  
[ScalarFieldCriticalPoints] 9 saddle(s).  
[ScalarFieldCriticalPoints] 0 multi-saddle(s).  
[ScalarFieldCriticalPoints] 6 minima.  
[ScalarFieldCriticalPoints] Data-set (69827 vertices) processed in 0.0316131 s.  
[vtkScalarFieldCriticalPoints] Memory usage: 0.00125 MB.  
[vtkSphereFromPoint] Spheres computed in 0.0126598 s.  
[vtkSphereFromPoint] Memory usage: 0 MB.  
[vtkSphereFromPoint] Spheres computed in 0.00694799 s.  
[vtkSphereFromPoint] Memory usage: 0.382812 MB.  
[Common] Welcome!  
[vtkContourForests] Launching computation for field 'Elevation'...  
[ContourForests] Initialization : 4.72869e-05  
[ContourForests] Sort scalars (-mirror) : 0.00710797  
[ContourForests] Interface and overlap init. : 0.00518107  
[ContourForests] Parallel allocations : 0.001382  
[ParallelBuild] Merge Tree 0 constructed in : 0.031607  
[ContourForests] ParallelBuild : 3.09944e-06  
[ContourForests] Stitch : 0.00048182  
[ContourForests] Create Contour tree : 0.0458591  
Tree computed ...  
[vtkContourForests] Topological skeleton built in 0.065634s :  
[vtkContourForests] Arc - Resolution: 20  
[vtkContourForests] Smoothing: 15  
[vtkContourForests] RegionType: true  
[vtkContourForests] Segmentation2D: true  
[vtkContourForests] Memory usage: 1.385 MB.  
[vtkSphereFromPoint] Spheres computed in 0.00801682 s.  
[vtkSphereFromPoint] Memory usage: 0.01215 MB.  
[vtkSphereFromPoint] Spheres computed in 0.0453 MB.
```

Properties (Tube):

- Scalars: Elevation
- Vectors: (empty)
- Number of Sides: 6
- Clipping:
- Radius: 1.0
- Vary Radius: Off
- Radius Factor: 10
- Use Default Normal:
- Default Normal: 0 0 1
- Representation: Surface
- Coloring: Elevation
- Scalar Coloring: Map Scalars, Interpolate Scalars Before Mapping
- Styling: Opacity: 1, Point Size: 2, Line Width: 1
- Lighting: Interpolation: Gouraud
- Specular Color: Specular Power: 100

Line Chart: Number of Pairs (all pairs) vs Persistence

Persistence	Number of Pairs
0.001	100
0.01	100
0.1	80
1	40
10	10
100	2

Scatter Plot: Birth vs Death

Birth	Death
0	100
20	50
40	40
60	90
80	80

13:26 / 14:22

CC HD YouTube

Morse Persistence Demo (MSC)

TTK usage tutorial - Morse persistence demo

MORE VIDEOS

6:41 / 16:03

CC HD YouTube

A few tips on Project 2

- Start today! The installation is going to take a while (4+ hours).
- Follow the demo closely, however pay attention to some differences in different versions of TTK
- Follow the reading materials for this week.



Thanks!

Any questions?

You can find me at: beiwang@sci.utah.edu

CREDITS

Special thanks to all people who made and share these awesome resources for free:

- ☐ Presentation template designed by [Slidesmash](#)
- ☐ Photographs by [unsplash.com](#) and [pexels.com](#)
- ☐ Vector Icons by [Matthew Skiles](#)

Presentation Design

This presentation uses the following typographies and colors:

Free Fonts used:

<http://www.1001fonts.com/oswald-font.html>

<https://www.fontsquirrel.com/fonts/open-sans>

Colors used

