Advanced Data Visualization

CS 6965

Fall 2019

Prof. Bei Wang Phillips

University of Utah

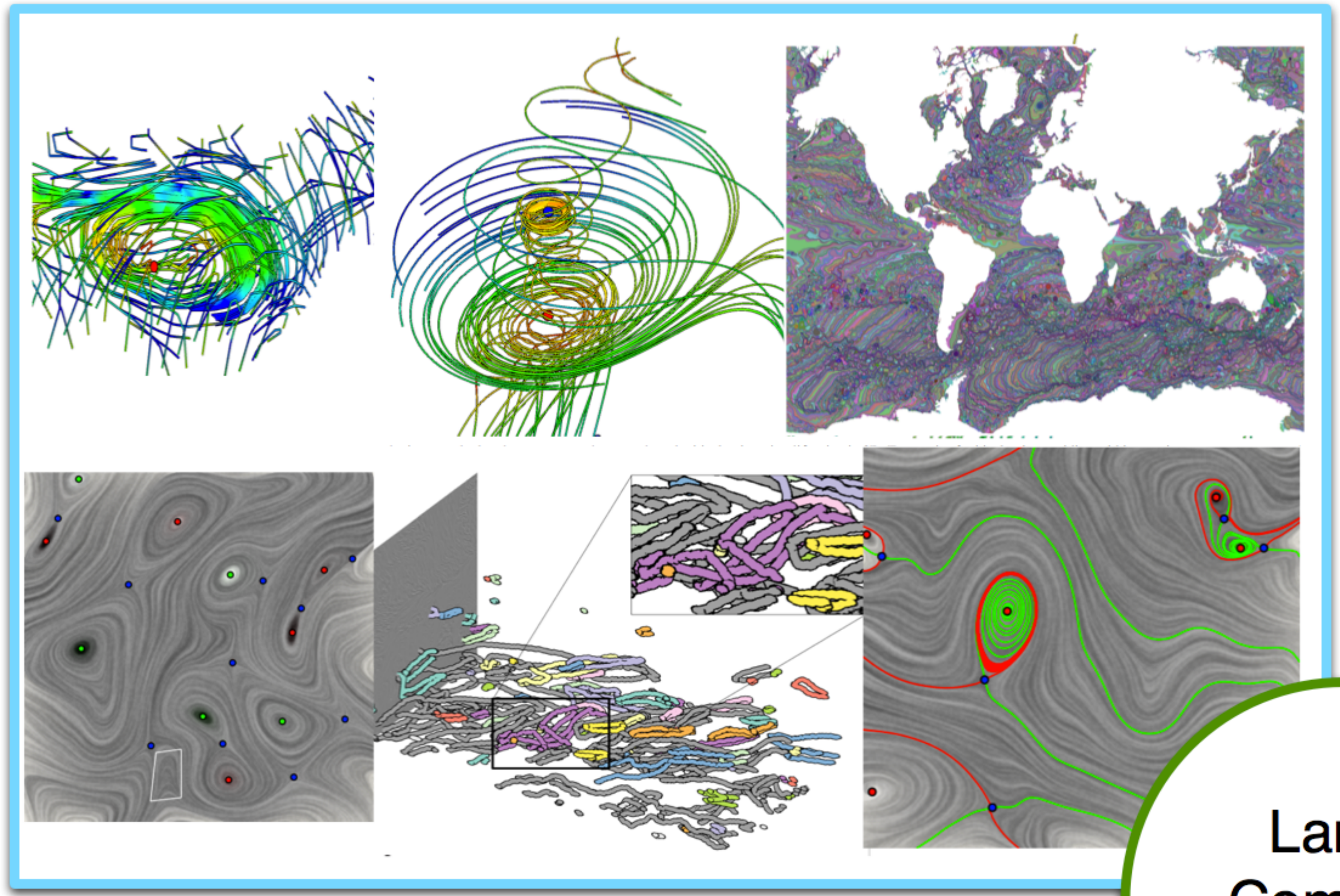
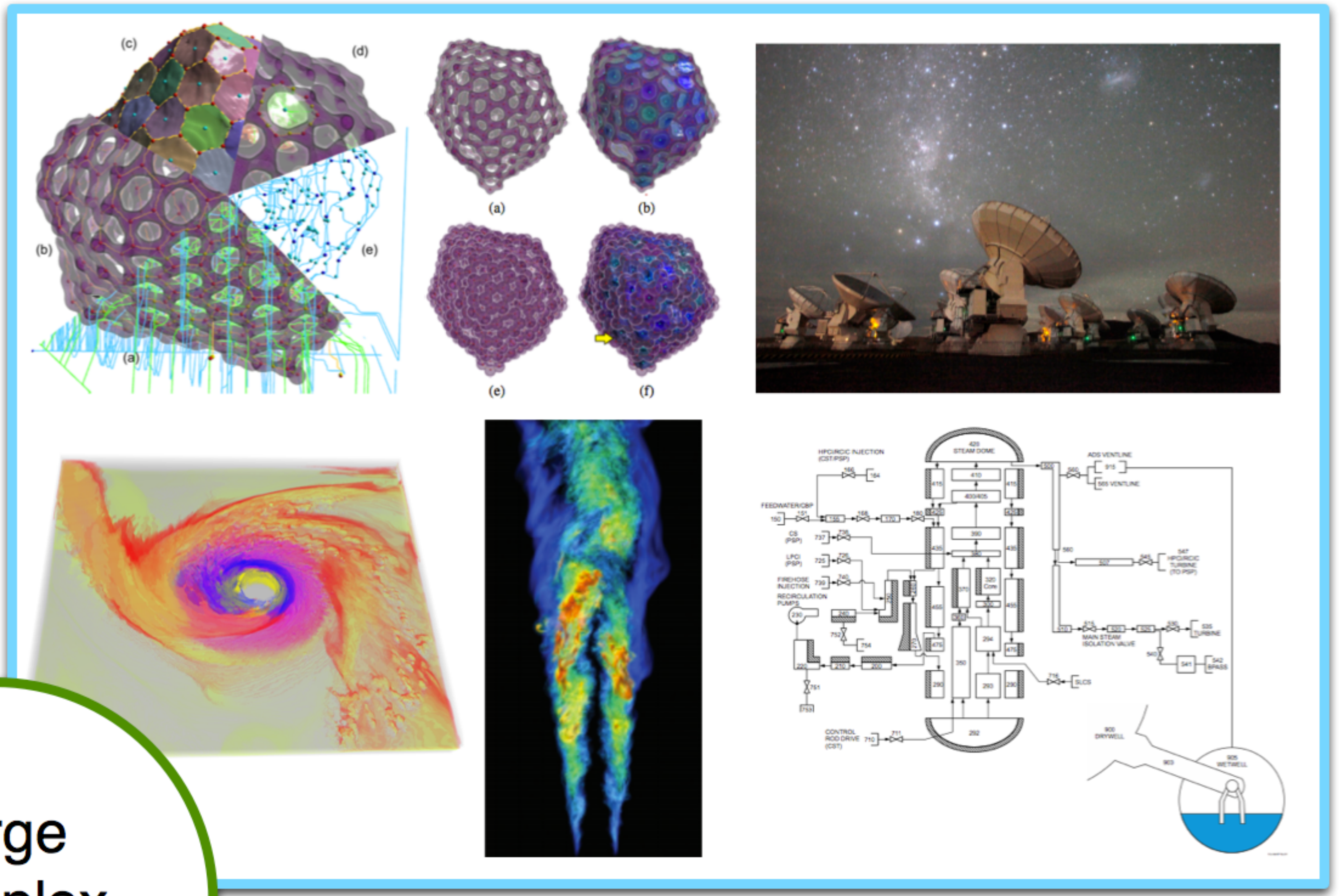
Lecture 24

Topological structures with applications

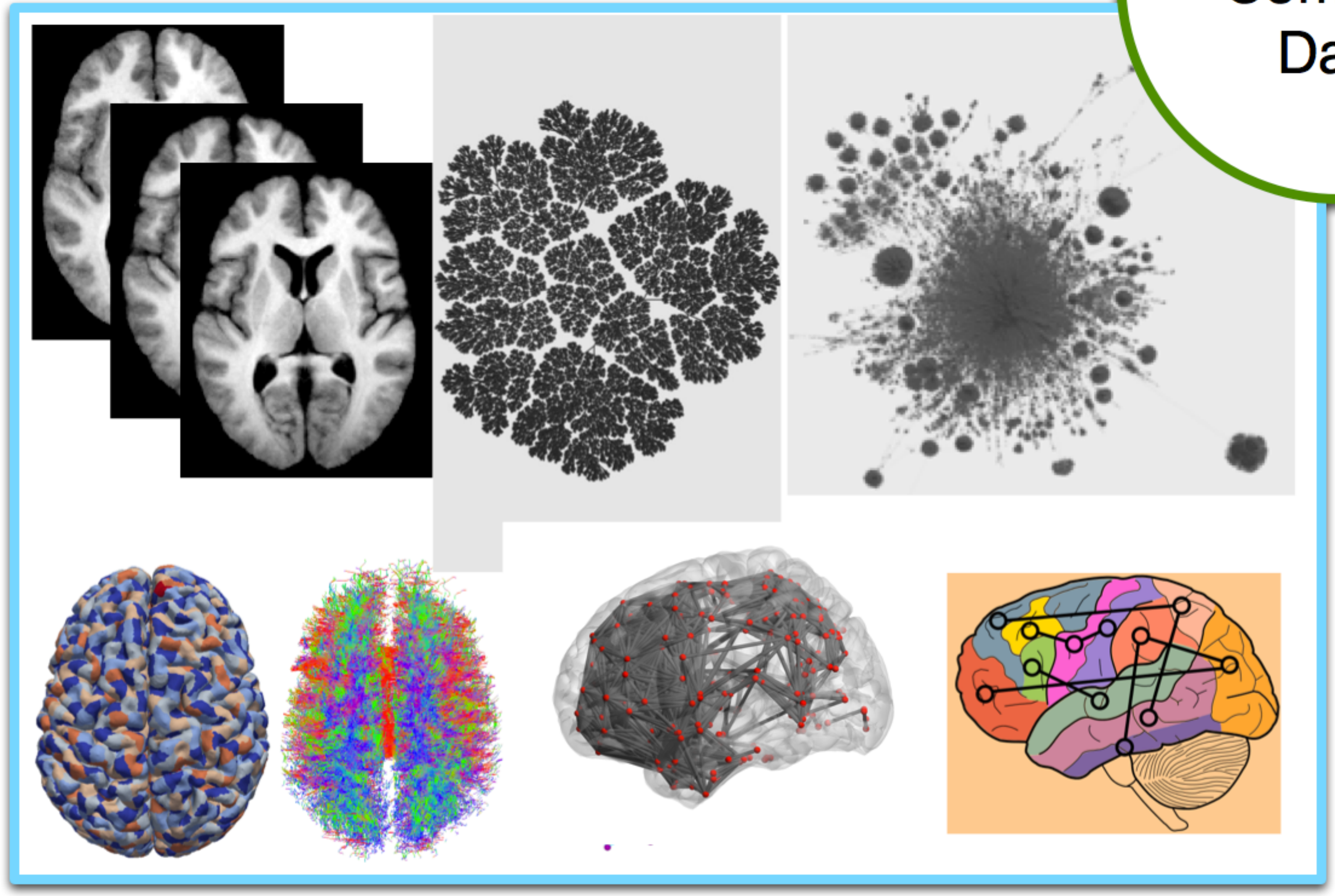
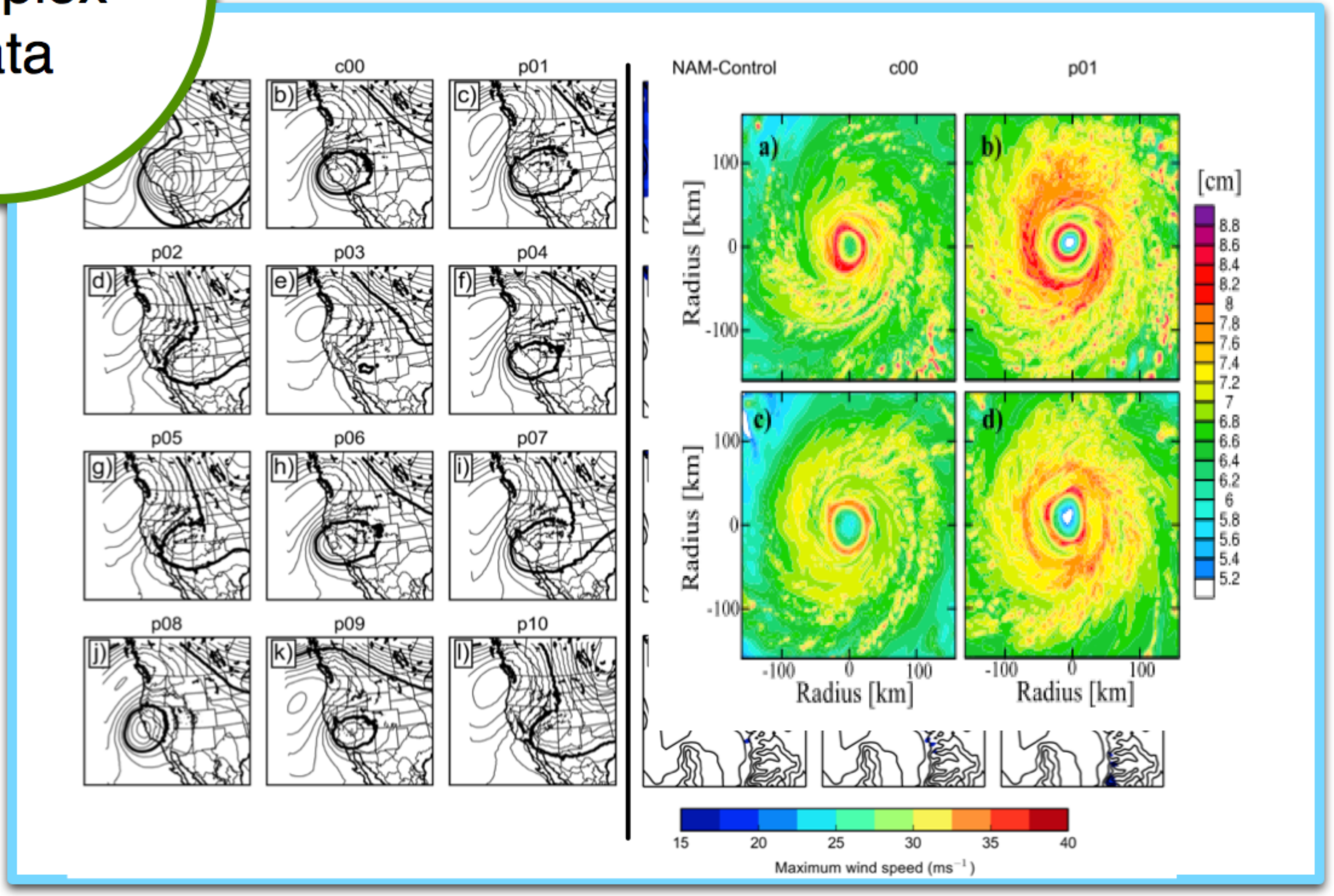
TOPO

Data-driven approach to TDA

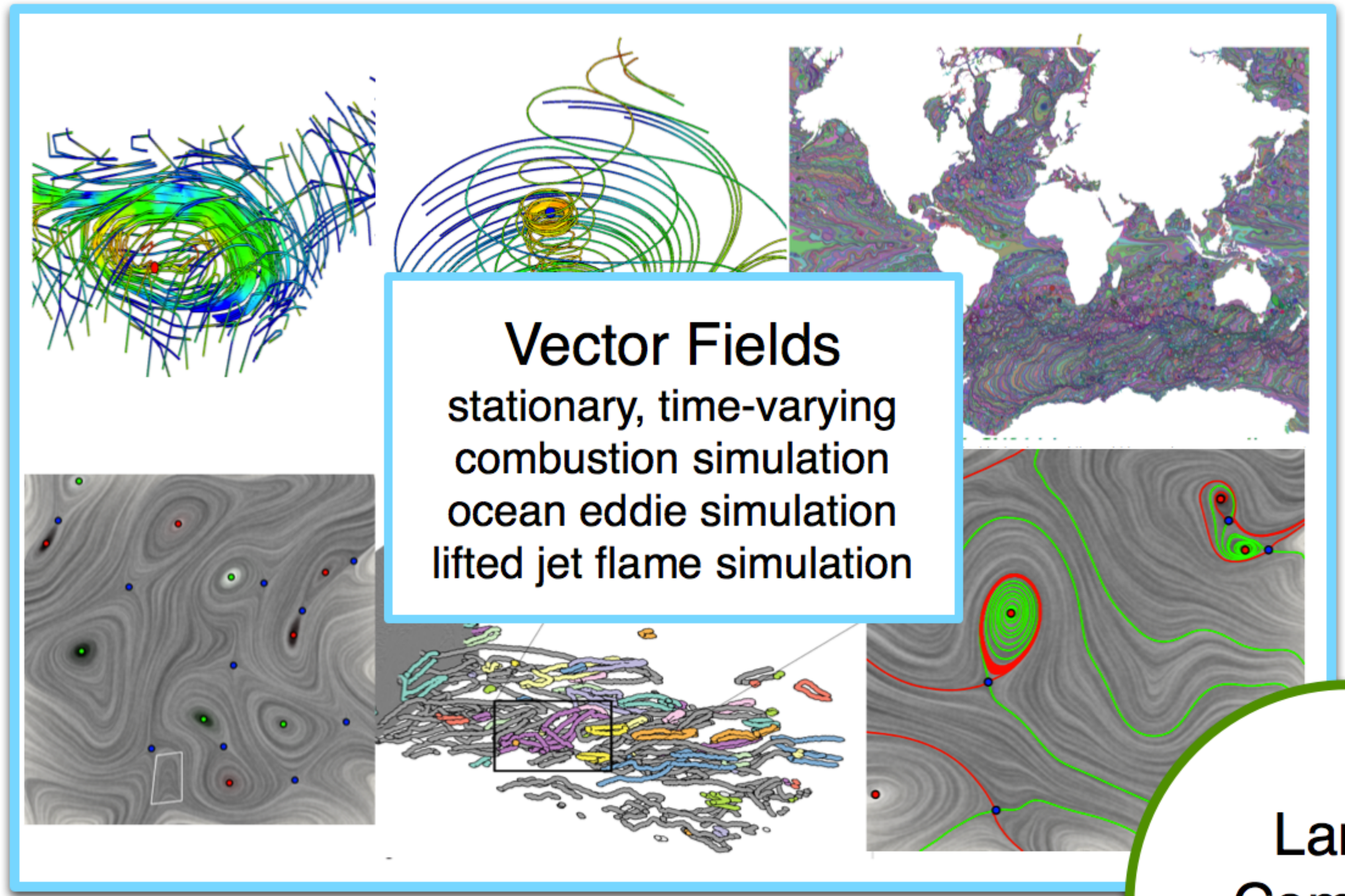
What is data?

Large Complex Data

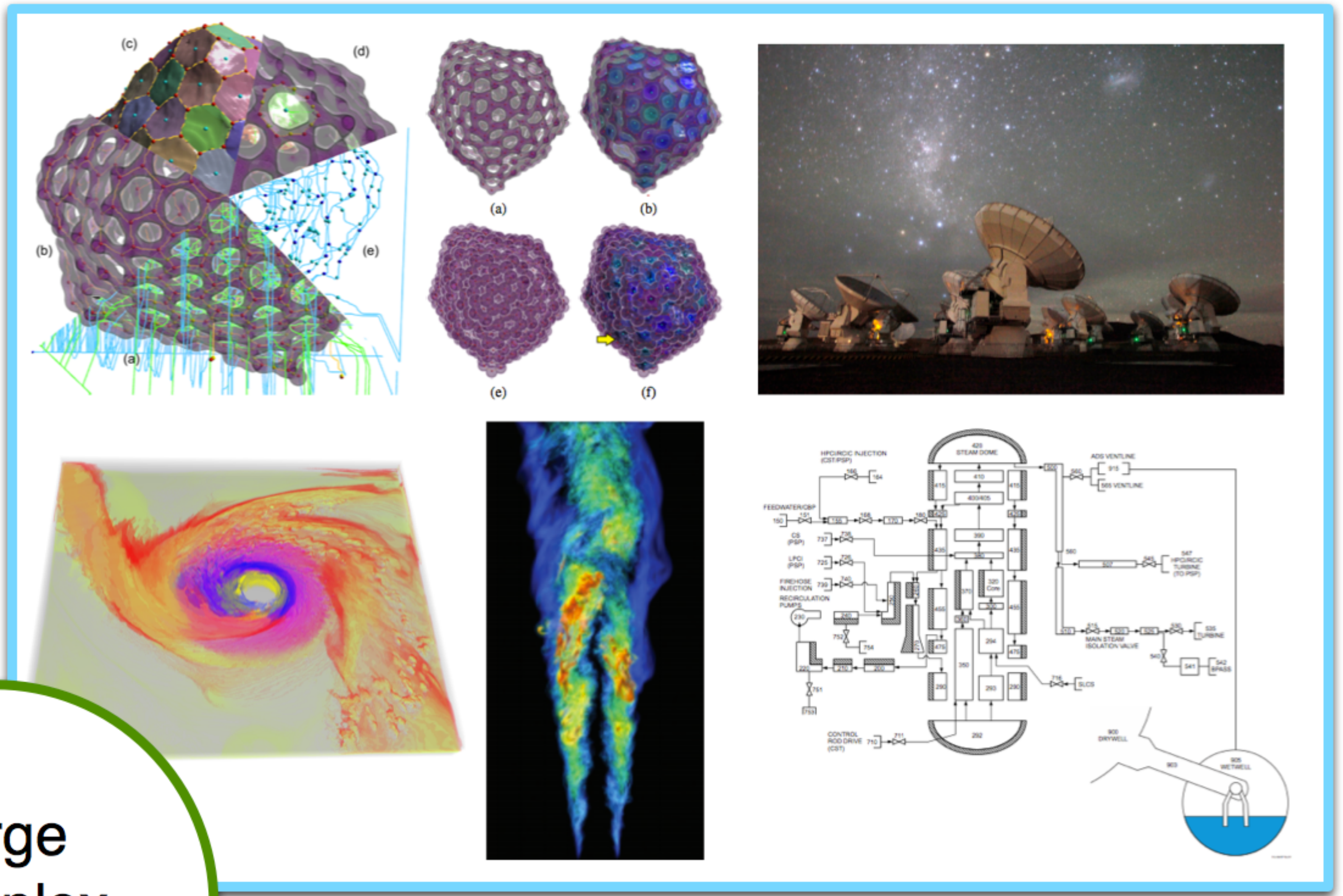



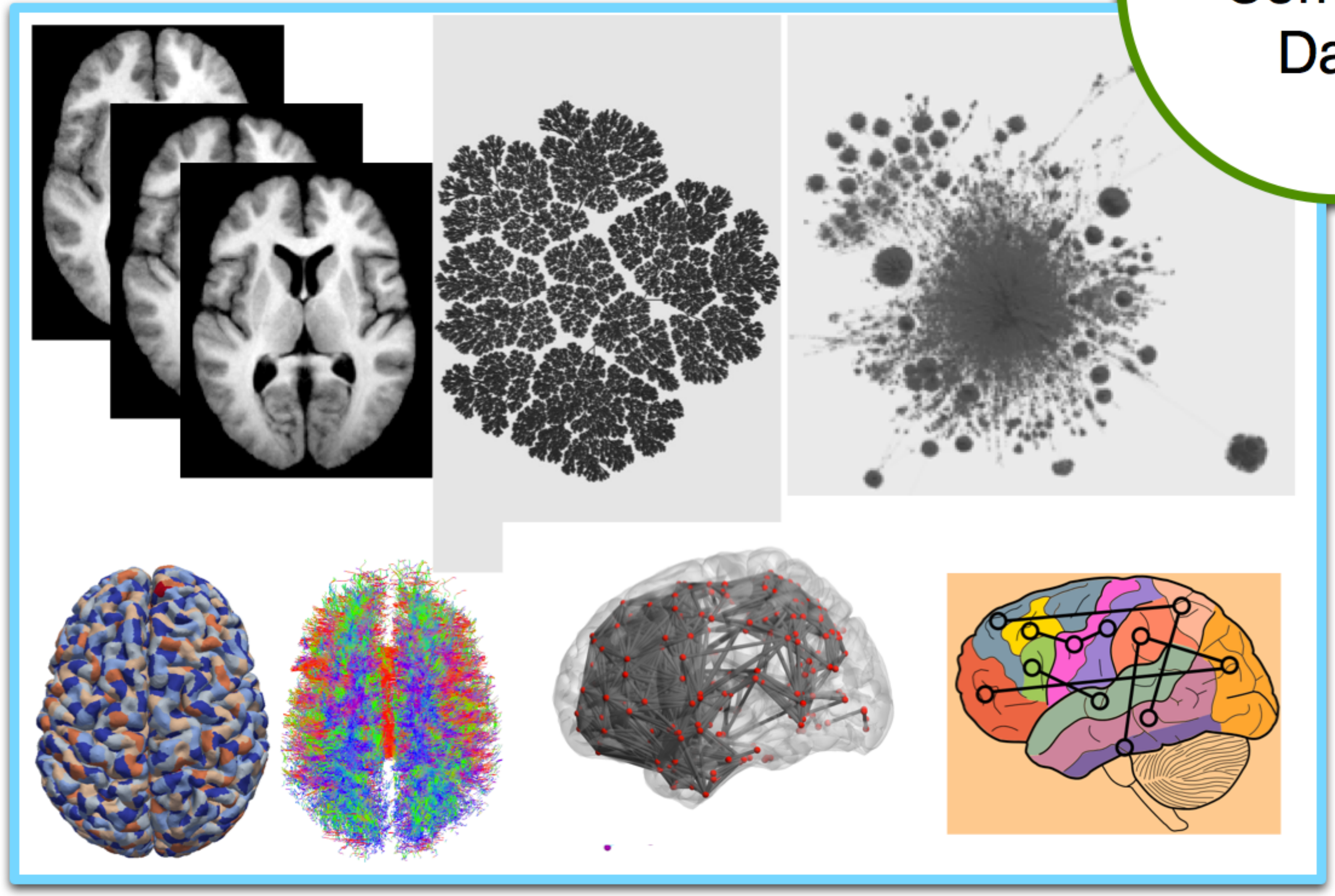
Large Complex Data

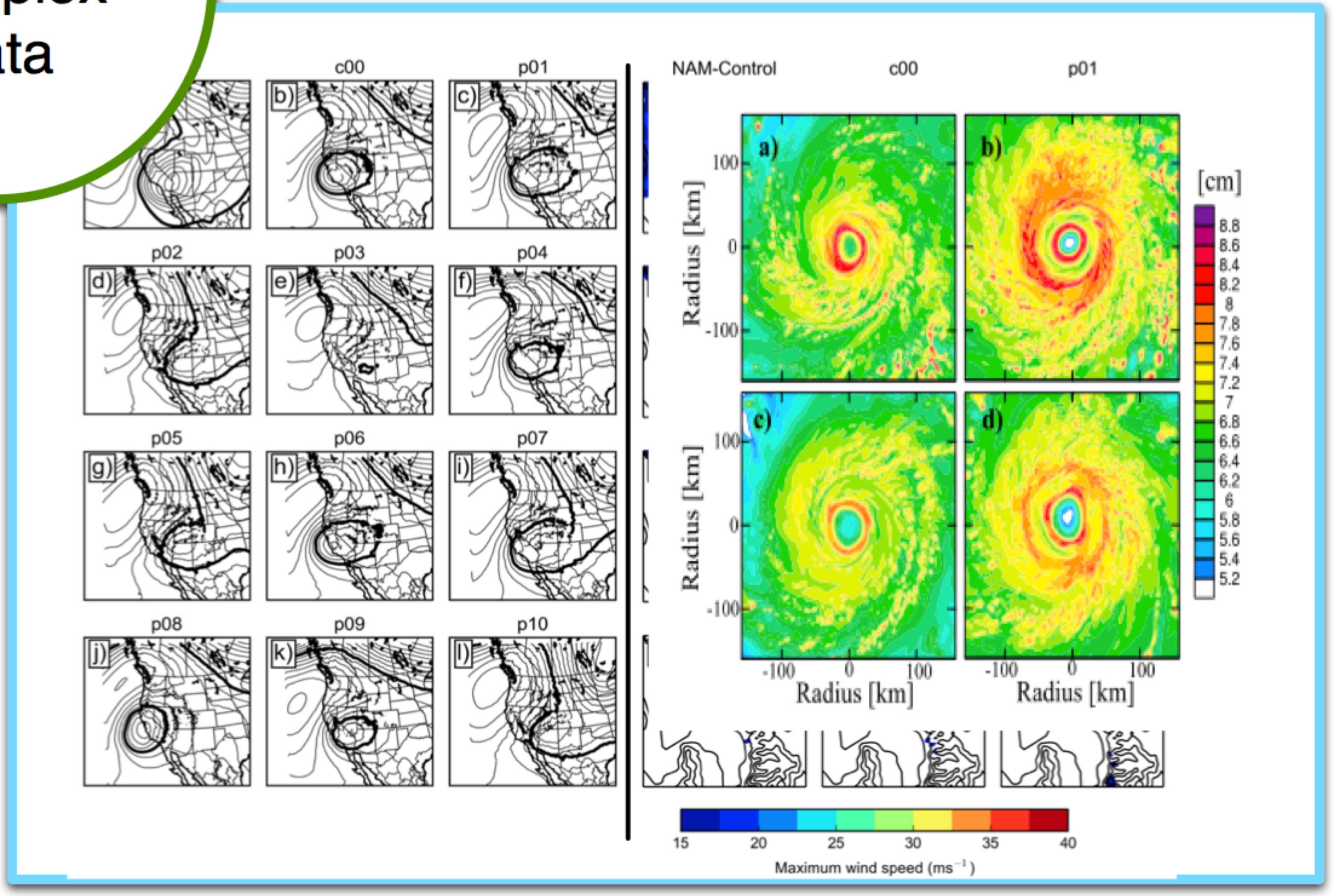


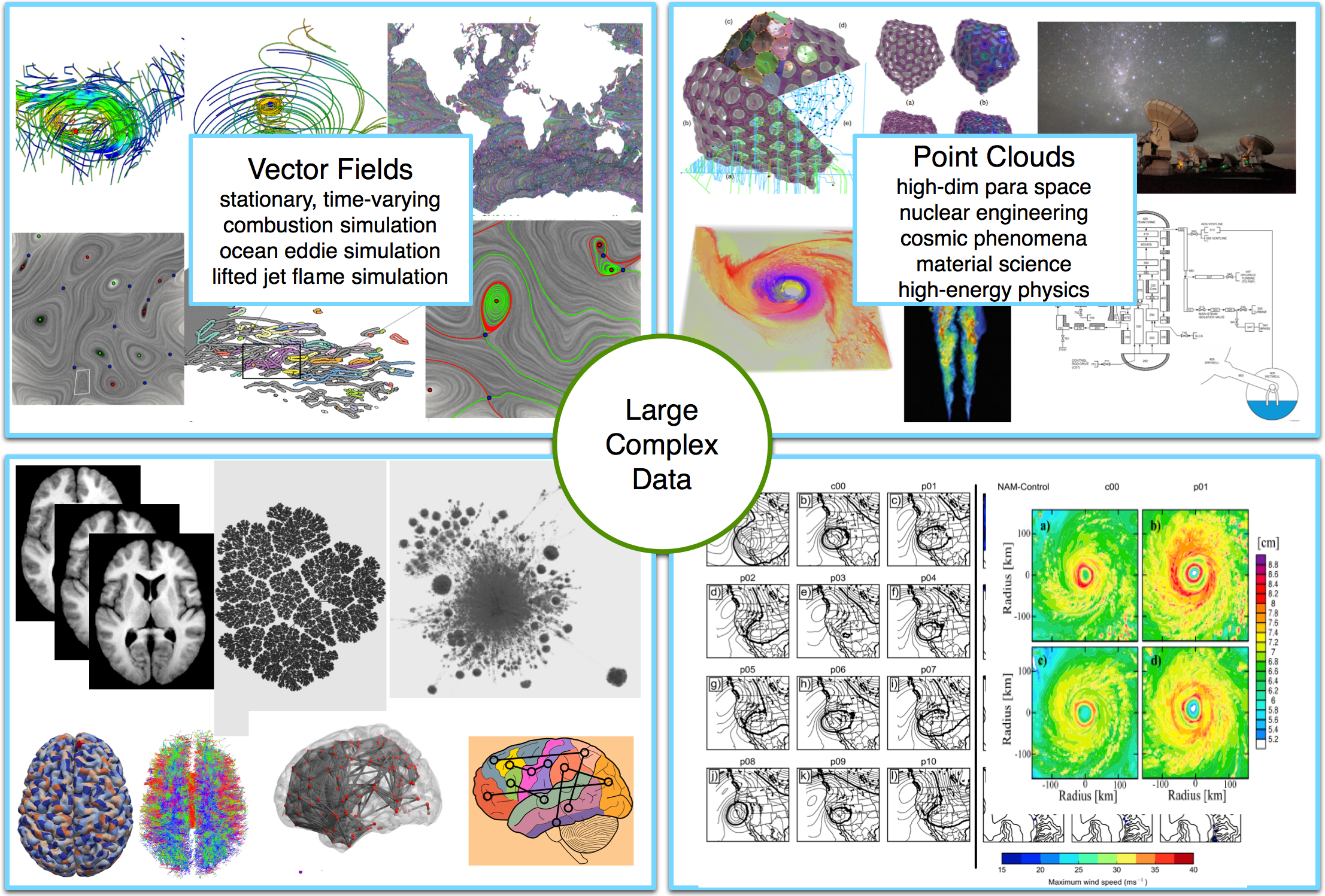
Vector Fields

stationary, time-varying
combustion simulation
ocean eddie simulation
lifted jet flame simulation









Vector Fields

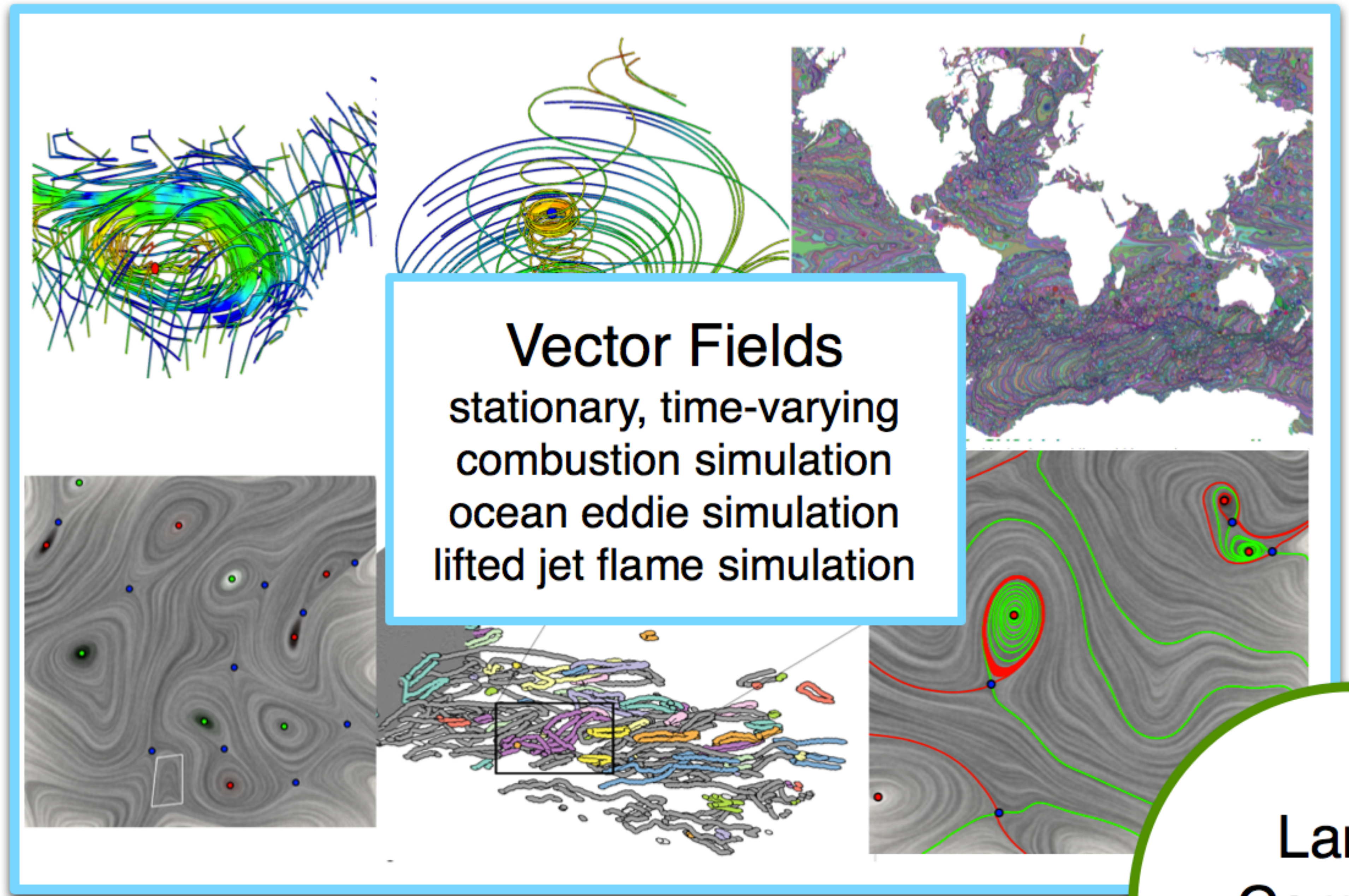
stationary, time-varying
combustion simulation
ocean eddie simulation
lifted jet flame simulation

Point Clouds

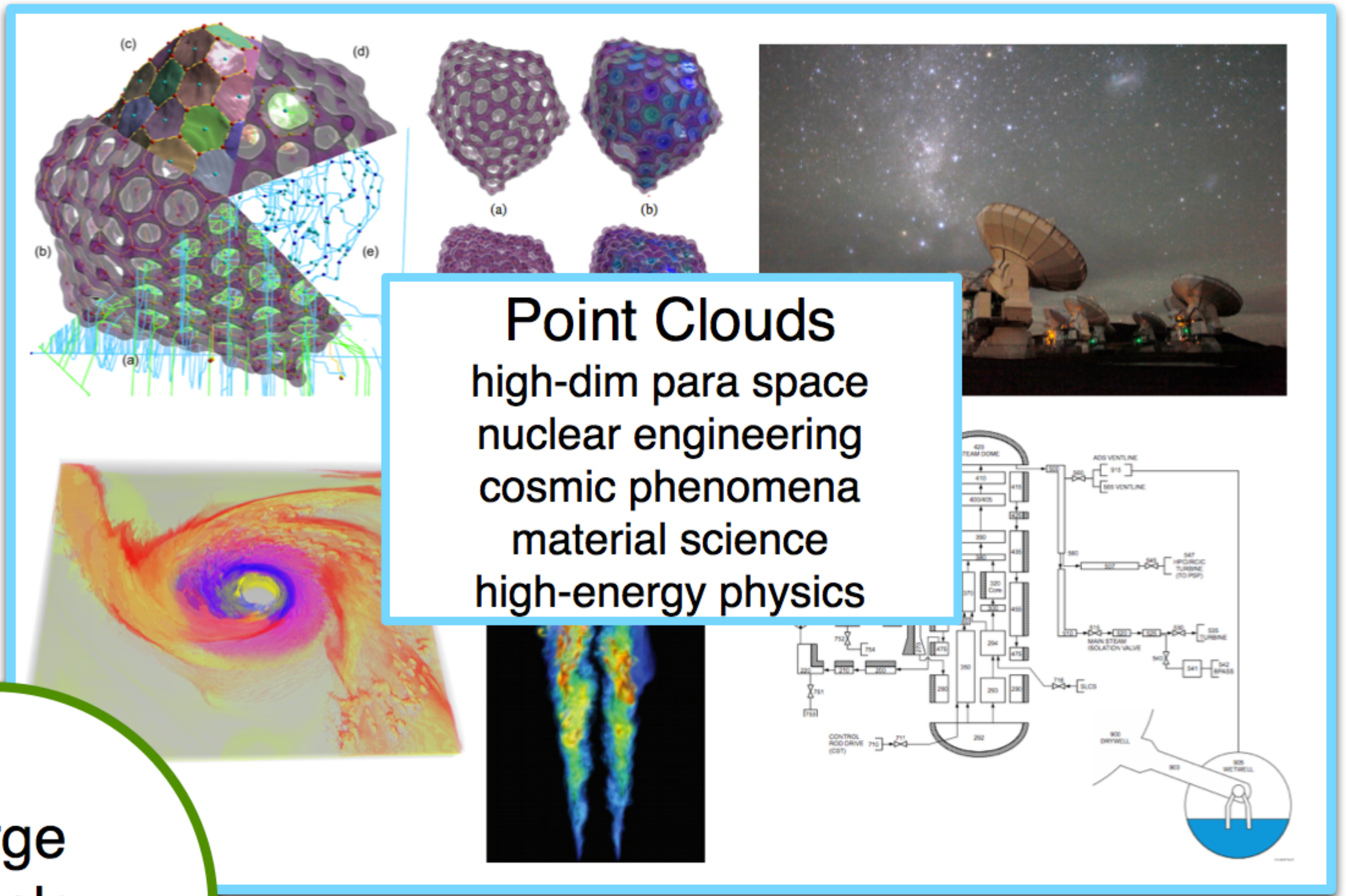
high-dim para space
nuclear engineering
cosmic phenomena
material science
high-energy physics

Large Complex Data

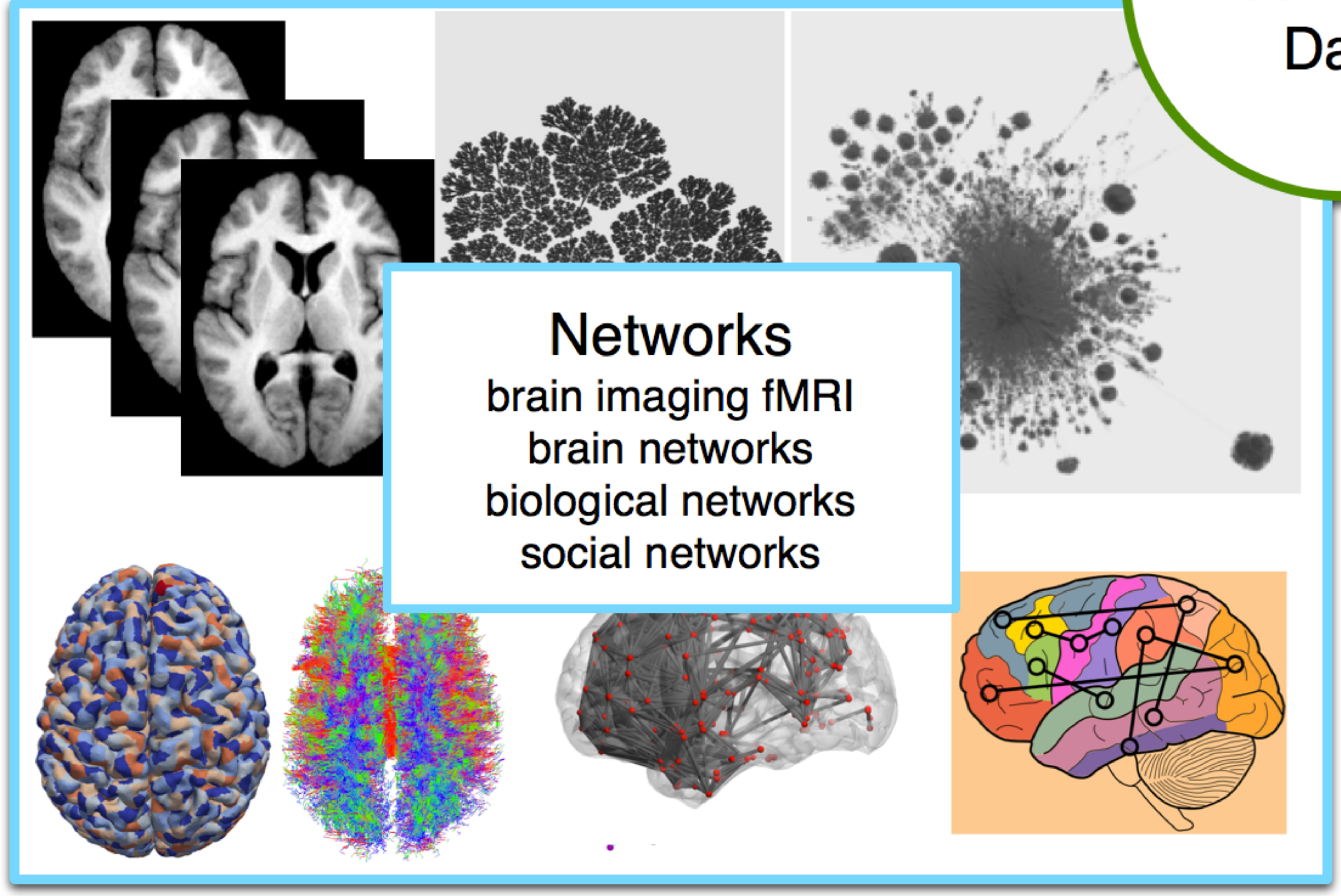
Large
Complex
Data



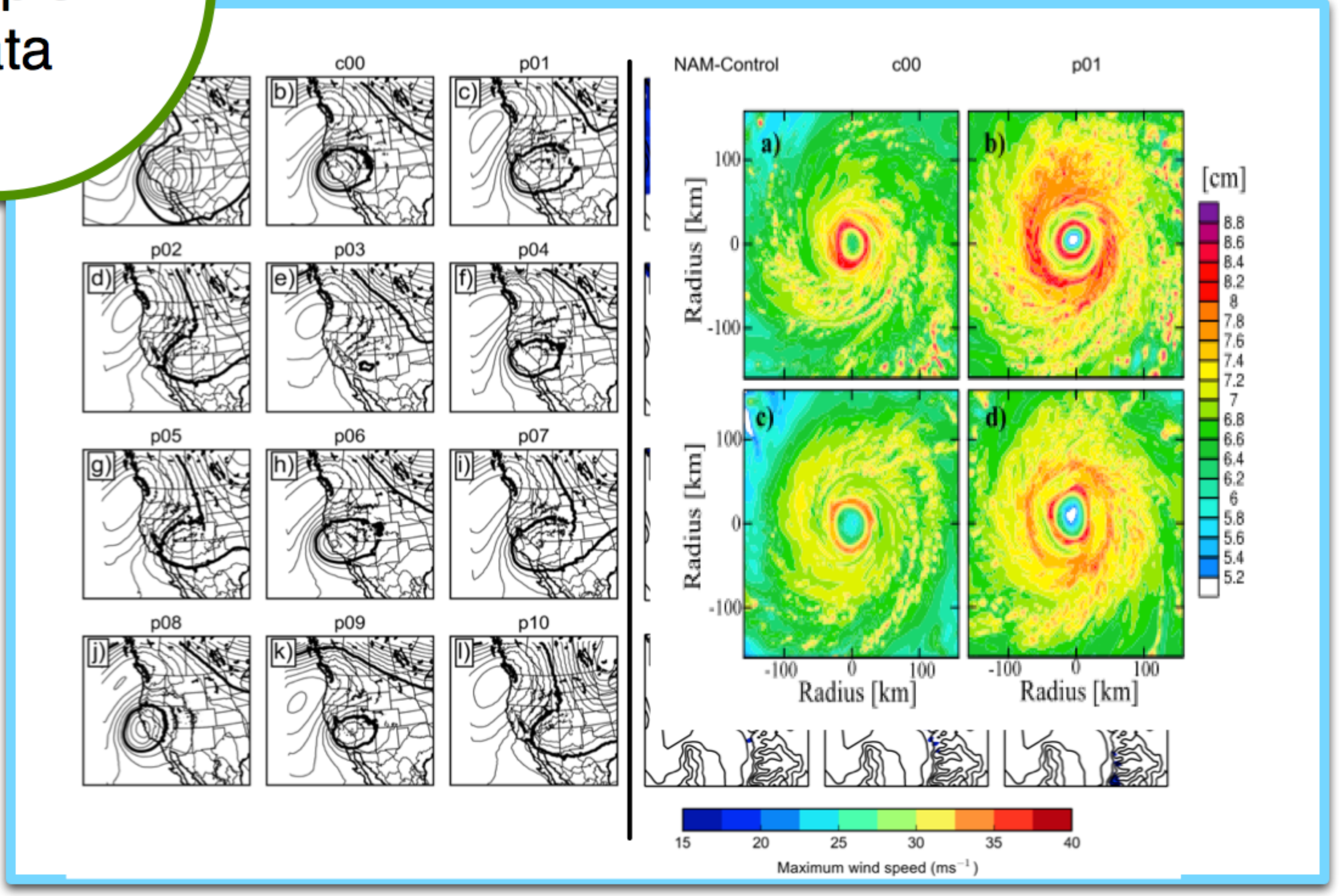
Vector Fields
stationary, time-varying
combustion simulation
ocean eddie simulation
lifted jet flame simulation



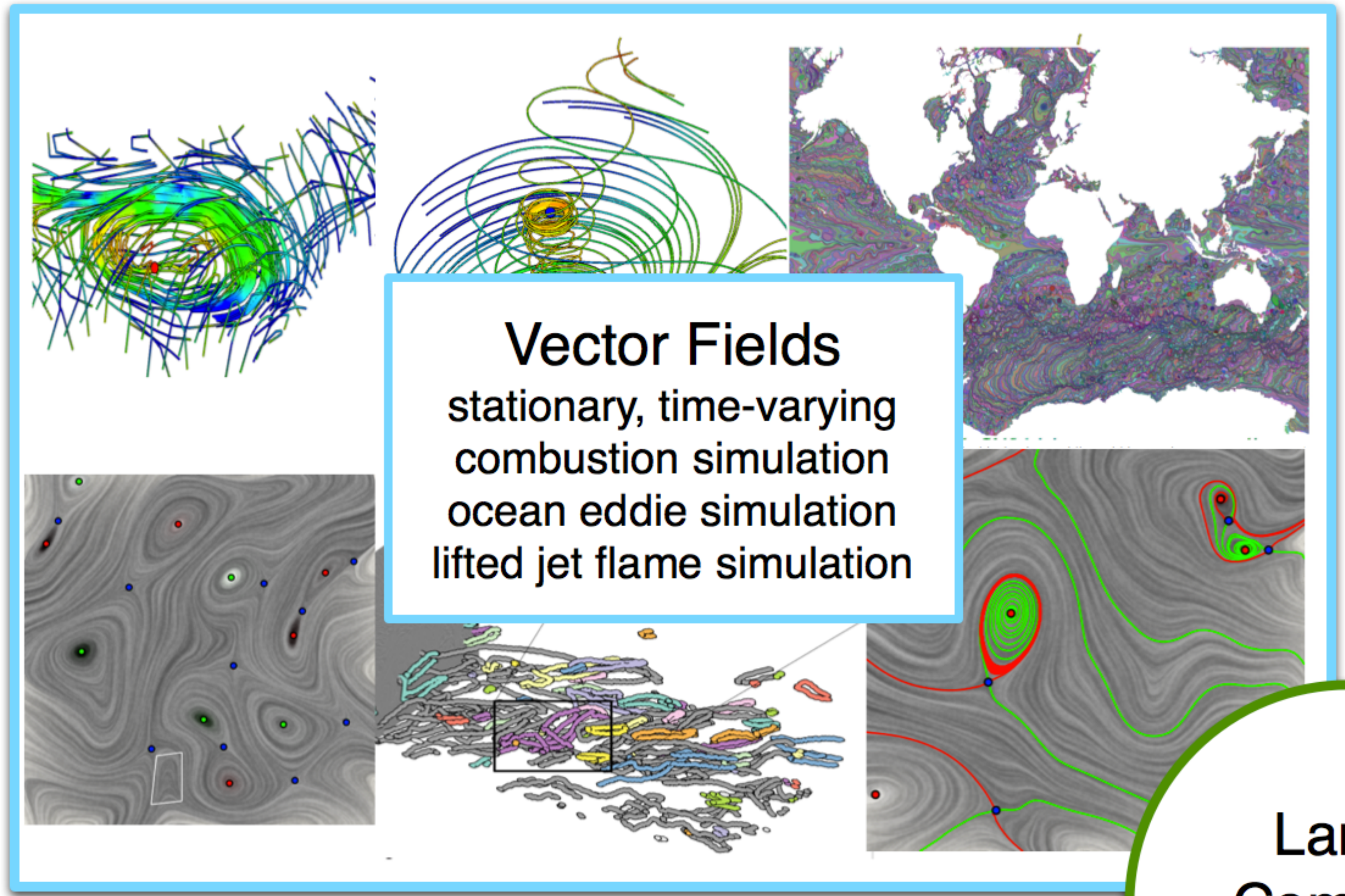
Point Clouds
high-dim para space
nuclear engineering
cosmic phenomena
material science
high-energy physics



Networks
brain imaging fMRI
brain networks
biological networks
social networks

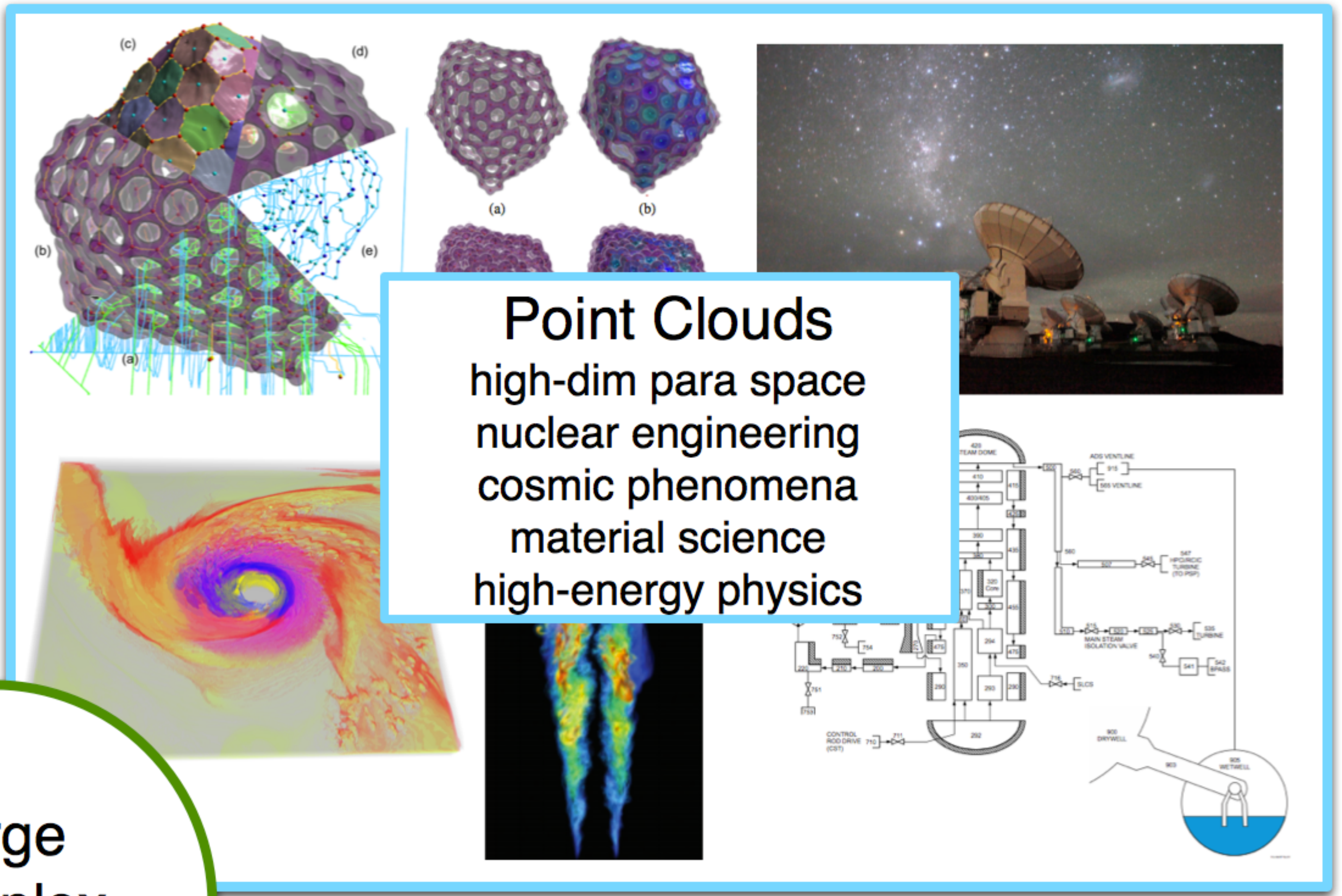


Large Complex Data



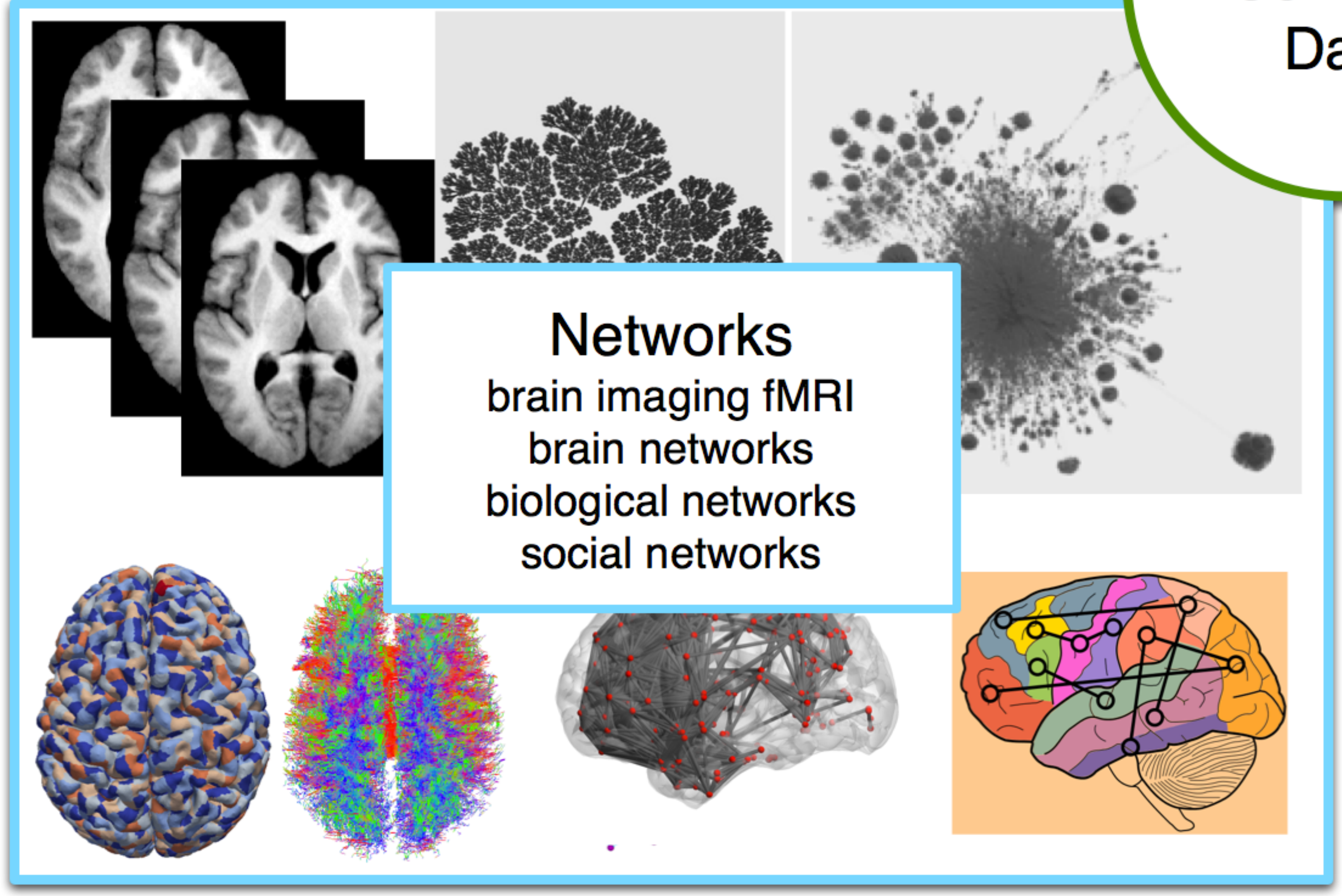
Vector Fields

stationary, time-varying
combustion simulation
ocean eddie simulation
lifted jet flame simulation



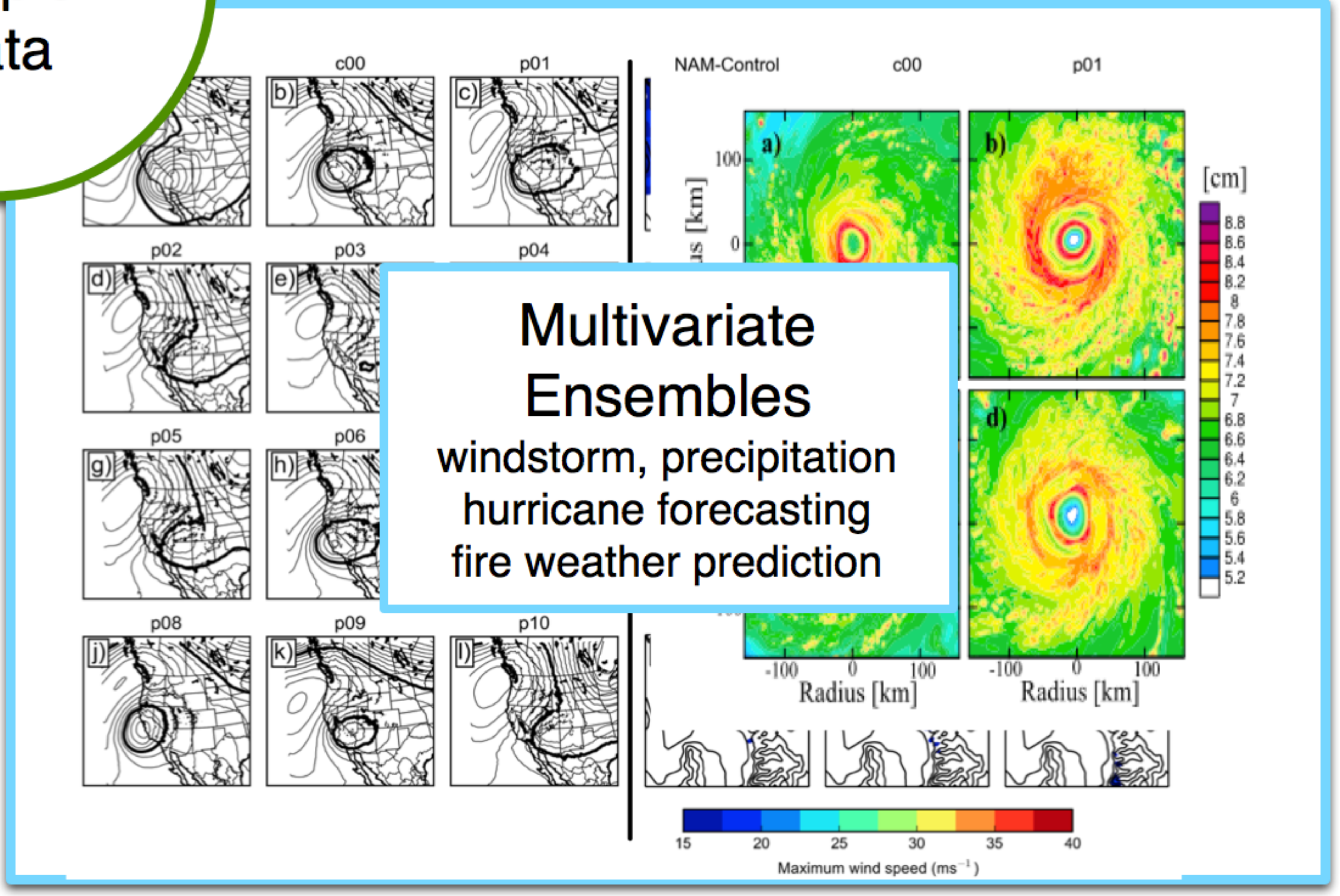
Point Clouds

high-dim para space
nuclear engineering
cosmic phenomena
material science
high-energy physics



Networks

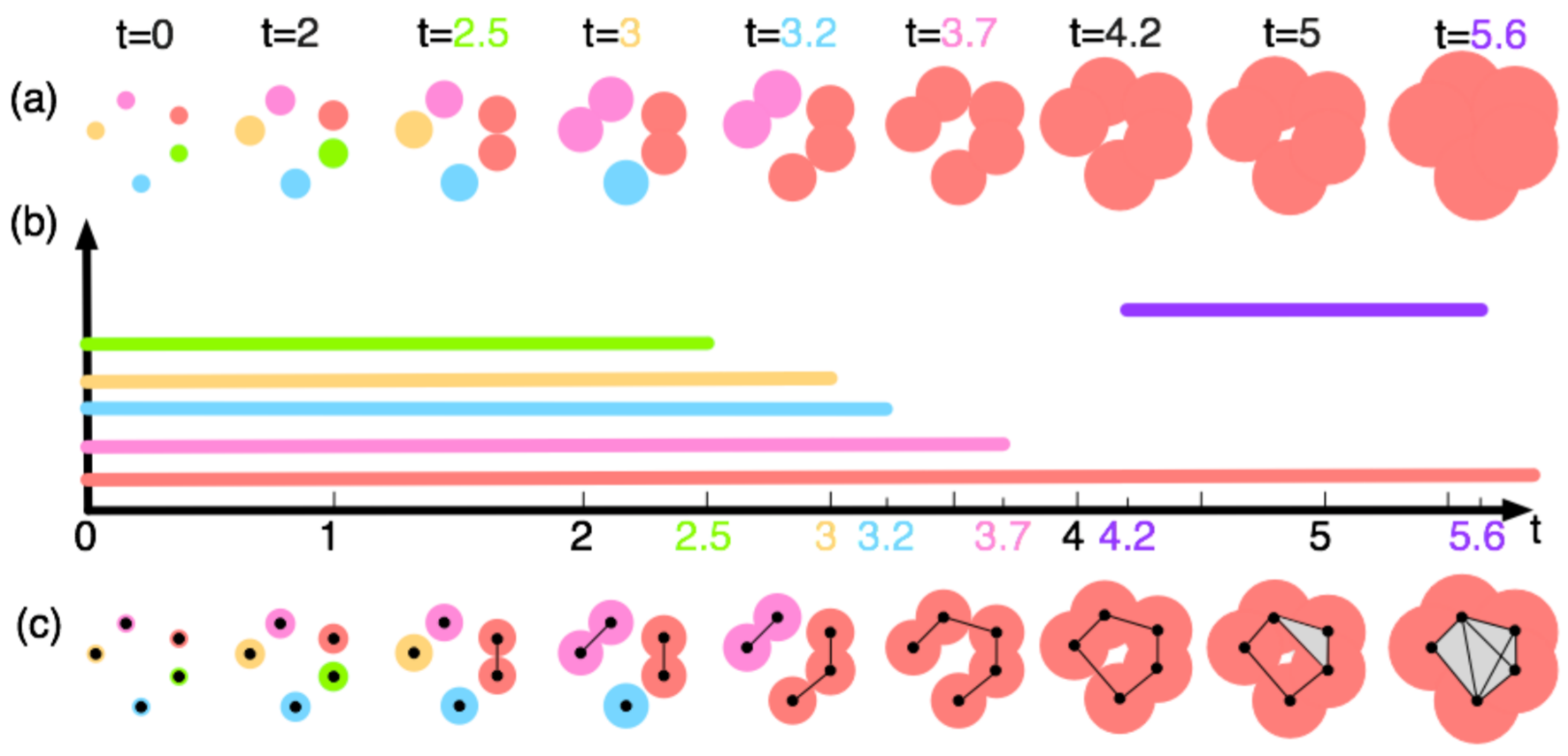
brain imaging fMRI
brain networks
biological networks
social networks



Multivariate Ensembles

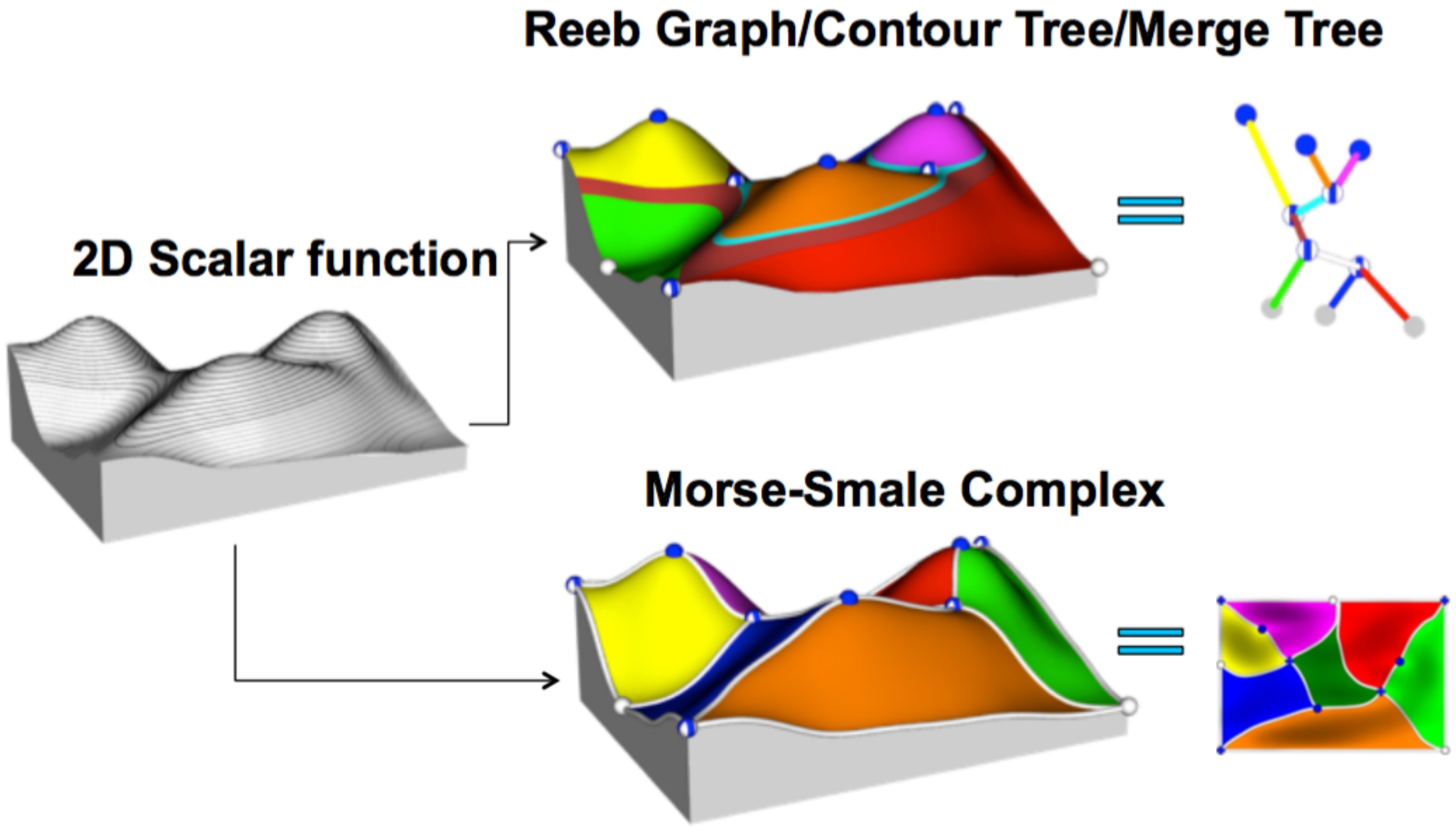
windstorm, precipitation
hurricane forecasting
fire weather prediction

Common tools in TDA and Vis



Persistent Homology: abstraction, compression, simplification

Topological Structures:
Contour Tree
Morse-Smale Complex (MSC)



Persistent Homology with Visualization

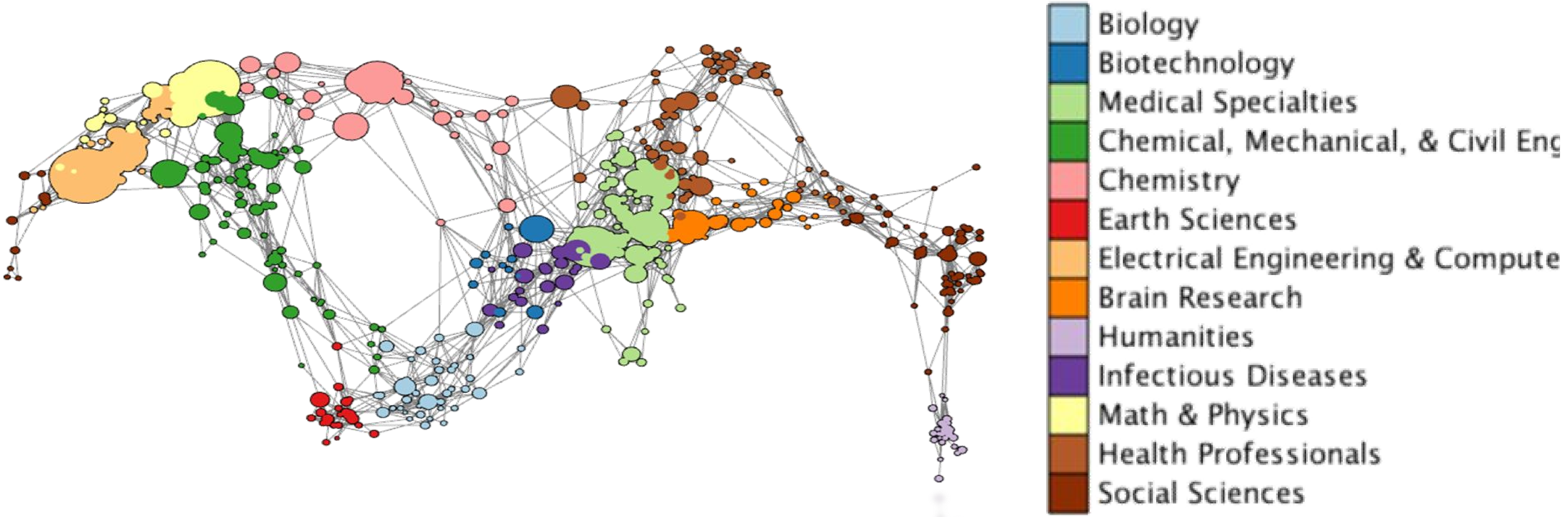
An application story

Case study 1:
A Map of Science Example

Map of Science?

554 subdiscipline belong to 13 scientific discipline

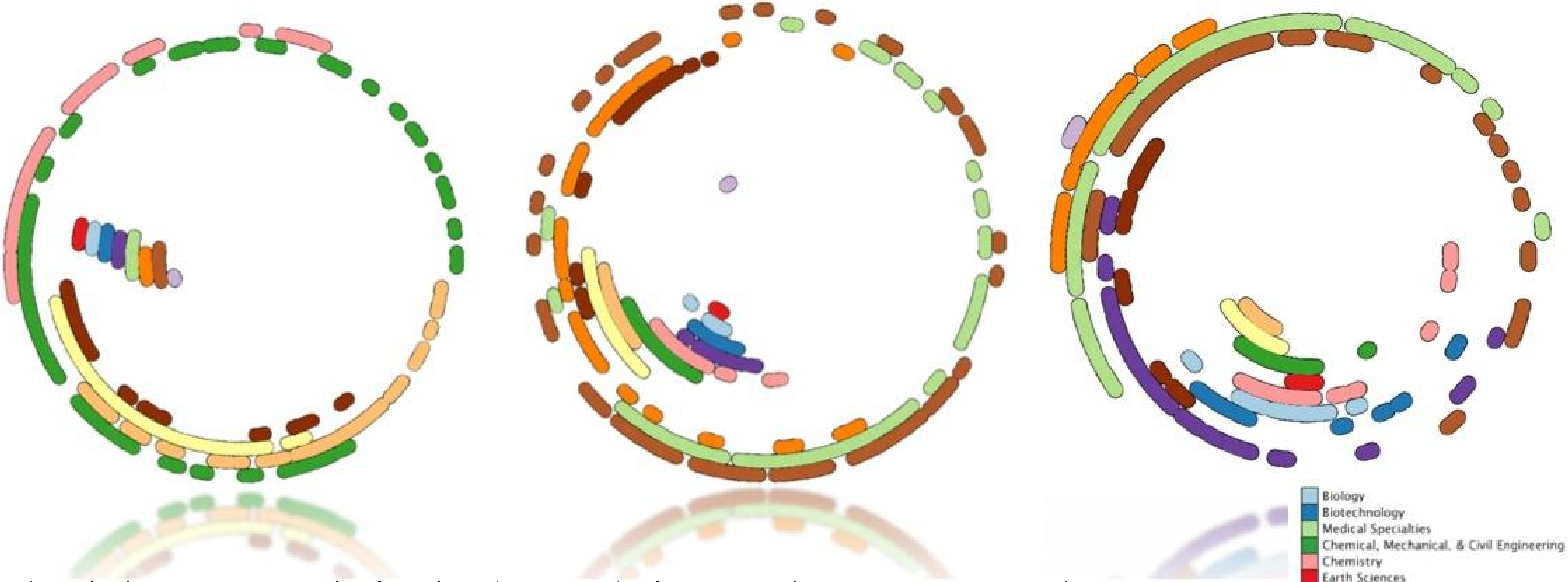
Map of Science



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?



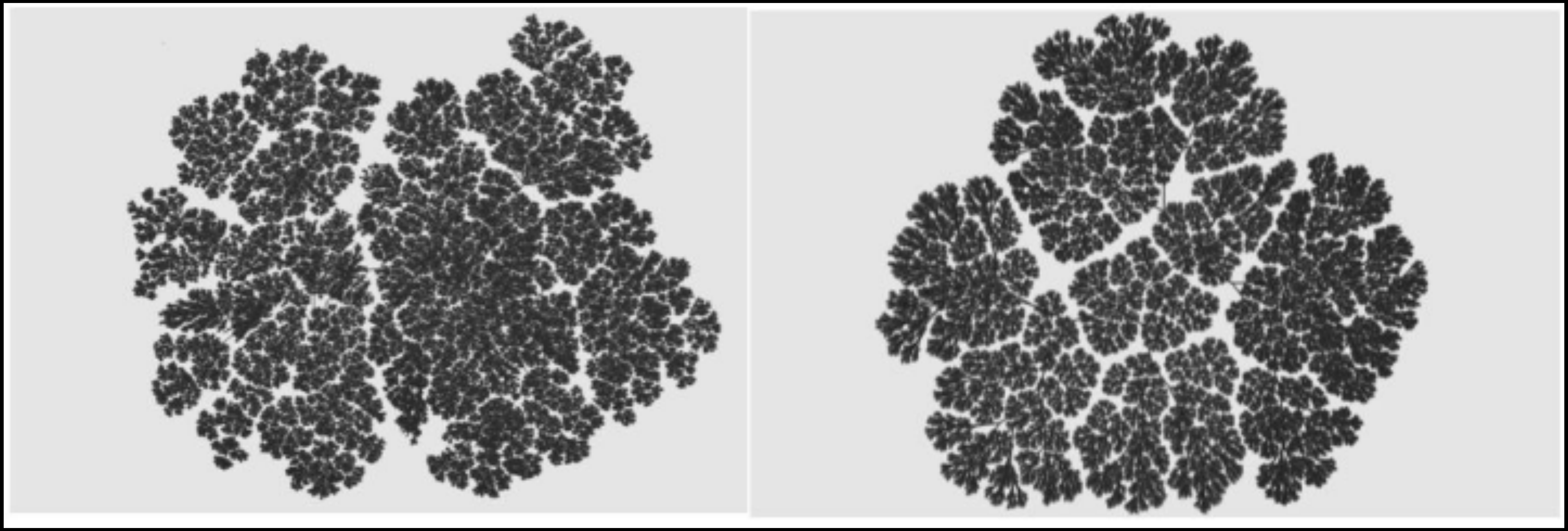
Three high persistence cycles found on the network of science – showing interconnectedness of specialties – left: core science and engineering, middle: healthcare, right: science (non-engineering)

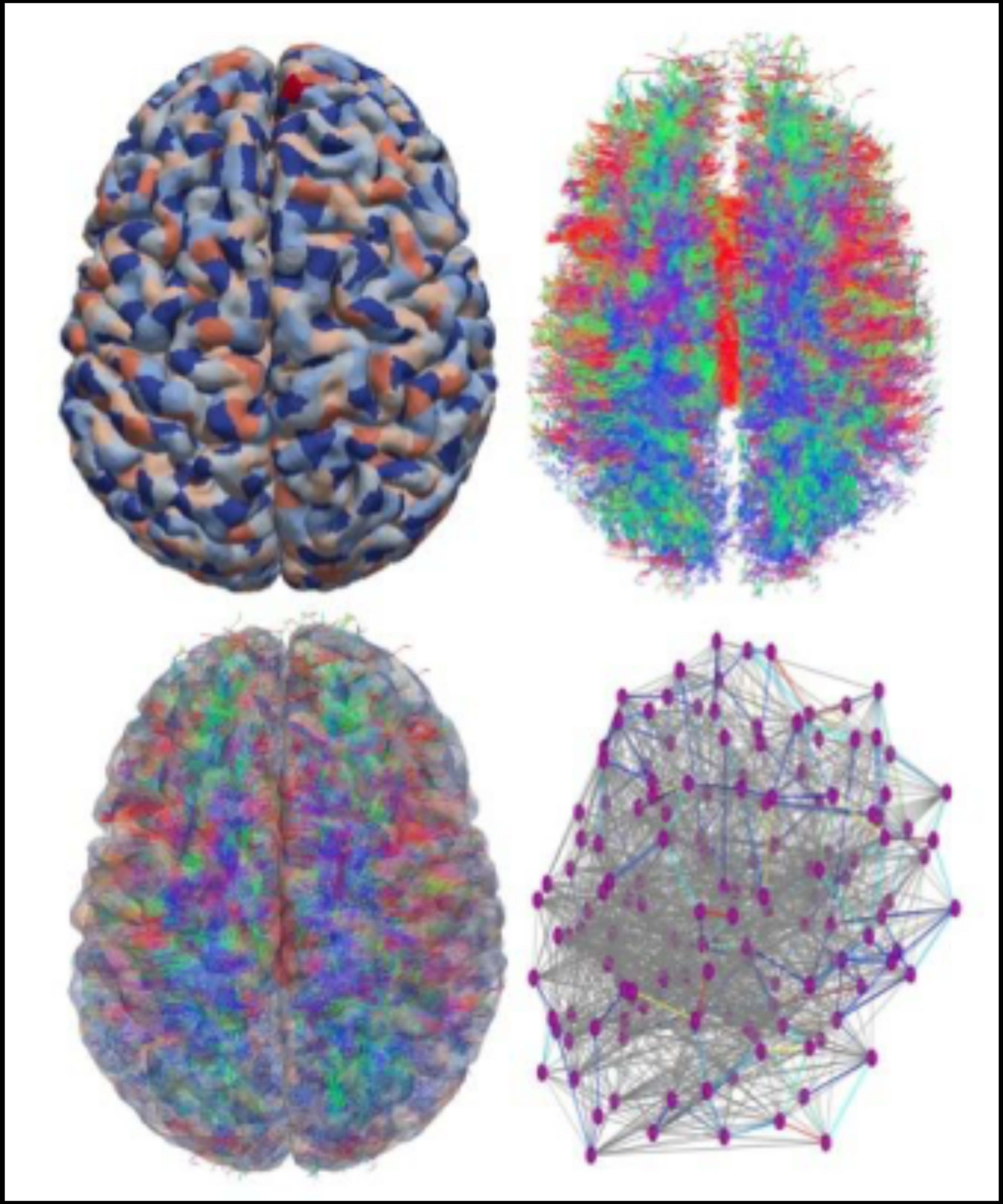
Bei Wang, Carlos Scheidegger, Paul Rosen, unpublished

Case study 2: Networks

Brain networks

Inadequate Network Visualization

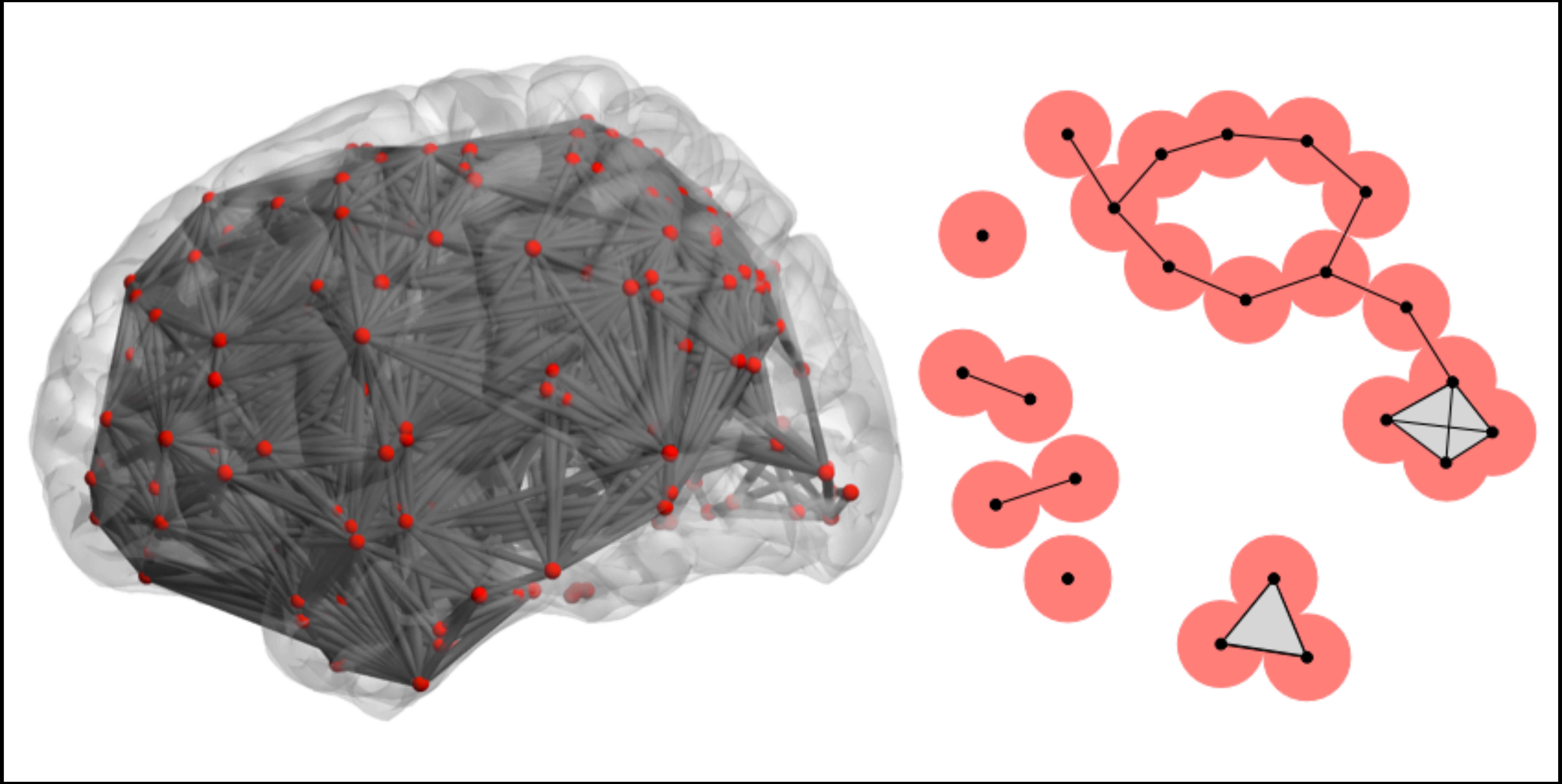




Brain Network Visualization

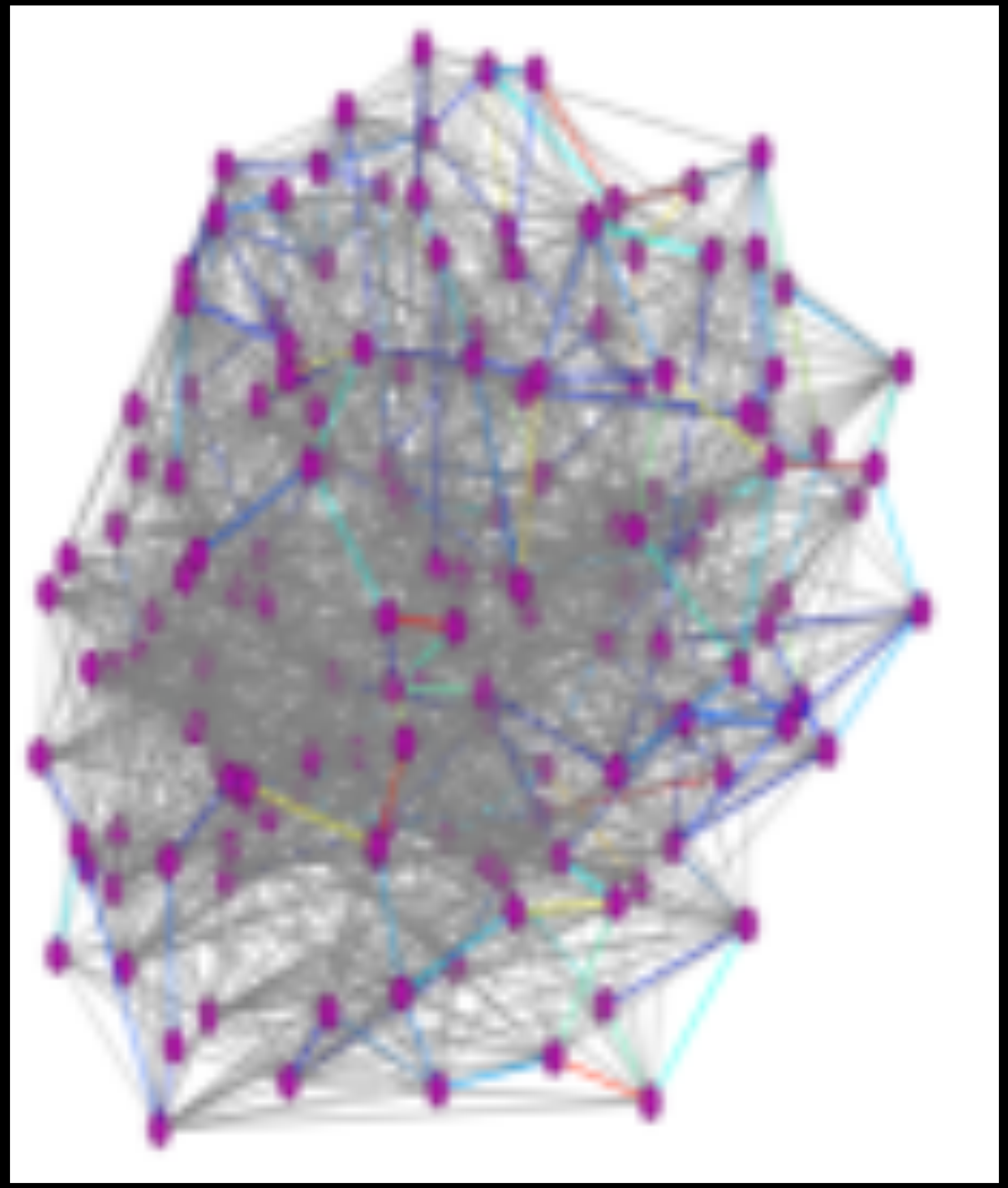
Avoid network hairballs
while preserving structure?

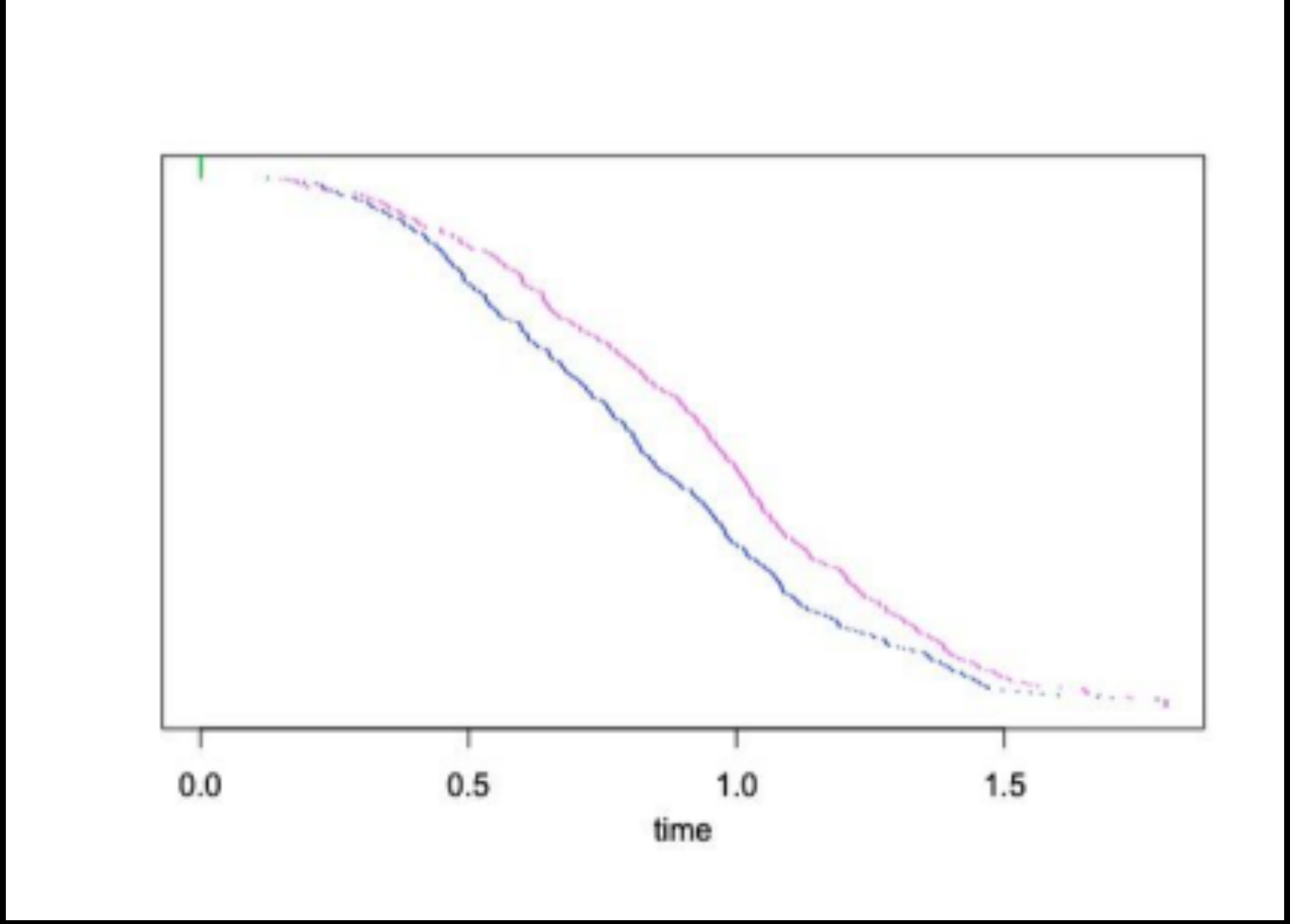
Topology and brain networks

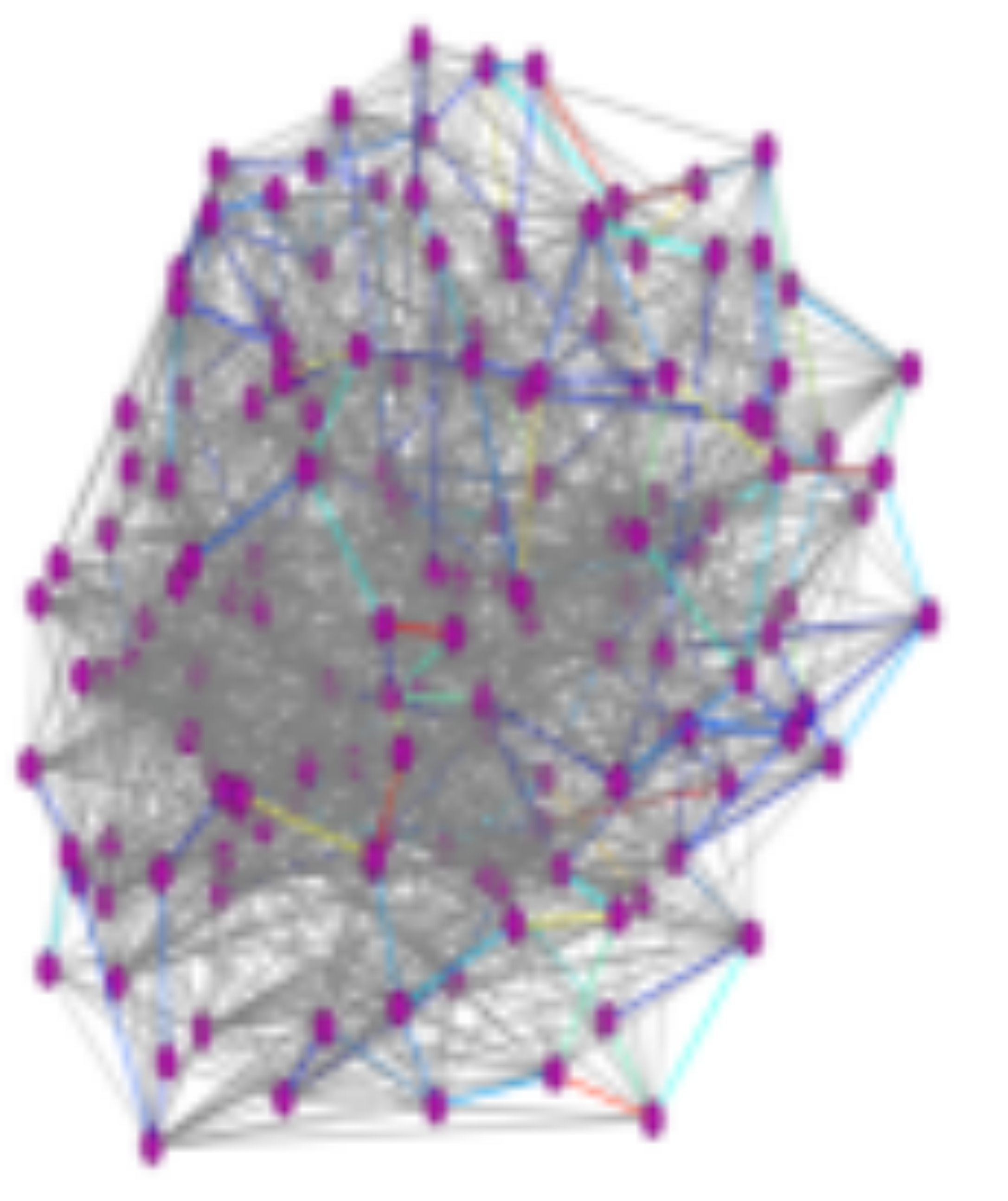


Autism Brain Networks

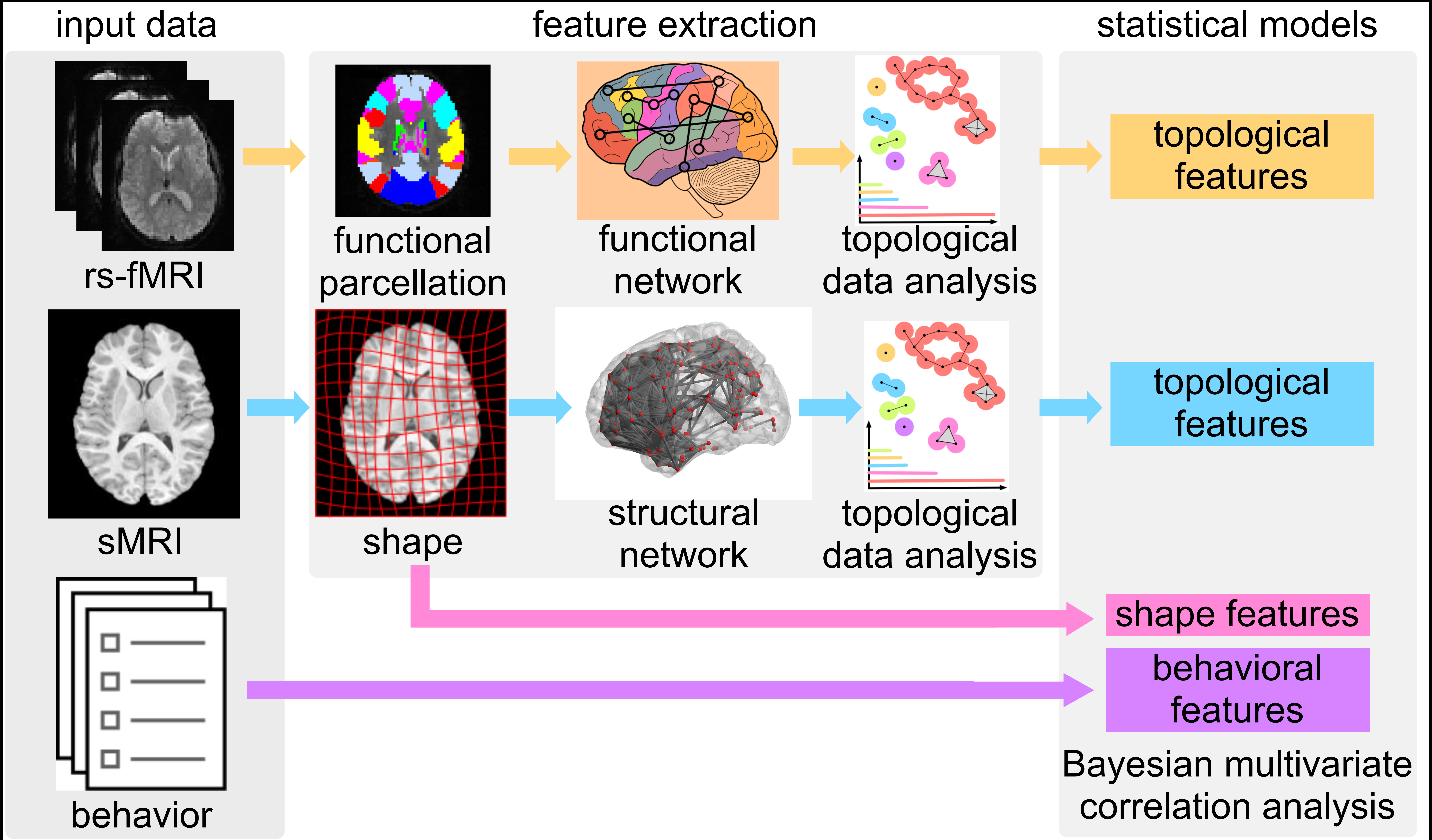
Can we tell autism subject from control?







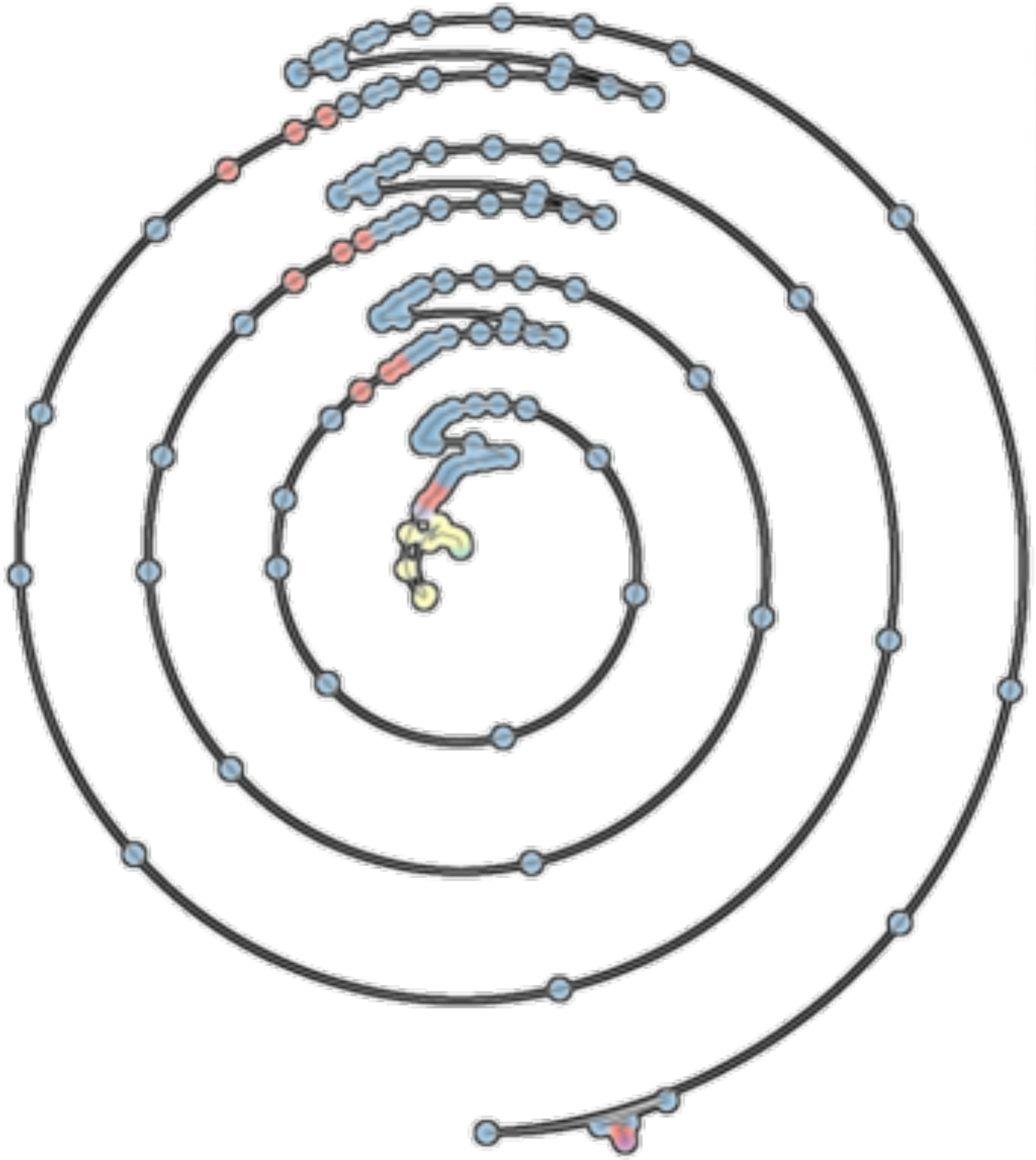
Autism Brain Networks

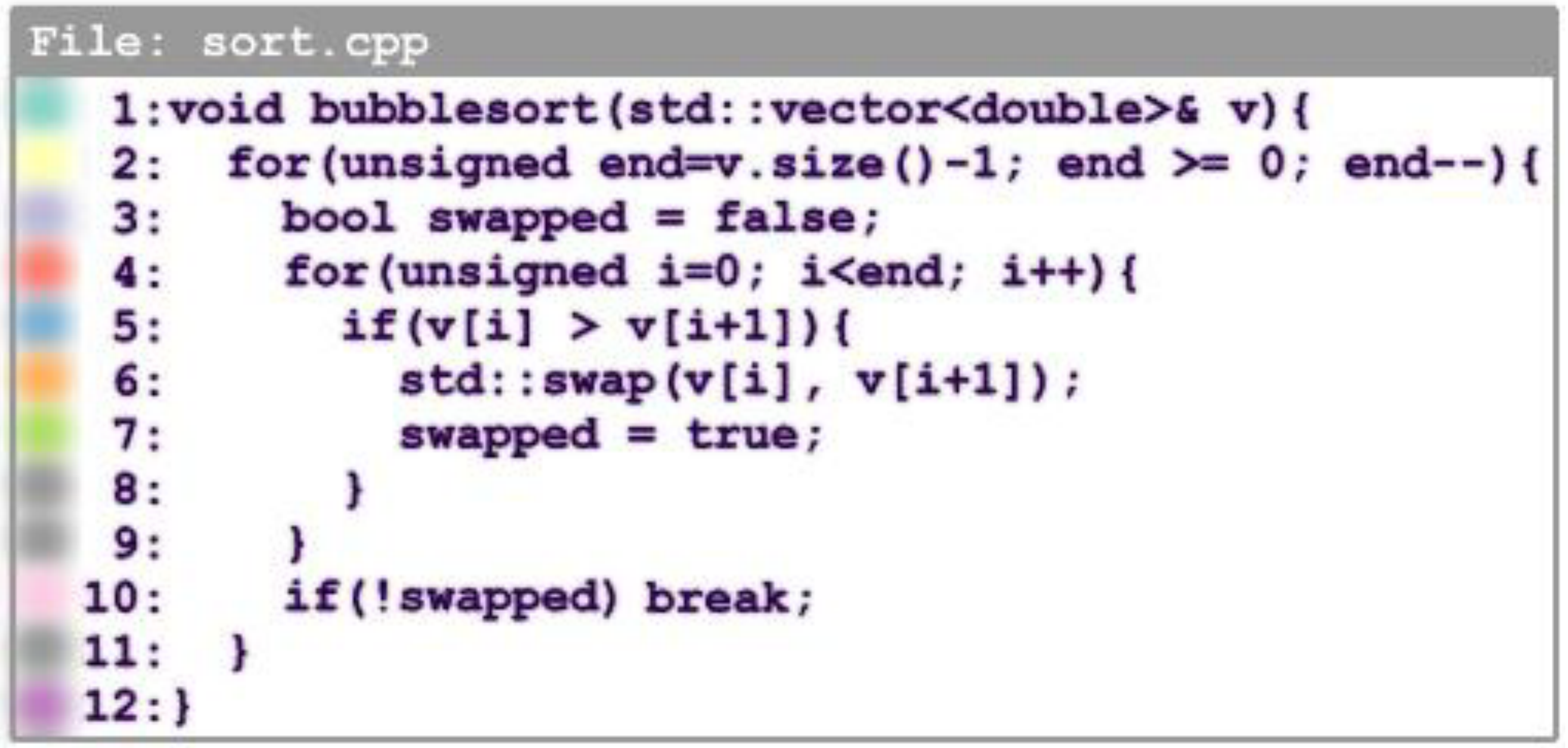


Case study 3: Software Visualization

Circular patterns in a program

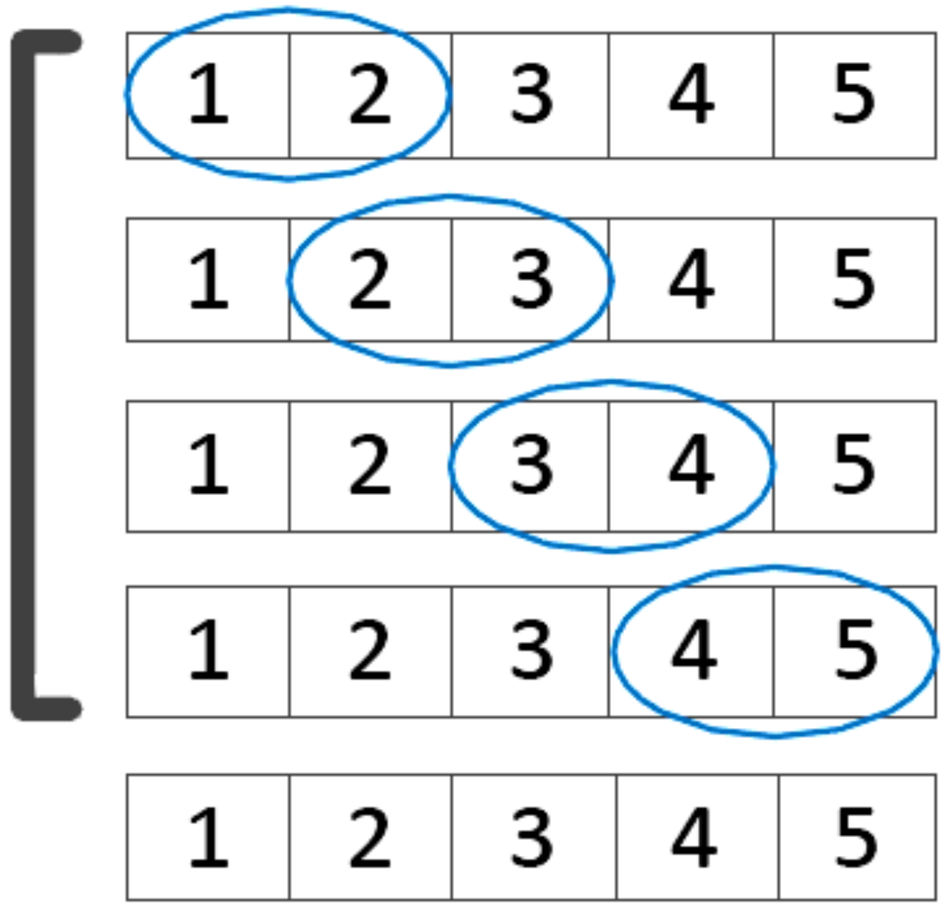
An example





File: sort.cpp

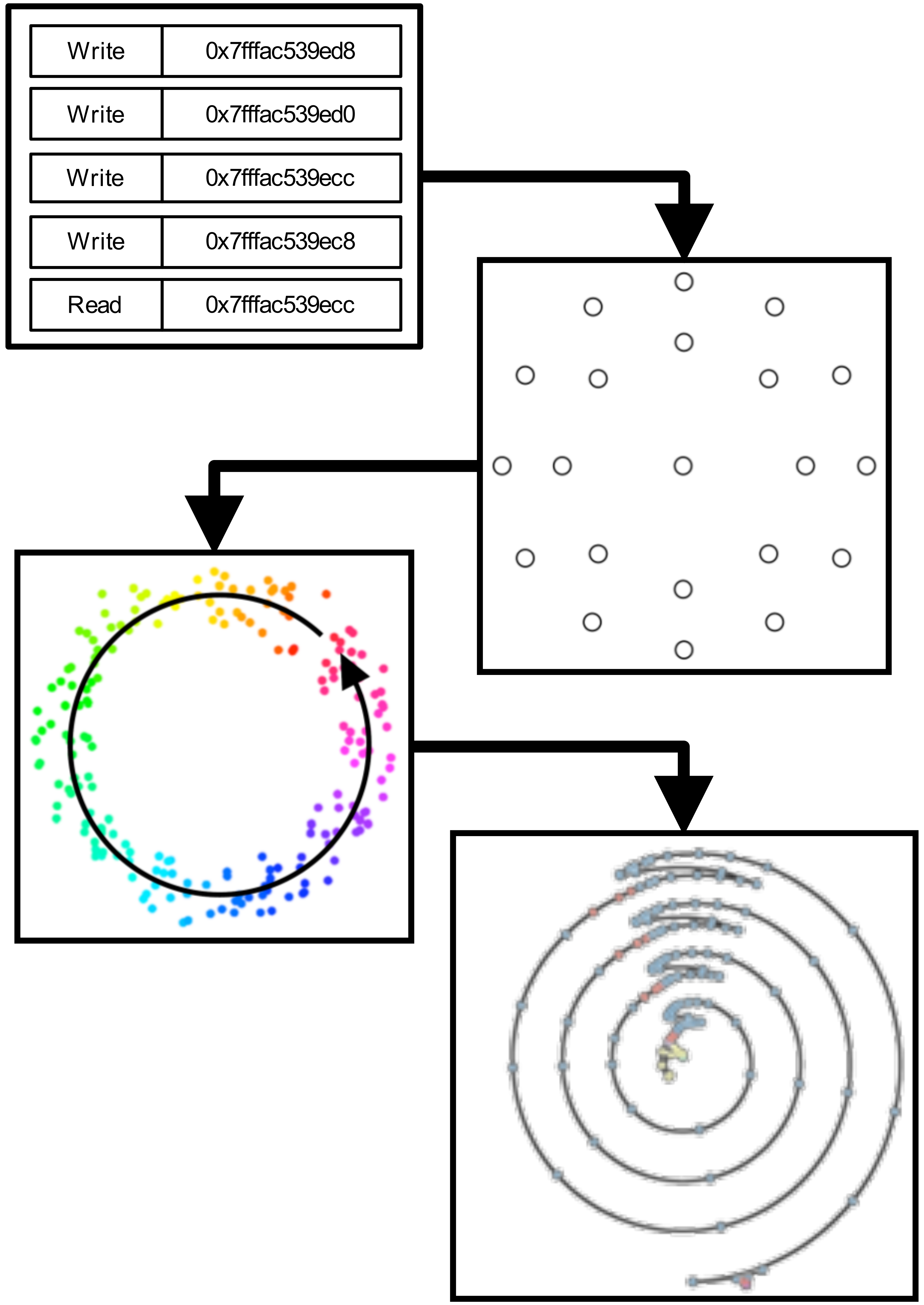
```
void bubblesort(std::vector<double>& v){
  for(unsigned end=v.size()-1; end >= 0; end--){
    bool swapped = false;
    for(unsigned i=0; i<end; i++){
      if(v[i] > v[i+1]){
        std::swap(v[i], v[i+1]);
        swapped = true;
      }
    }
    if(!swapped) break;
  }
}
```



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

A.N.M. Imroz Choudhury, Bei Wang, Paul Rosen and Valerio Pascucci, 2012

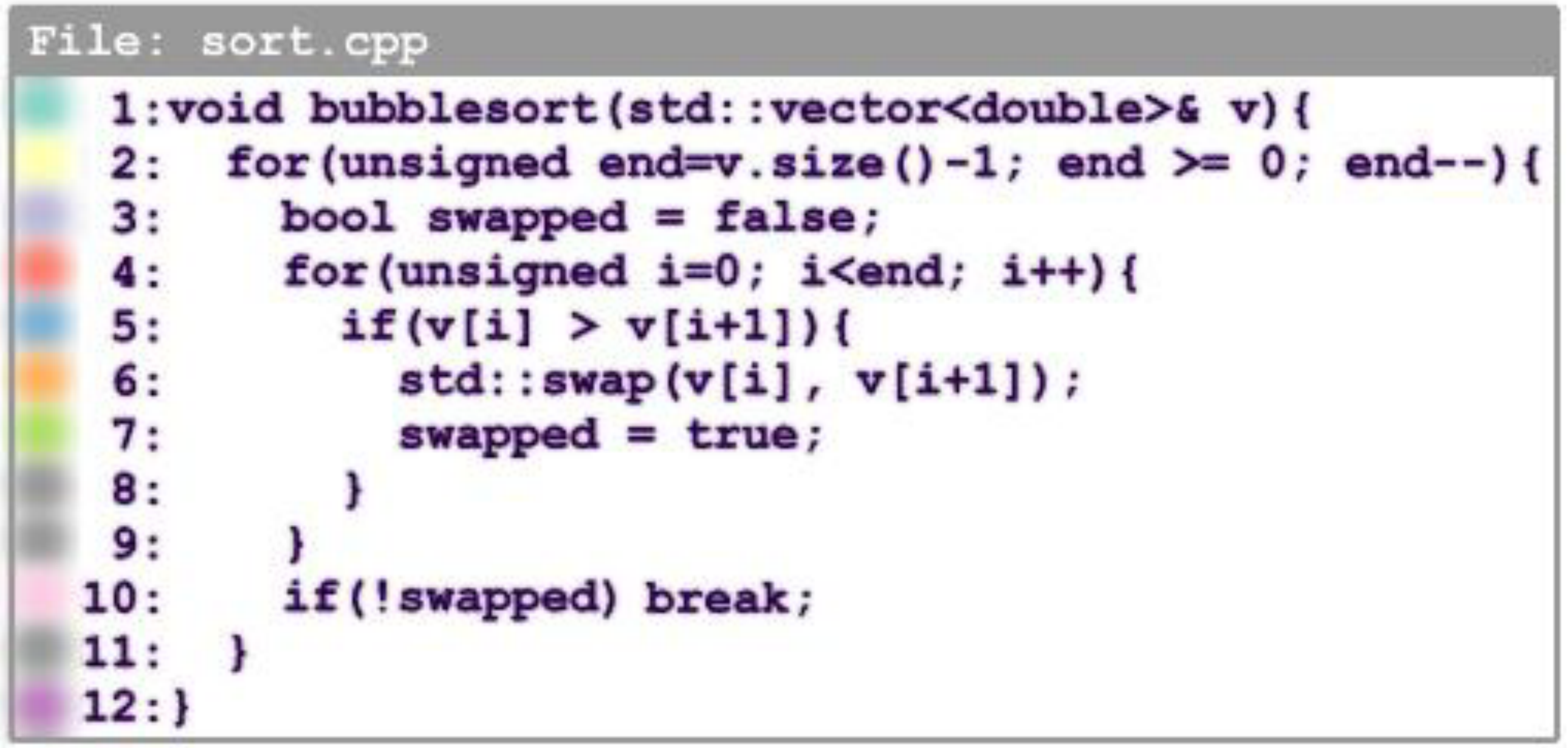
Convert memory reference traces to a point cloud



Write	0x7fffac539ed8
Write	0x7fffac539ed0
Write	0x7fffac539ecc
Write	0x7fffac539ec8
Read	0x7fffac539ecc

- 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



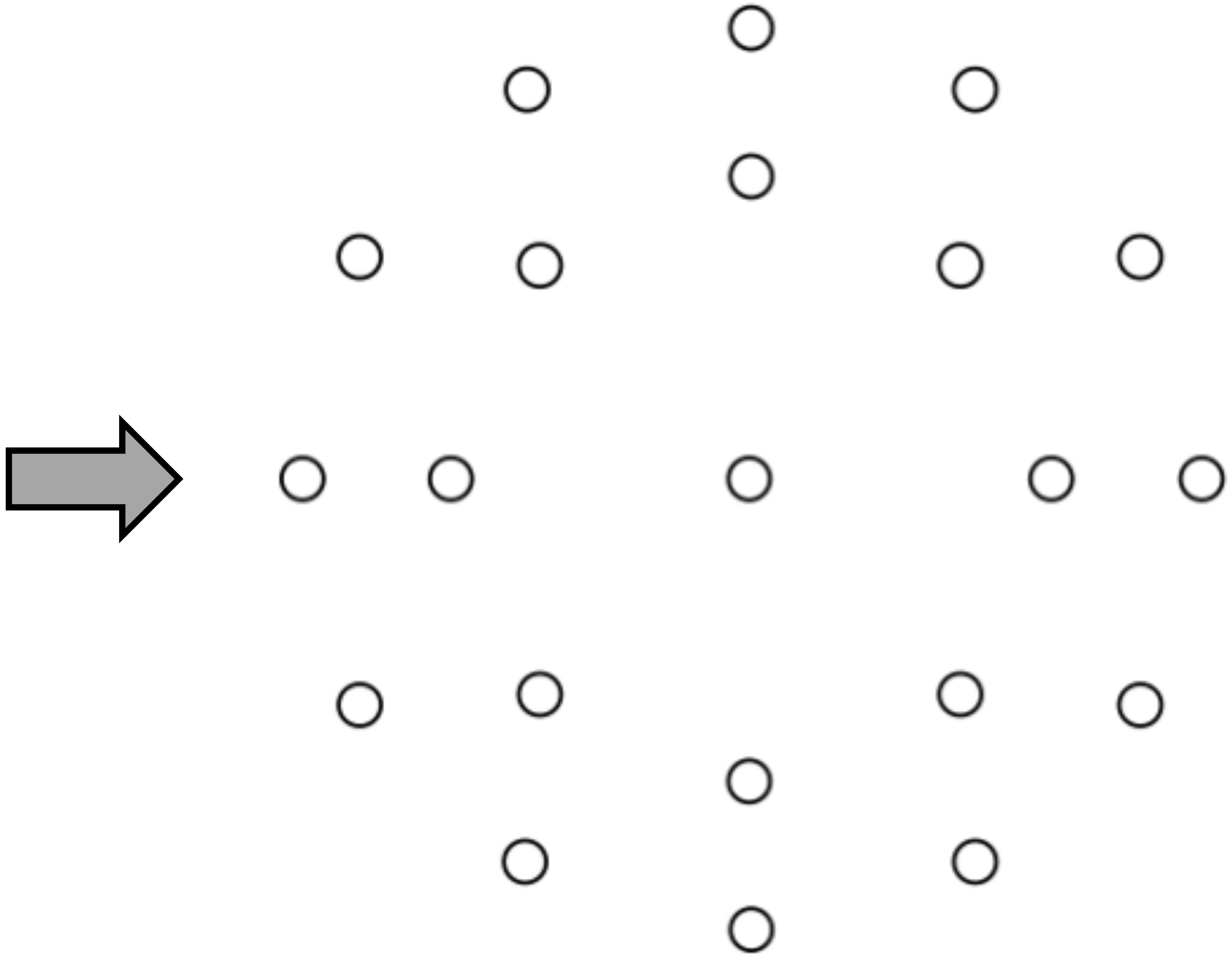
File: sort.cpp

```
void bubblesort(std::vector<double>& v){
  for(unsigned end=v.size()-1; end >= 0; end--){
    bool swapped = false;
    for(unsigned i=0; i<end; i++){
      if(v[i] > v[i+1]){
        std::swap(v[i], v[i+1]);
        swapped = true;
      }
    }
    if(!swapped) break;
  }
}
```

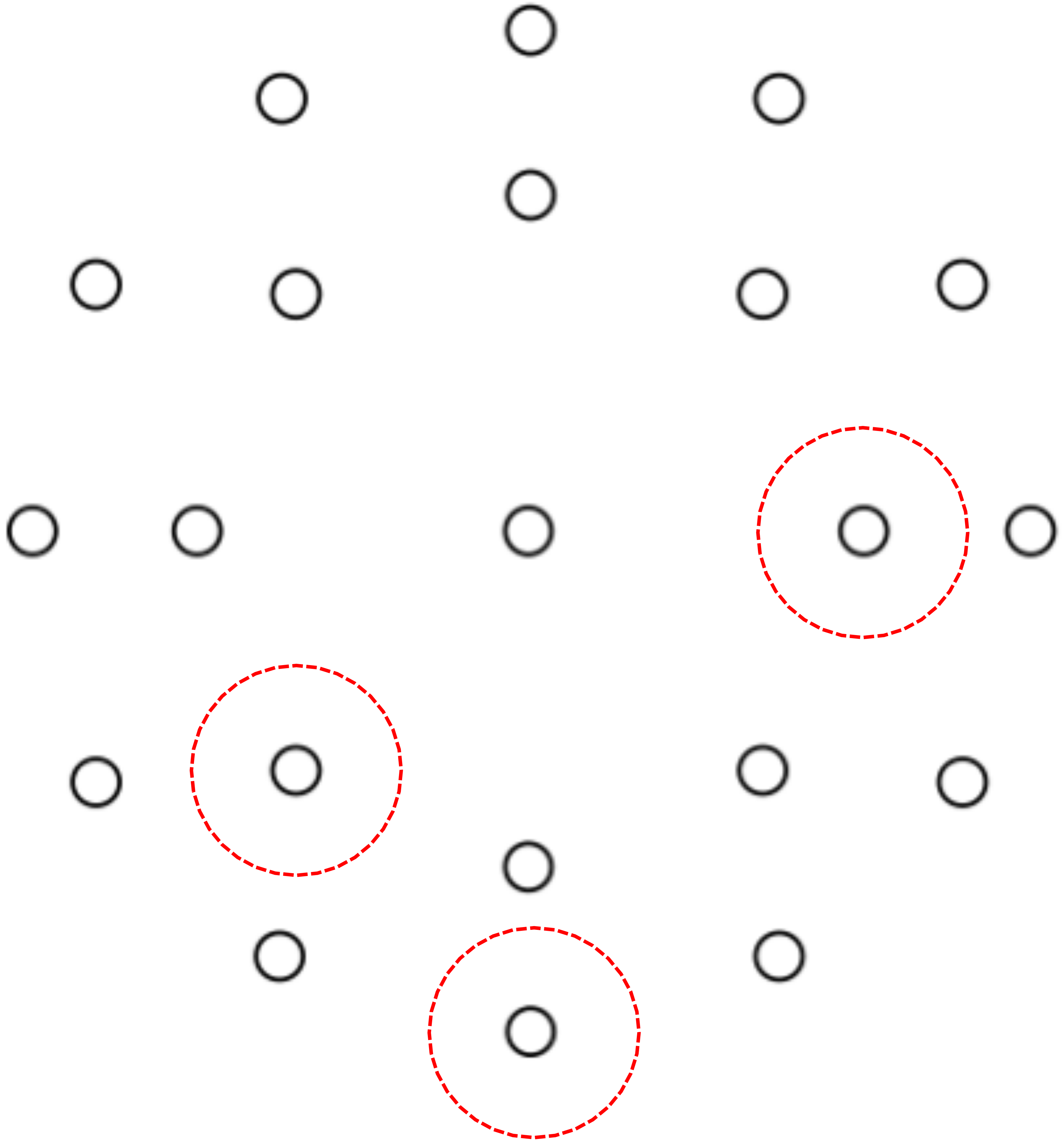
Write	0x7fffac539ed8
Write	0x7fffac539ed0
Write	0x7fffac539ecc
Write	0x7fffac539ec8
Read	0x7fffac539ecc
Read	0x7fffac539ec8
Write	0x7fffac539eb8
Write	0x7fffac539eb0

Memory reference trace to point cloud

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

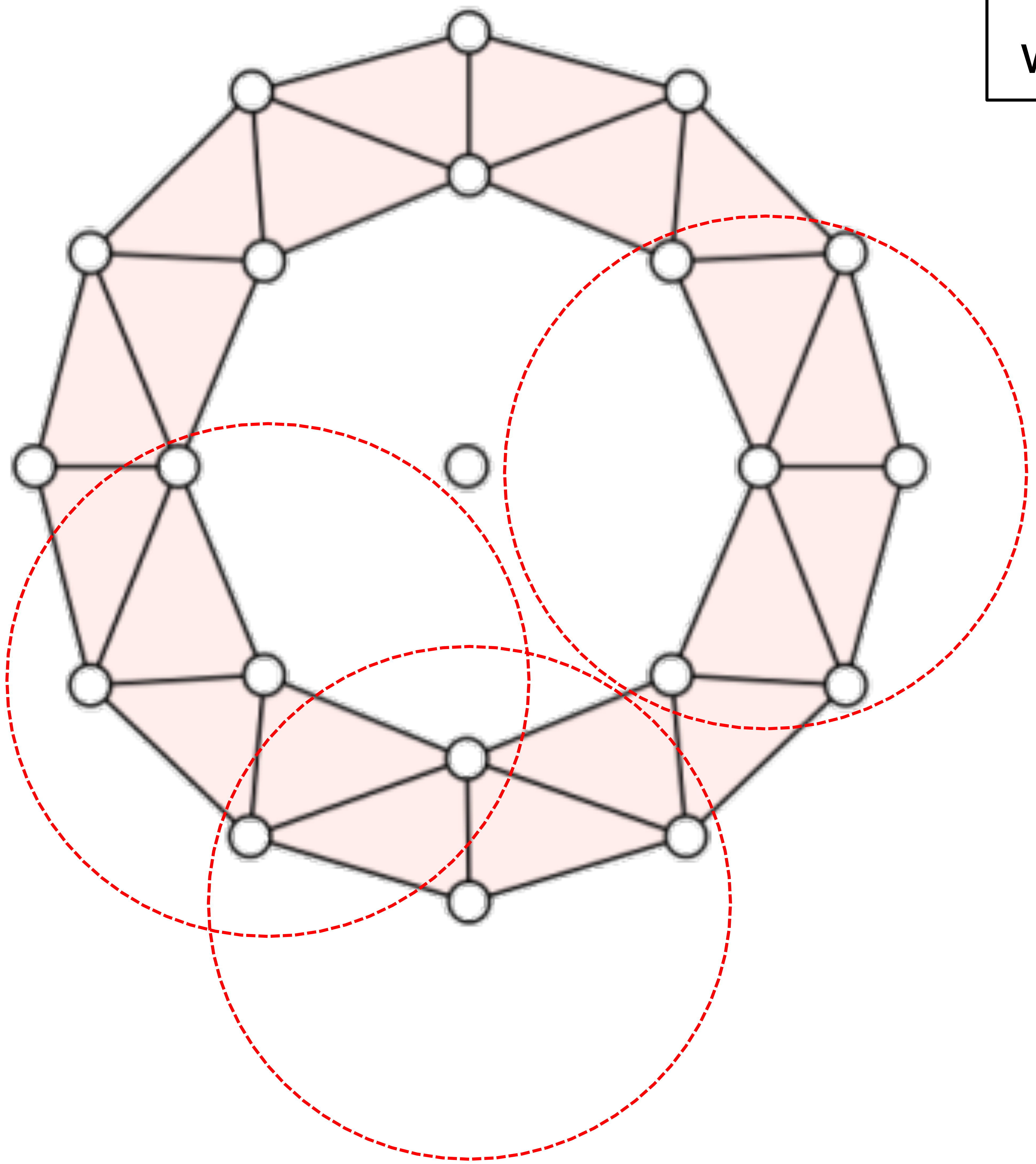


Topological data analysis to identify cycles



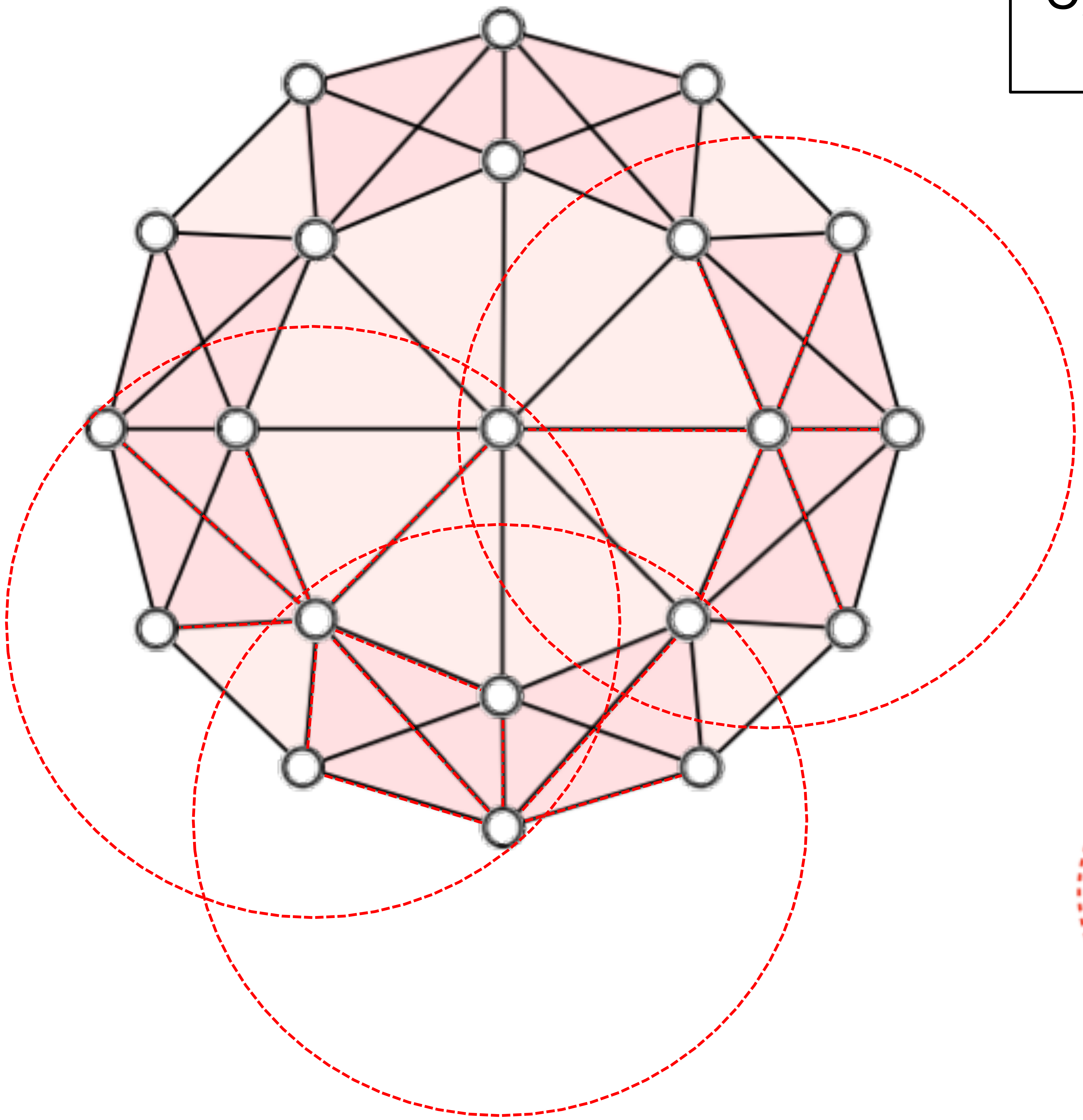
No connectivity
No cycles

Topological data analysis to identify cycles



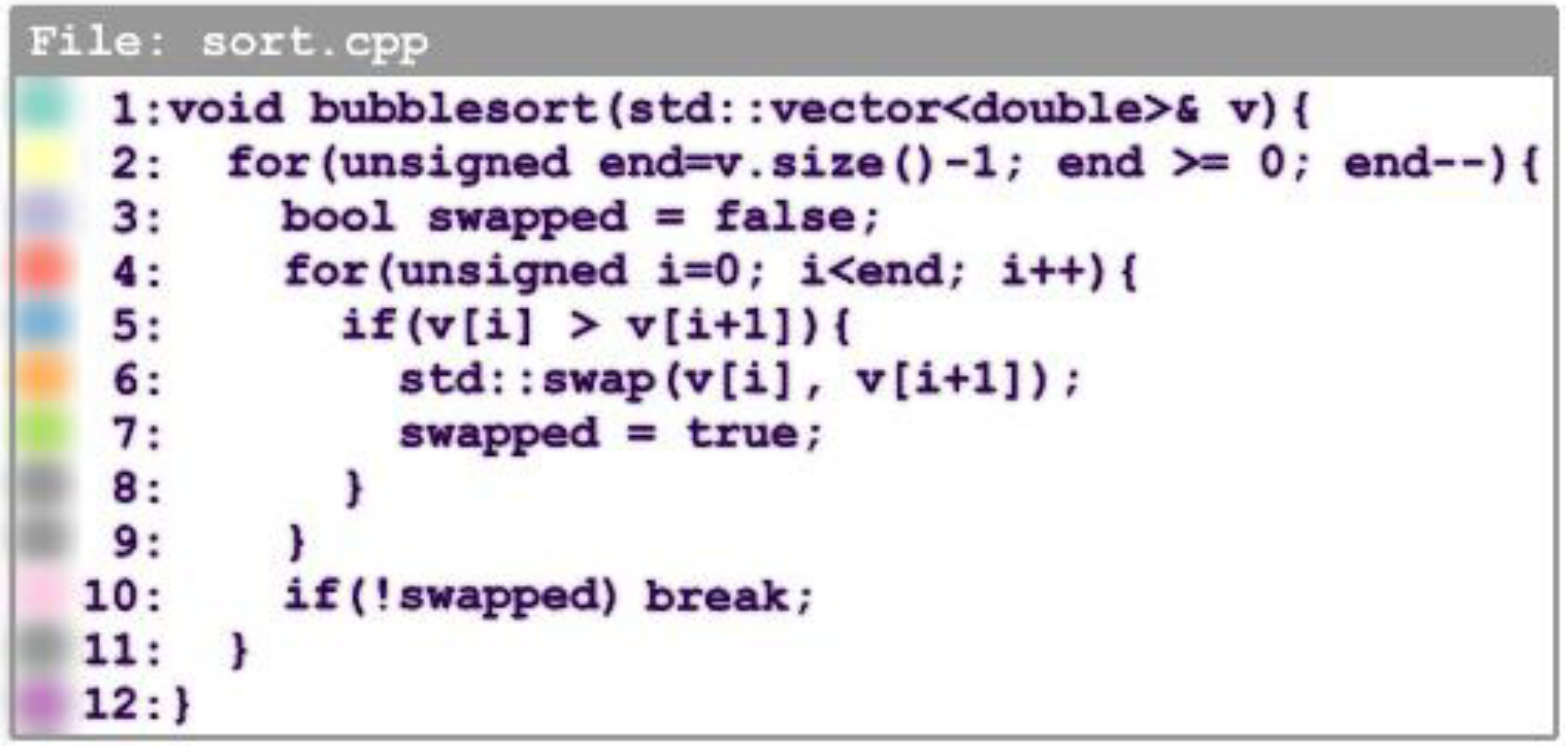
Cycle is found with birth time $\mathcal{E}_1$

Topological data analysis to identify cycles



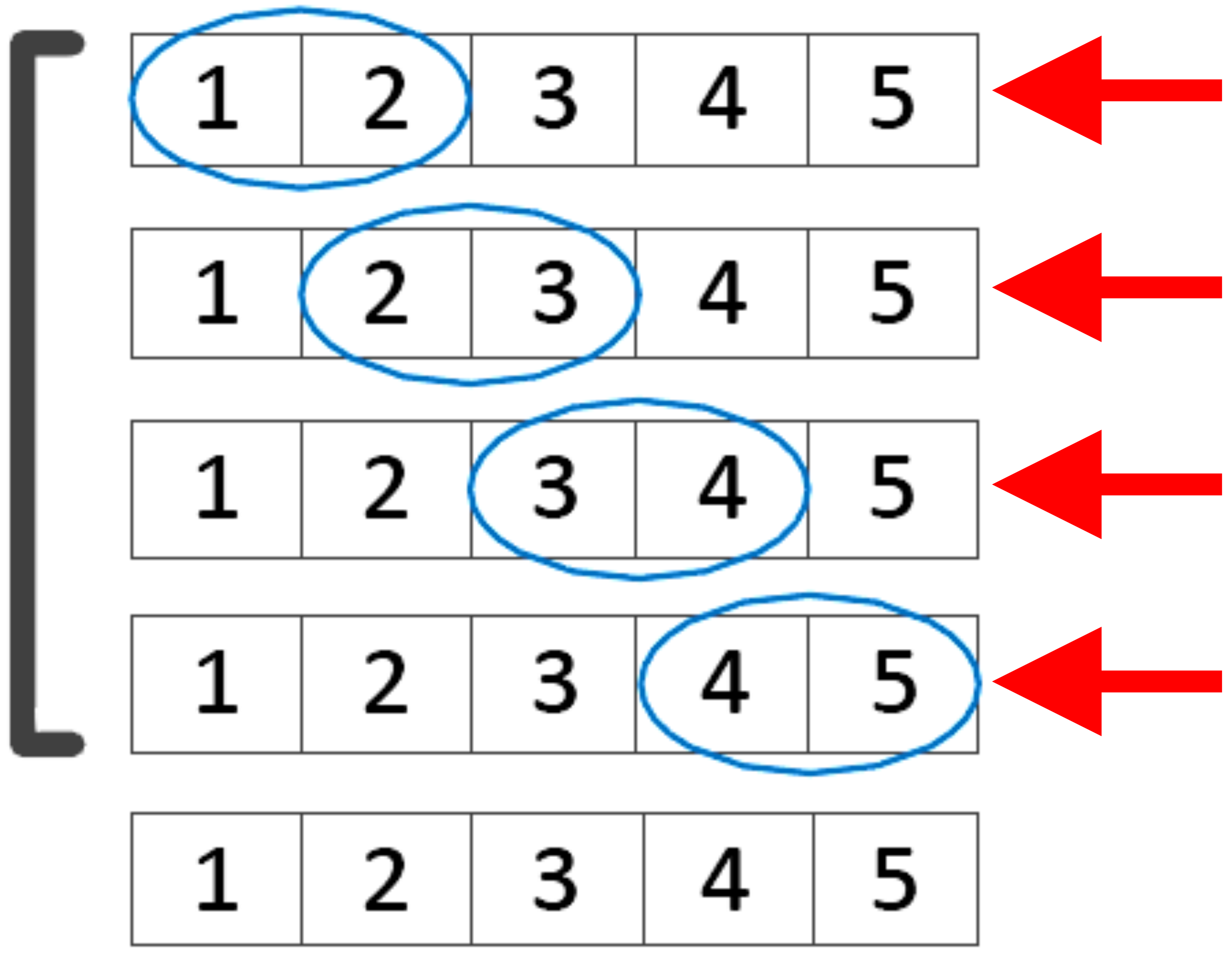
Cycle has died at time $\mathcal{E}_2$

Data dependent structures

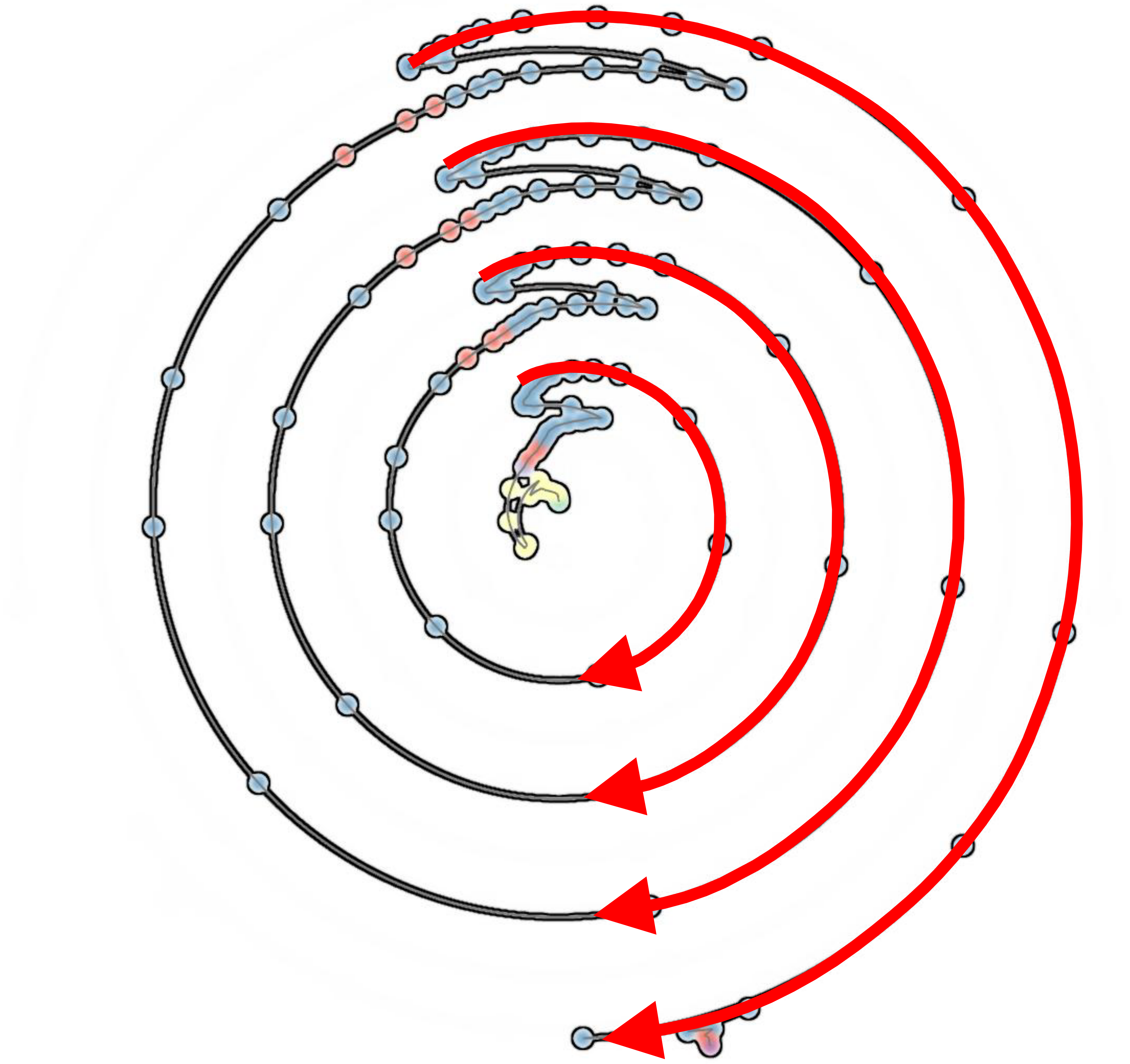


File: sort.cpp

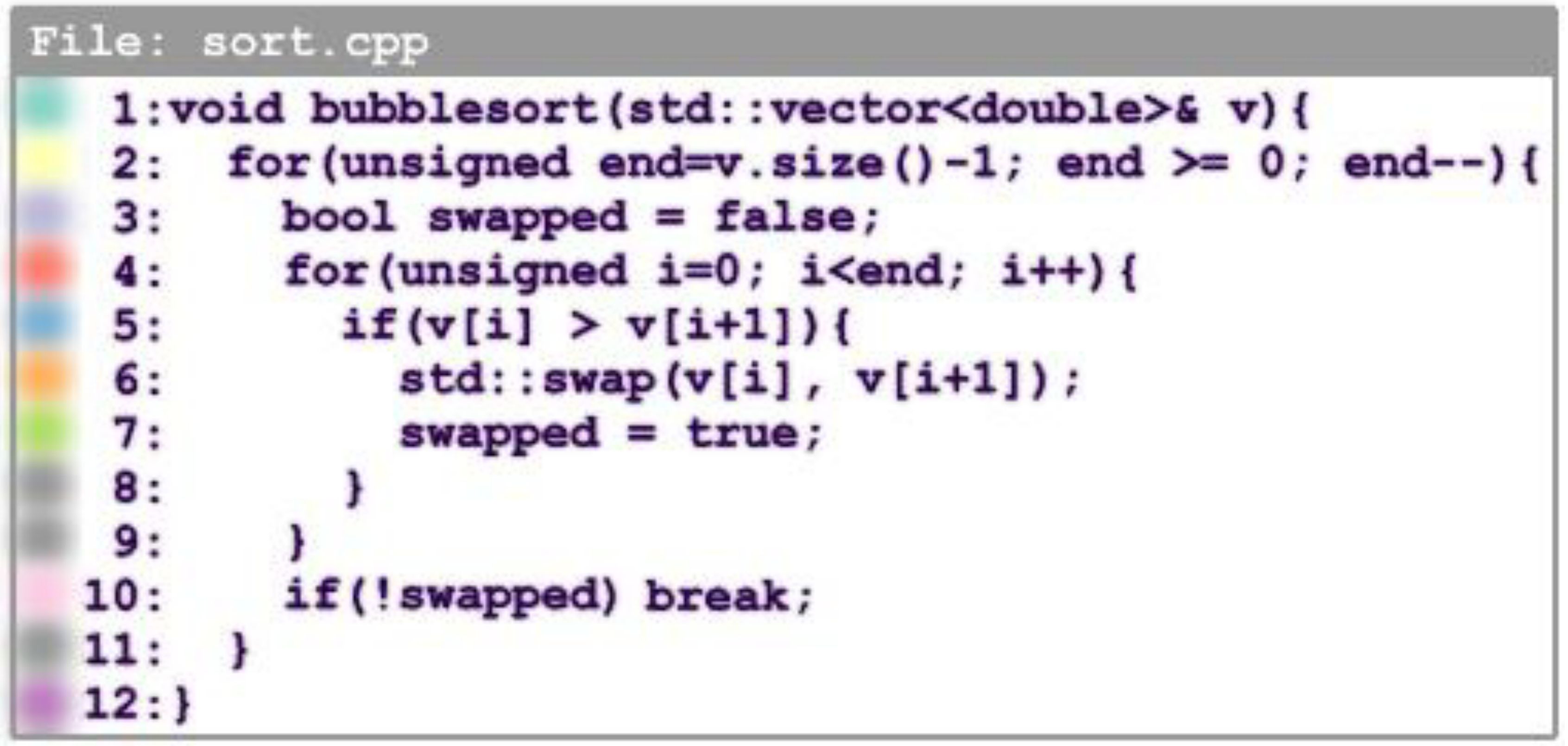
```
void bubblesort(std::vector<double>& v){
  for(unsigned end=v.size()-1; end >= 0; end--){
    bool swapped = false;
    for(unsigned i=0; i<end; i++){
      if(v[i] > v[i+1]){
        std::swap(v[i], v[i+1]);
        swapped = true;
      }
    }
    if(!swapped) break;
  }
}
```



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

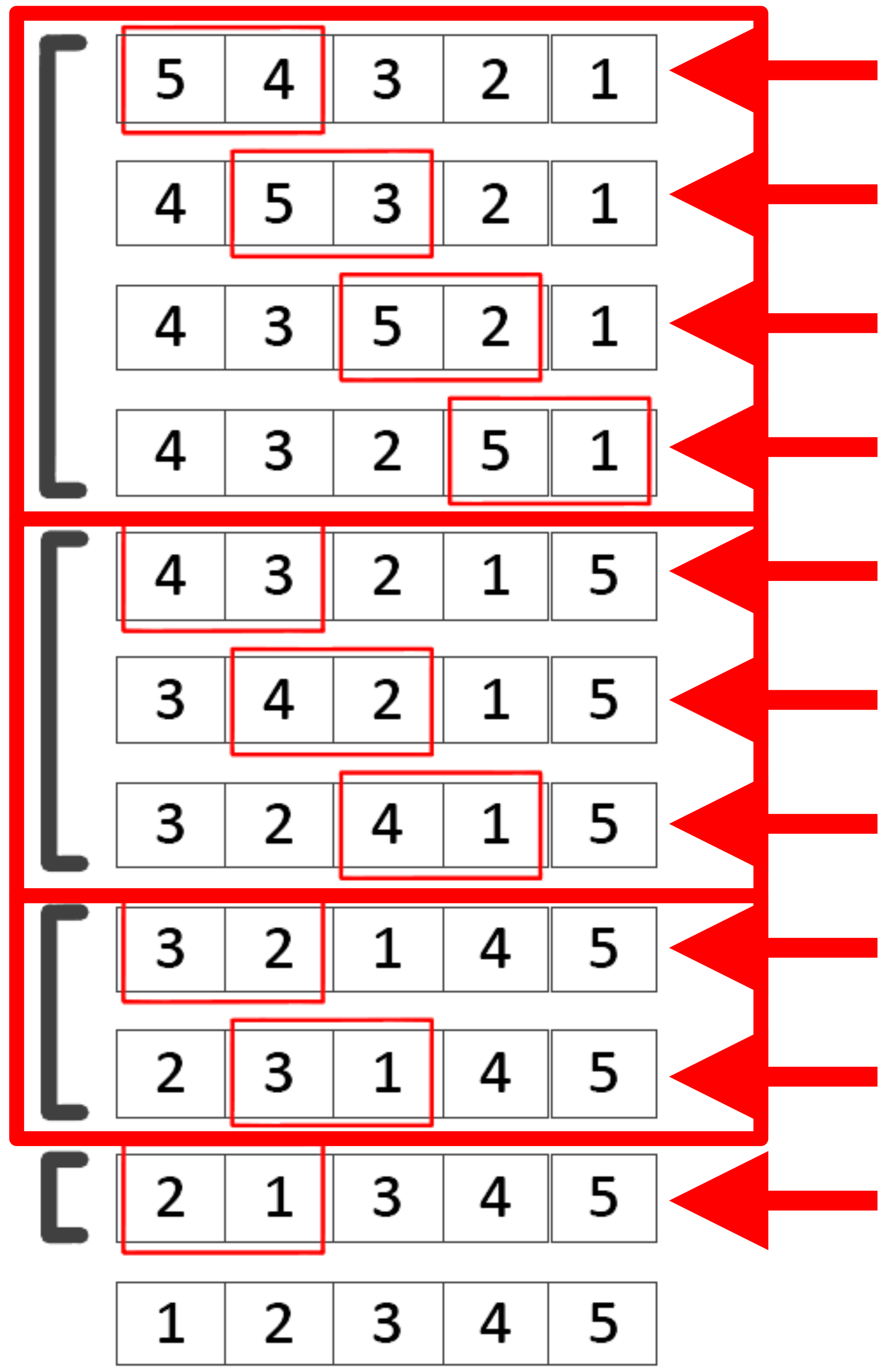


Data dependent structures

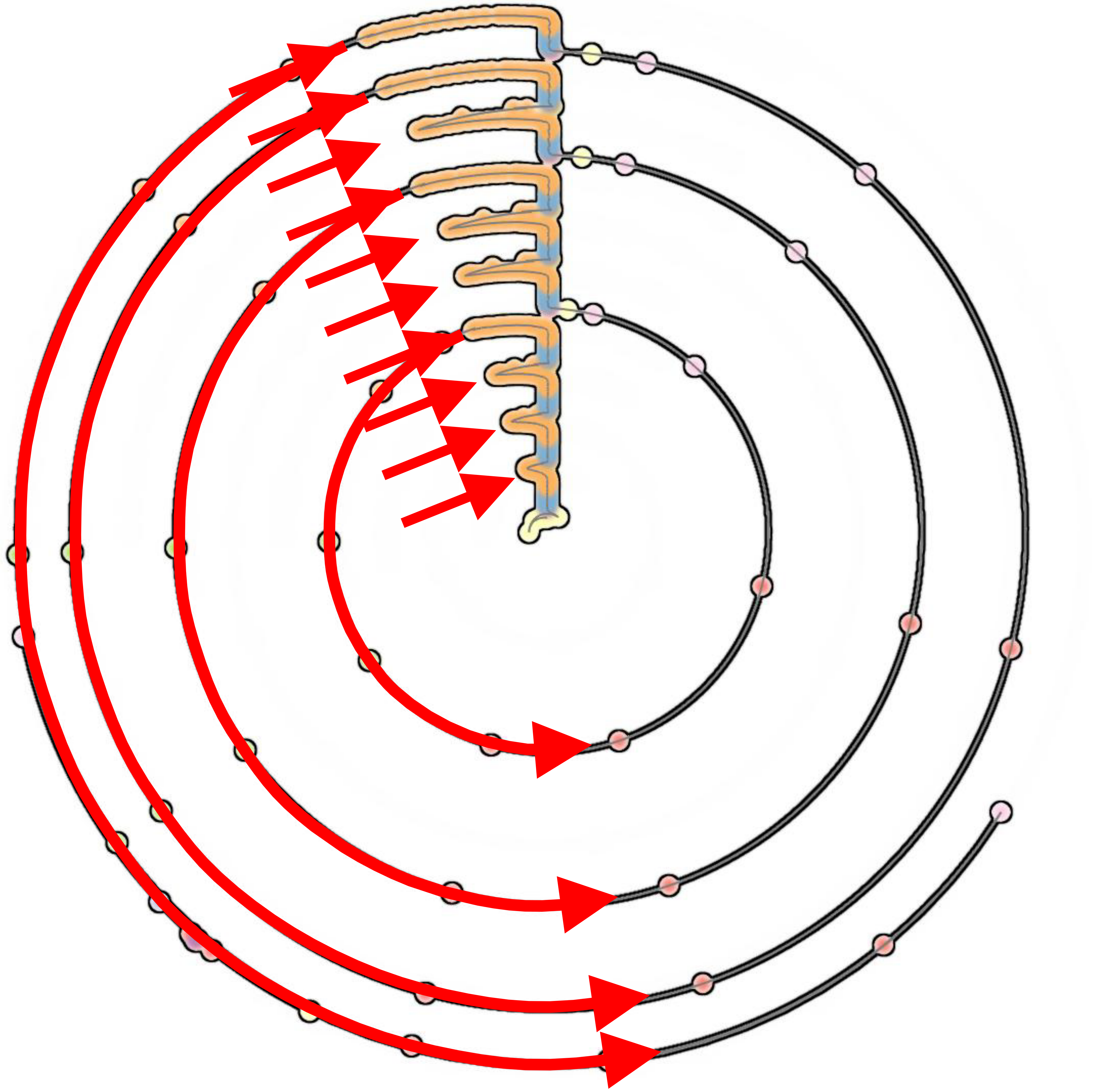


File: sort.cpp

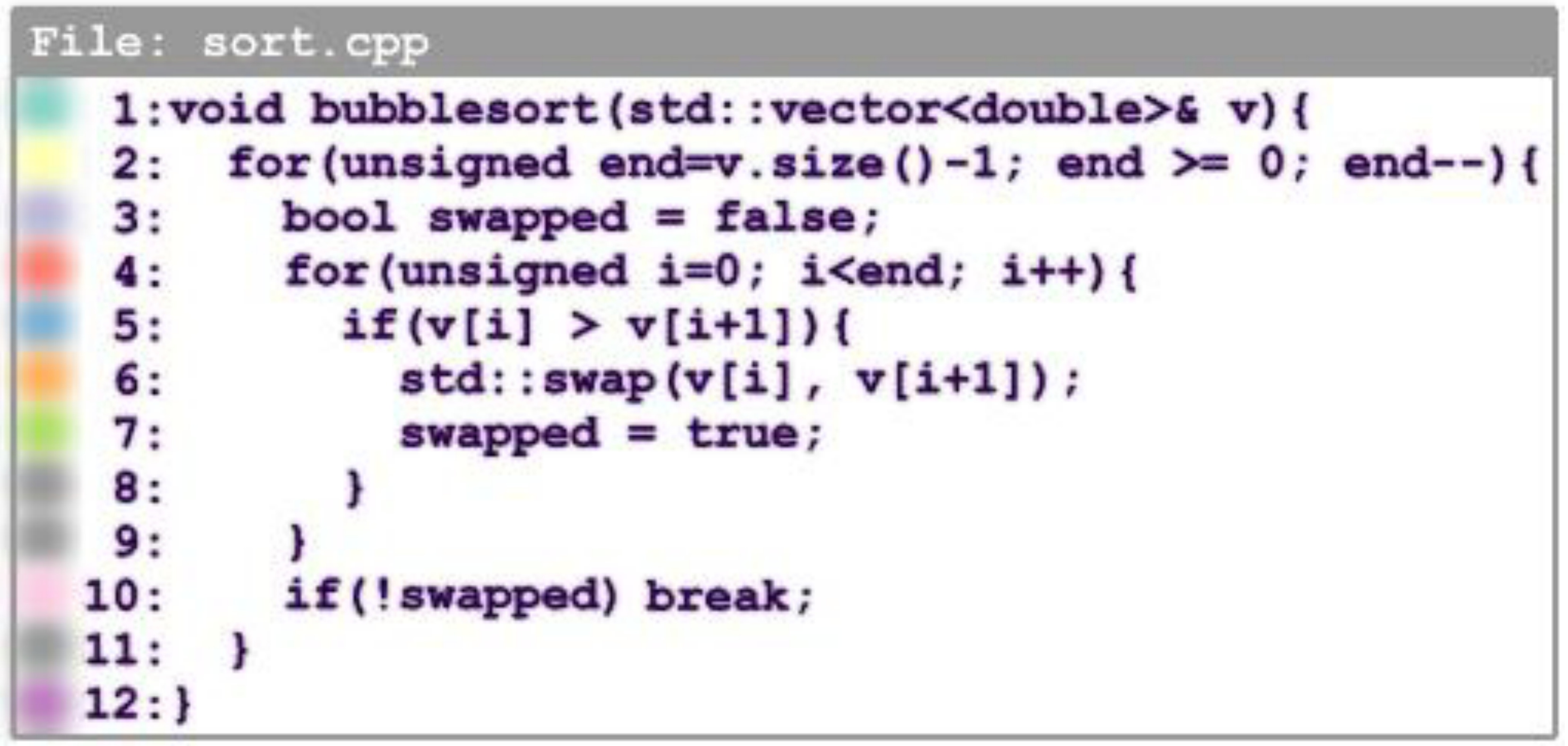
```
void bubblesort(std::vector<double>& v){
  for(unsigned end=v.size()-1; end >= 0; end--){
    bool swapped = false;
    for(unsigned i=0; i<end; i++){
      if(v[i] > v[i+1]){
        std::swap(v[i], v[i+1]);
        swapped = true;
      }
    }
    if(!swapped) break;
  }
}
```



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

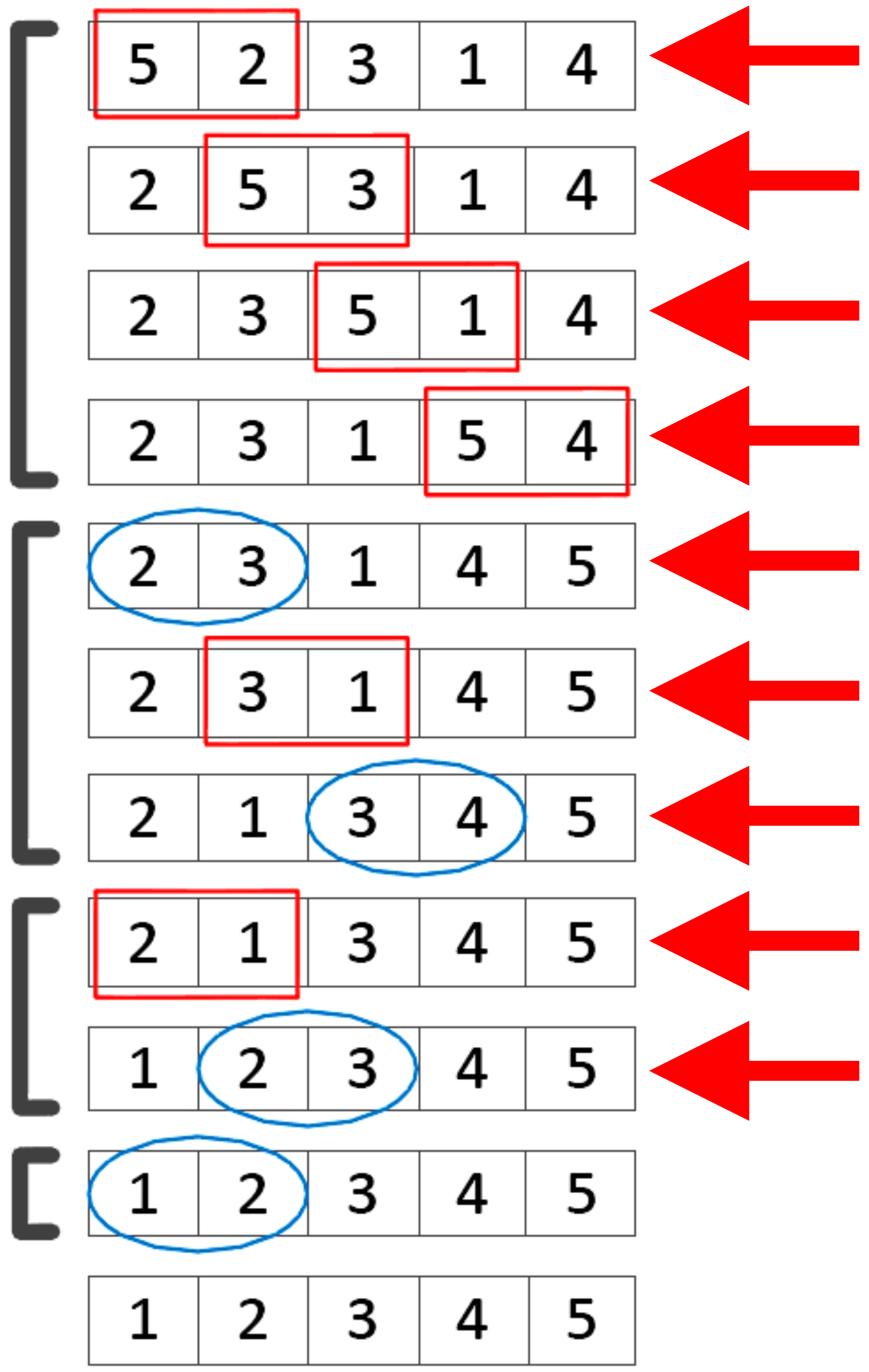


Data dependent structures

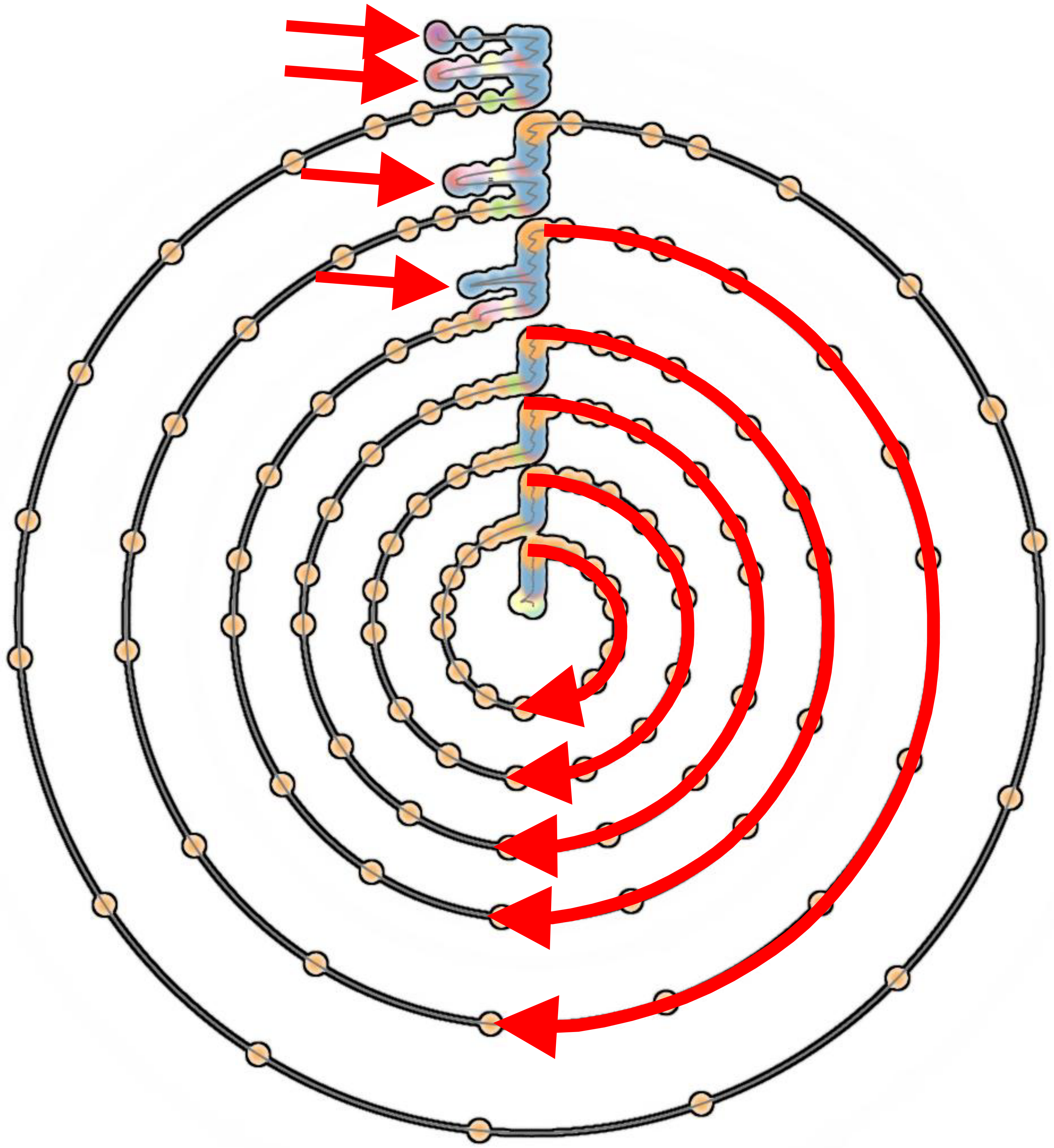


File: sort.cpp

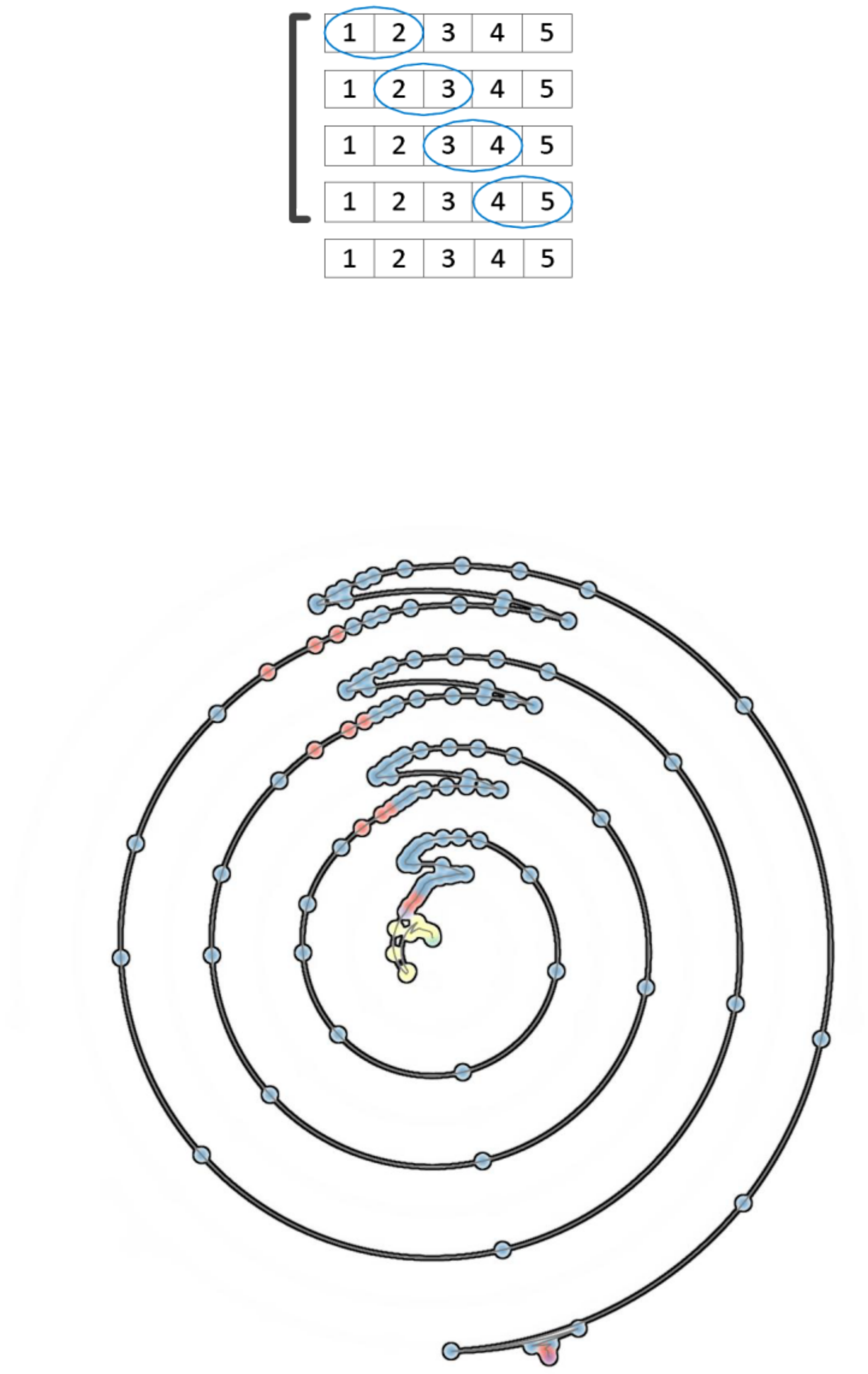
```
void bubblesort(std::vector<double>& v){
  for(unsigned end=v.size()-1; end >= 0; end--){
    bool swapped = false;
    for(unsigned i=0; i<end; i++){
      if(v[i] > v[i+1]){
        std::swap(v[i], v[i+1]);
        swapped = true;
      }
    }
    if(!swapped) break;
  }
}
```



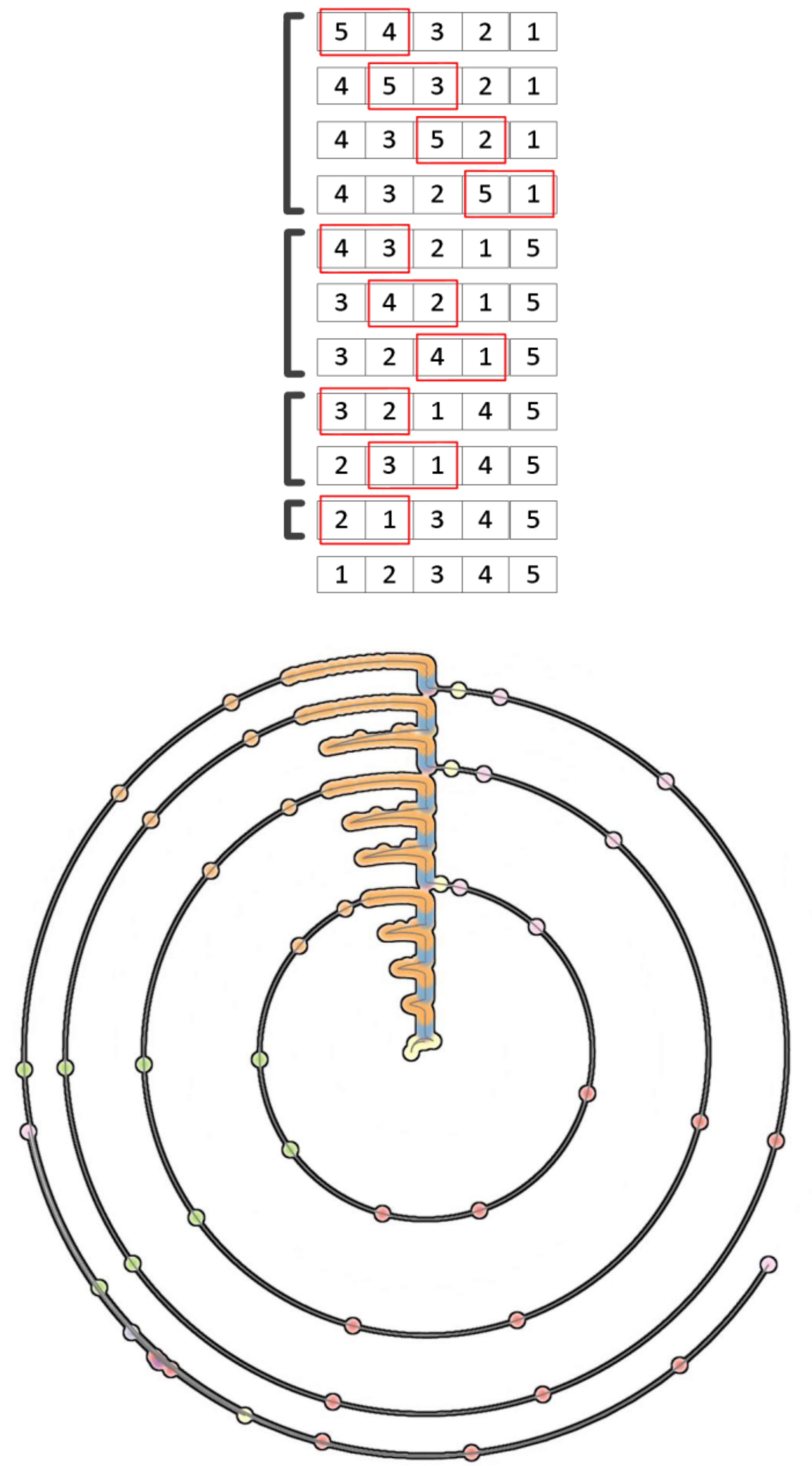
5	2	3	1	4
2	5	3	1	4
2	3	5	1	4
2	3	1	5	4
2	3	1	4	5
2	3	1	4	5
2	1	3	4	5
2	1	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5



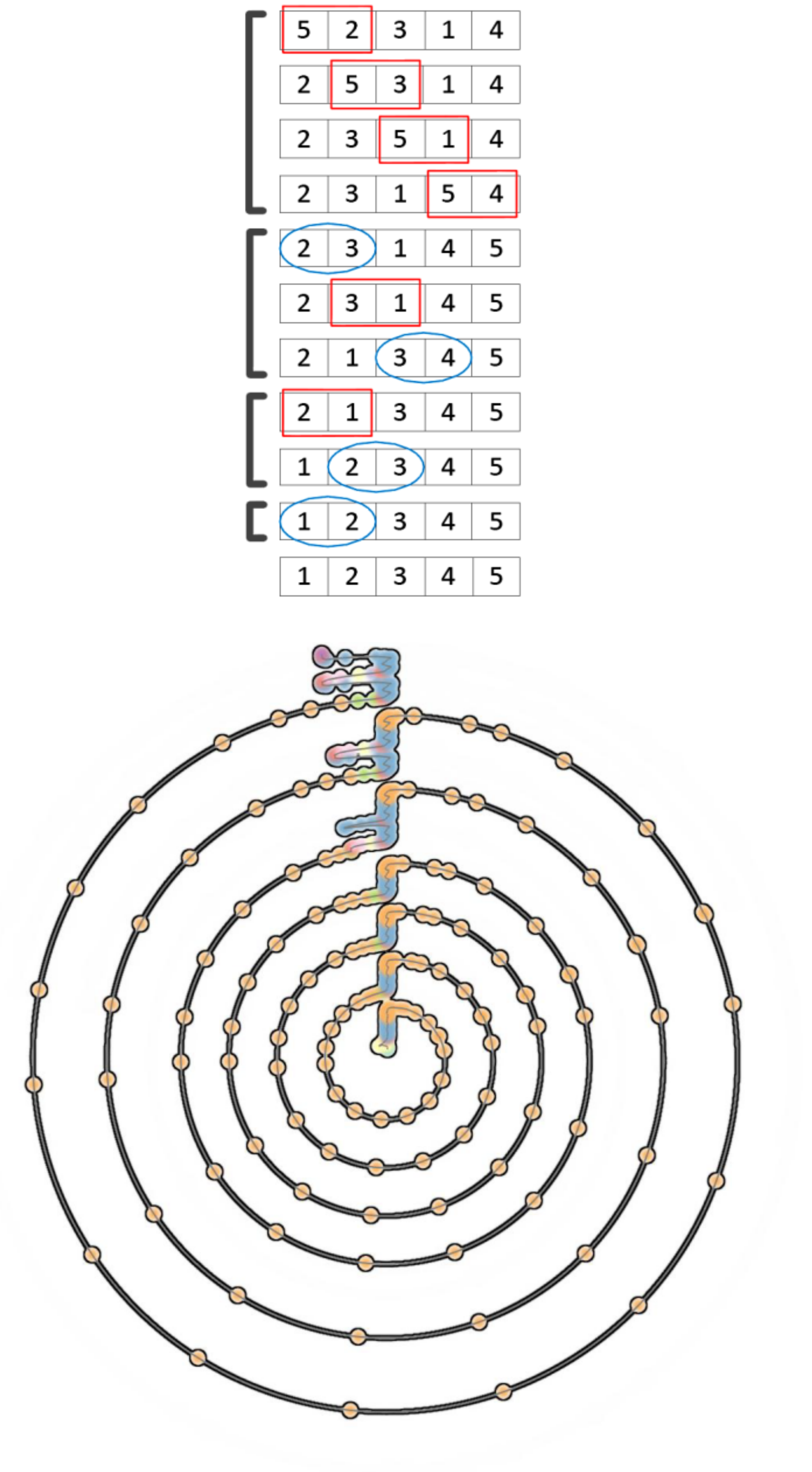
Data dependent structures



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

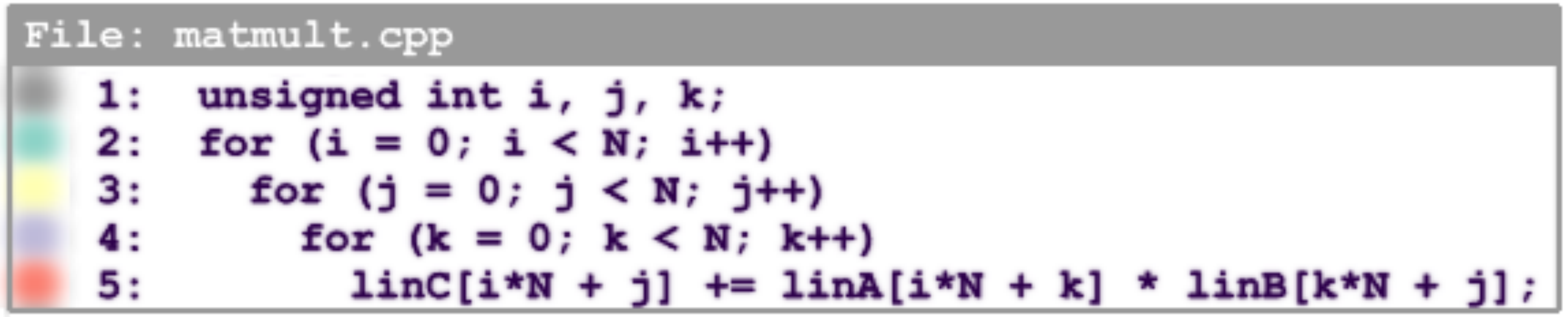


5	4	3	2	1
4	5	3	2	1
4	3	5	2	1
4	3	2	5	1
4	3	2	1	5
3	4	2	1	5
3	2	4	1	5
3	2	1	4	5
2	3	1	4	5
2	1	3	4	5
1	2	3	4	5



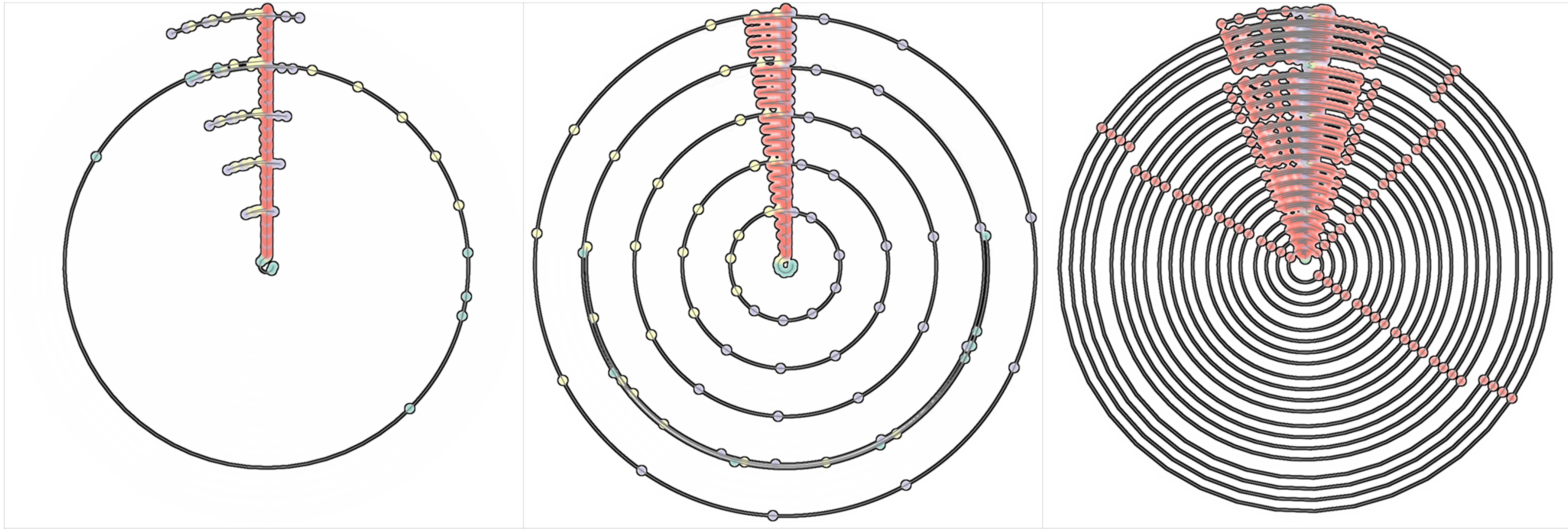
5	2	3	1	4
2	5	3	1	4
2	3	5	1	4
2	3	1	5	4
2	3	1	4	5
2	3	1	4	5
2	1	3	4	5
2	1	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

Algorithm dependent structures

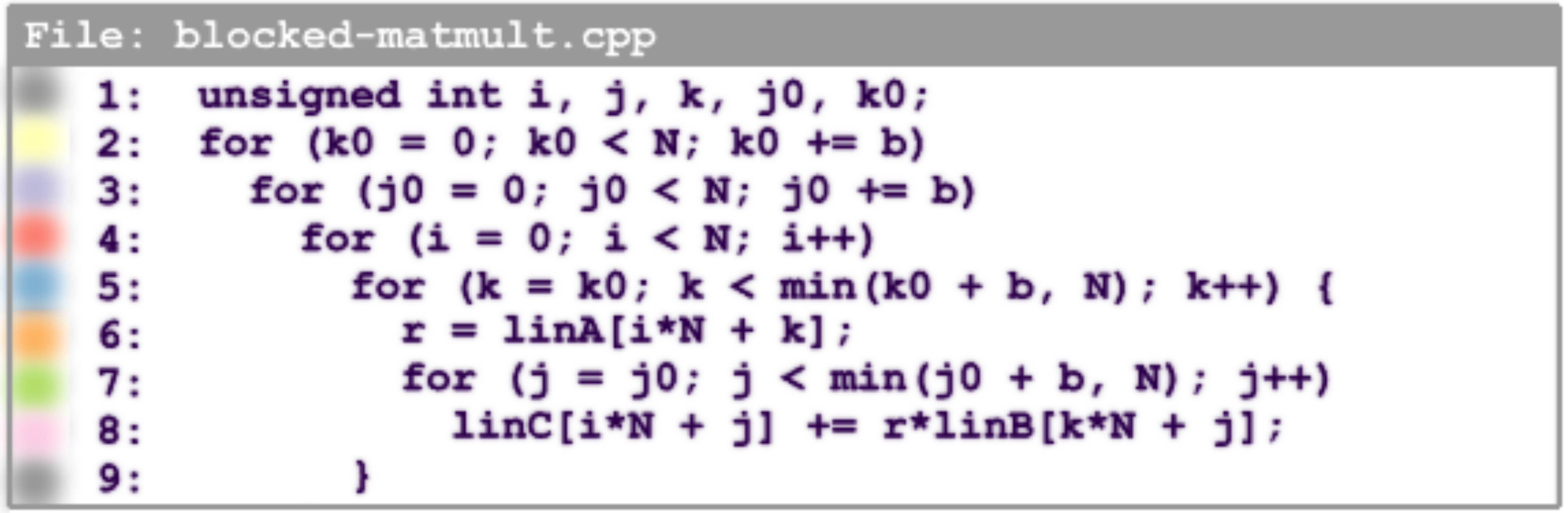


File: matmult.cpp

```
unsigned int i, j, k;
for (i = 0; i < N; i++)
  for (j = 0; j < N; j++)
    for (k = 0; k < N; k++)
      linC[i*N + j] += linA[i*N + k] * linB[k*N + j];
```

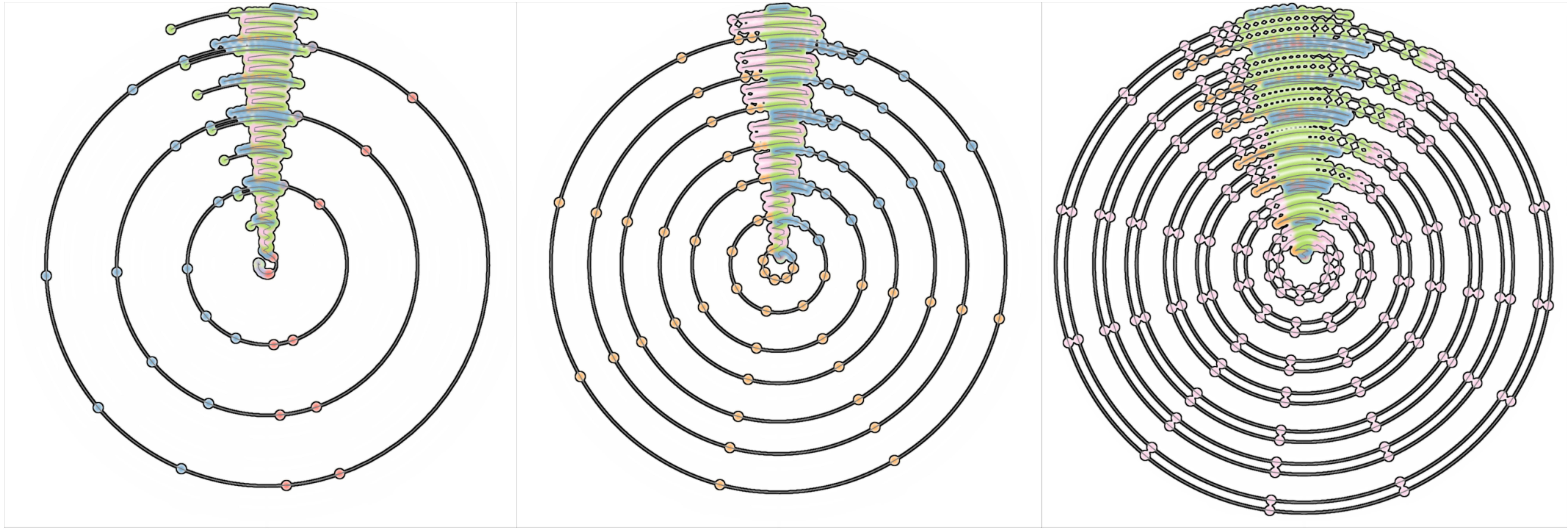


Algorithm dependent structures

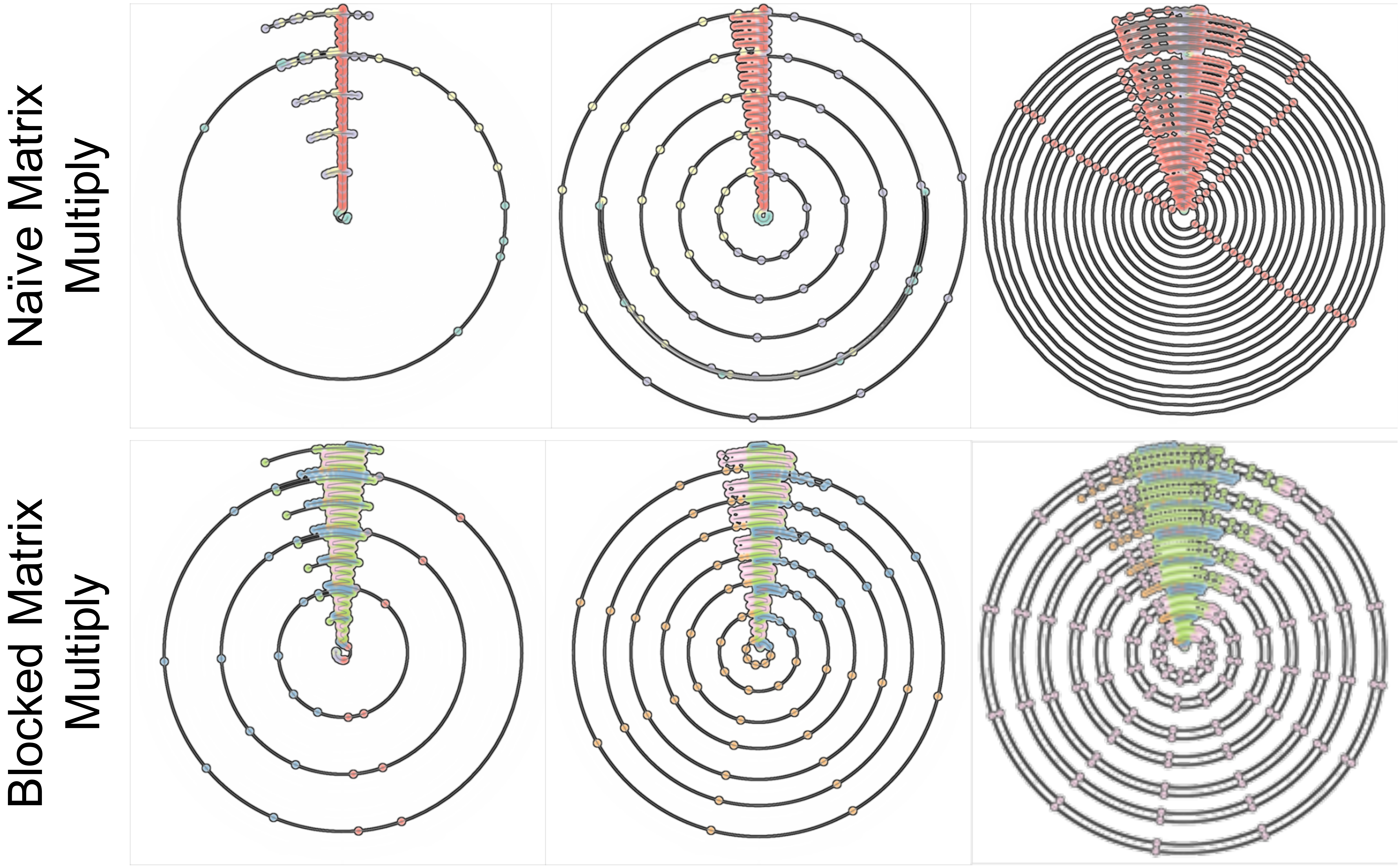


File: blocked-matmult.cpp

```
unsigned int i, j, k, j0, k0;
for (k0 = 0; k0 < N; k0 += b)
  for (j0 = 0; j0 < N; j0 += b)
    for (i = 0; i < N; i++)
      for (k = k0; k < min(k0 + b, N); k++) {
        r = linA[i*N + k];
        for (j = j0; j < min(j0 + b, N); j++)
          linC[i*N + j] += r*linB[k*N + j];
      }
```



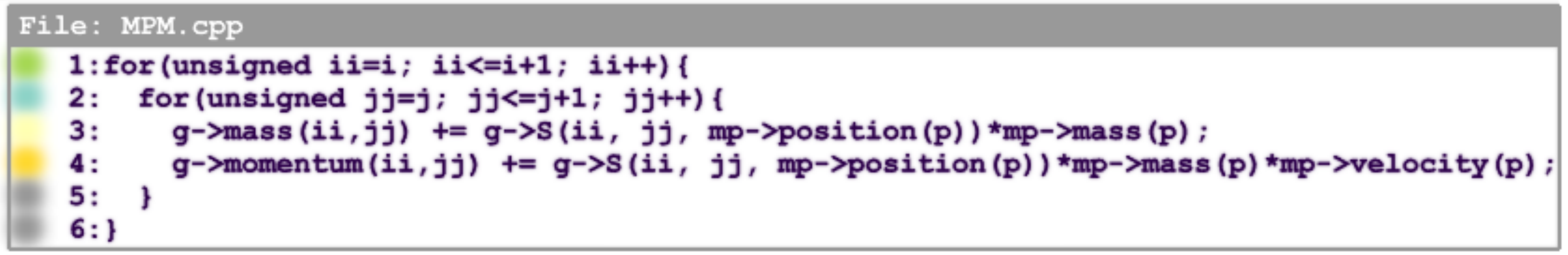
Algorithm dependent structures



Naïve Matrix Multiply

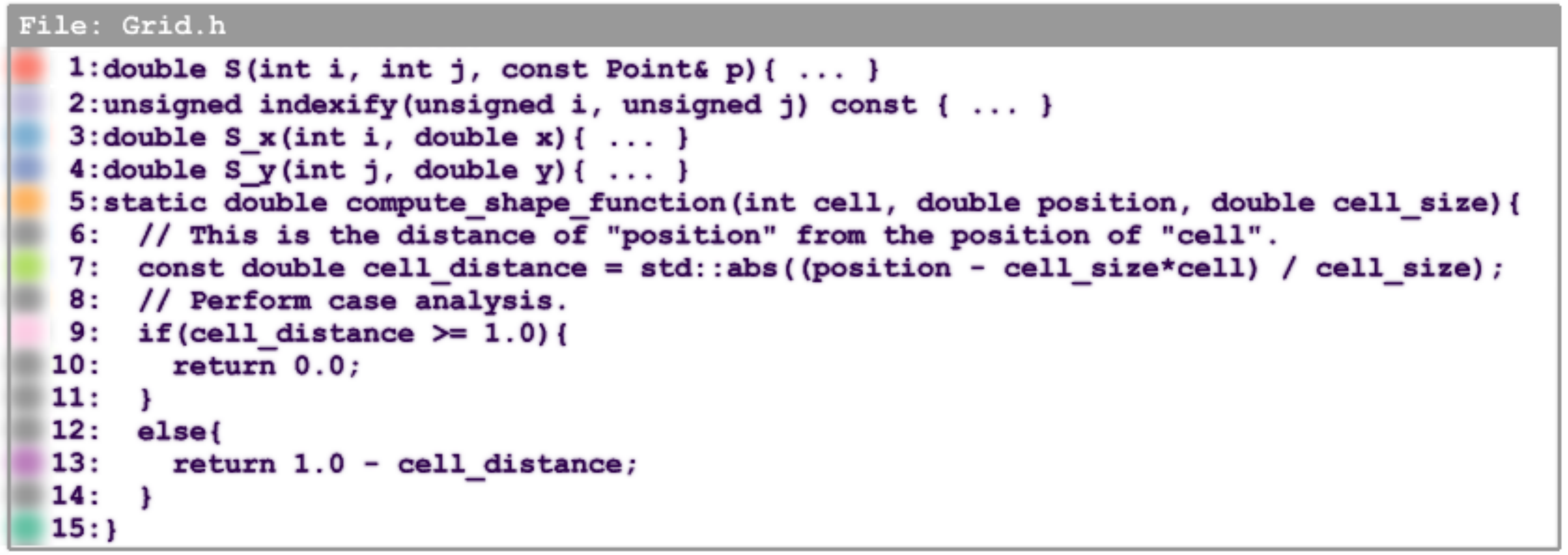
Blocked Matrix Multiply

Non-loop based structures



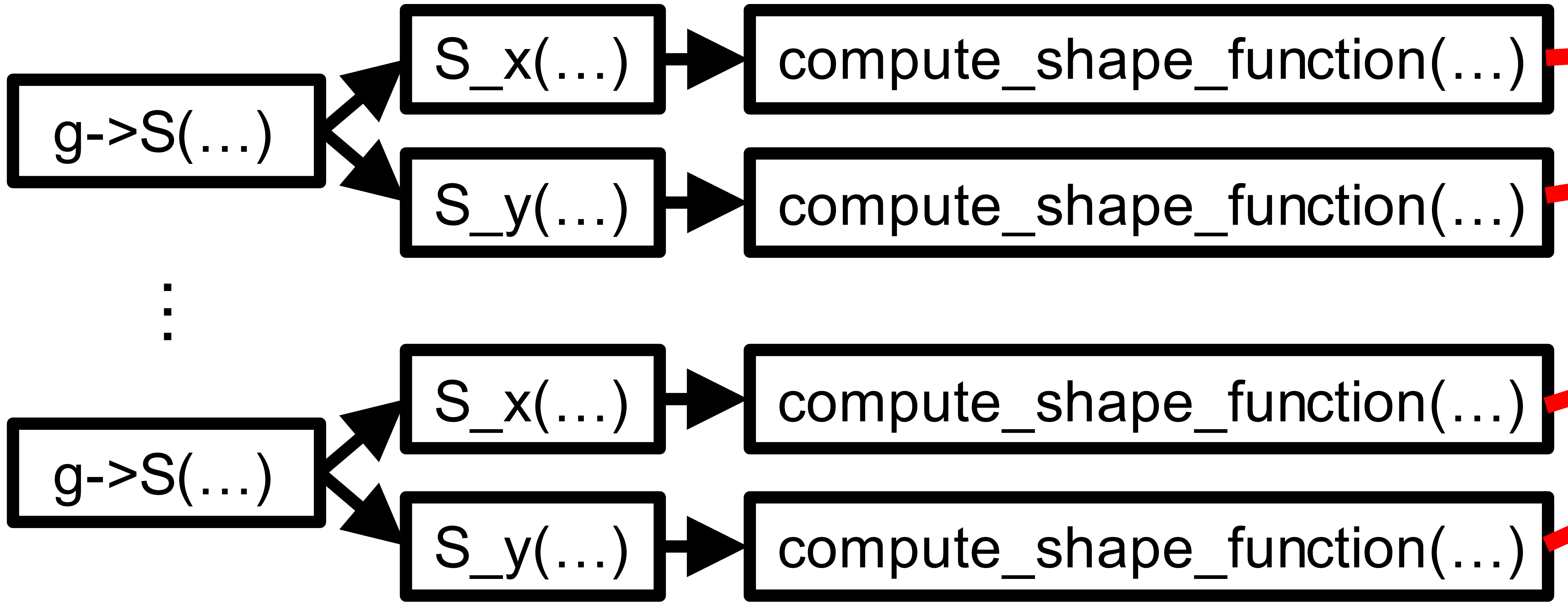
File: MPM.cpp

```
for(unsigned ii=i; ii<=i+1; ii++){
  for(unsigned jj=j; jj<=j+1; jj++){
    g->mass(ii,jj) += g->S(ii, jj, mp->position(p))*mp->mass(p);
    g->momentum(ii,jj) += g->S(ii, jj, mp->position(p))*mp->mass(p)*mp->velocity(p);
  }
}
```



File: Grid.h

```
double S(int i, int j, const Point& p){ ... }
unsigned indexify(unsigned i, unsigned j) const { ... }
double S_x(int i, double x){ ... }
double S_y(int j, double y){ ... }
static double compute_shape_function(int cell, double position, double cell_size){
  // This is the distance of "position" from the position of "cell".
  const double cell_distance = std::abs((position - cell_size*cell) / cell_size);
  // Perform case analysis.
  if(cell_distance >= 1.0){
    return 0.0;
  }
  else{
    return 1.0 - cell_distance;
  }
}
```



g->S(...)

S_x(...)

compute_shape_function(...)

S_y(...)

compute_shape_function(...)

⋮

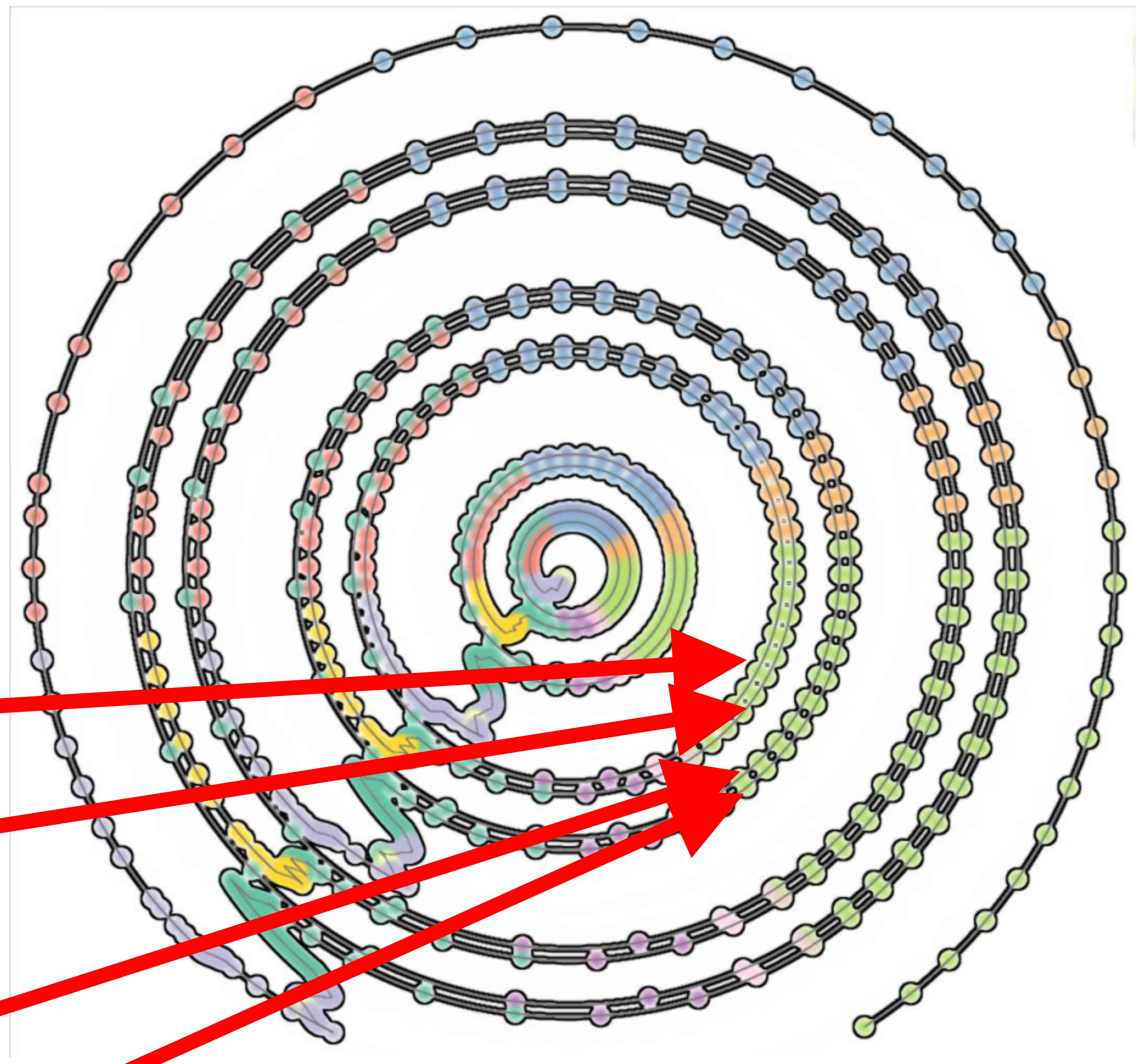
g->S(...)

S_x(...)

compute_shape_function(...)

S_y(...)

compute_shape_function(...)

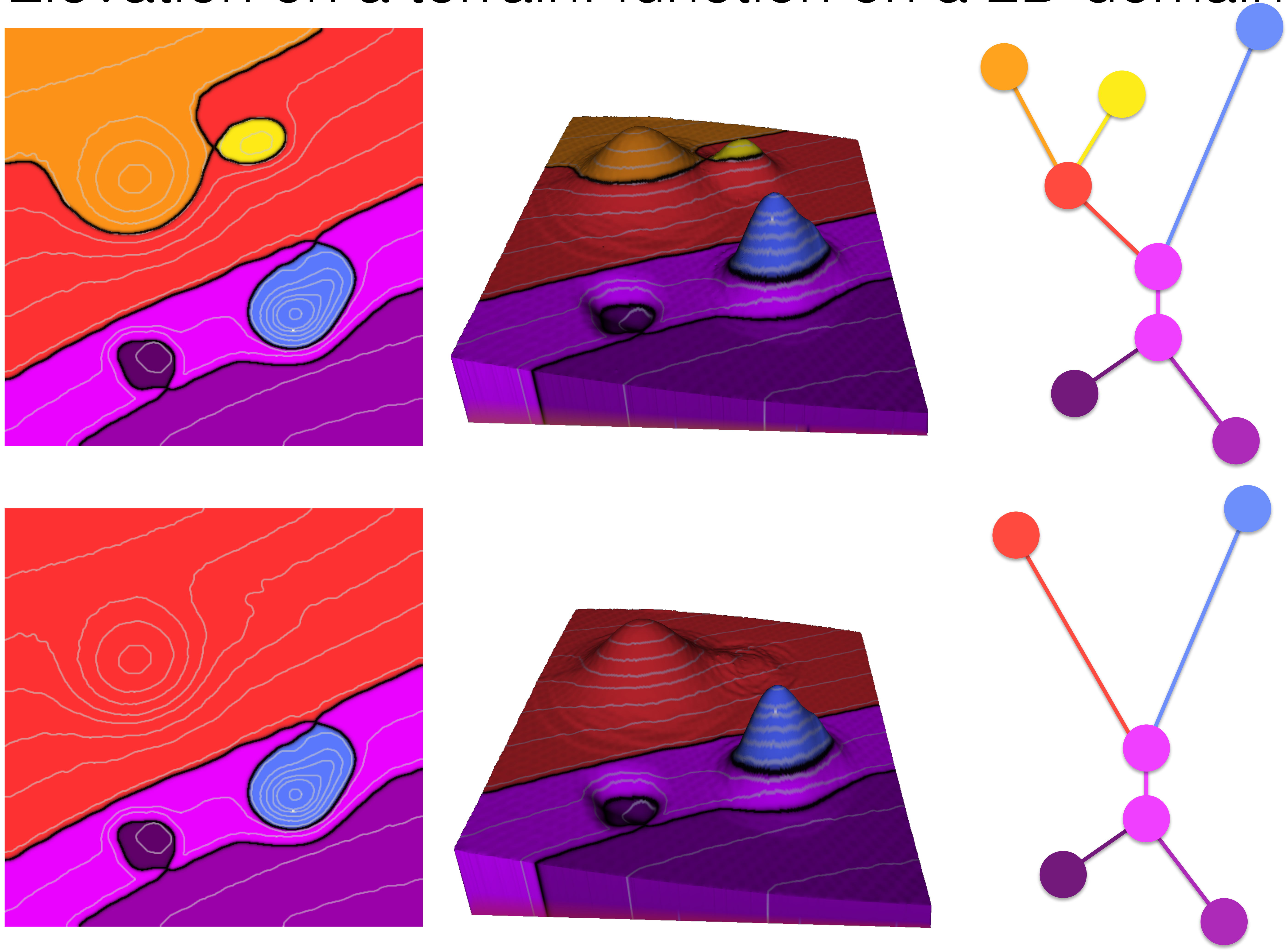


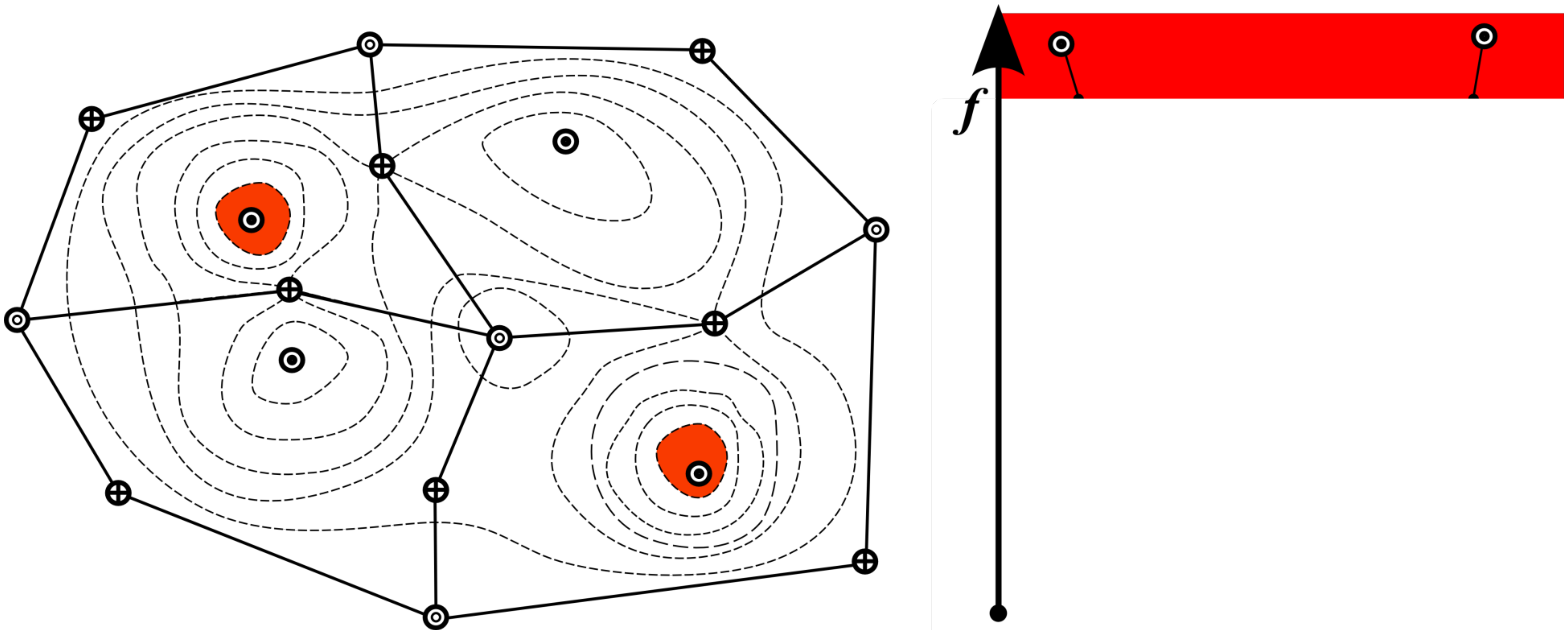
Contour Tree

A review

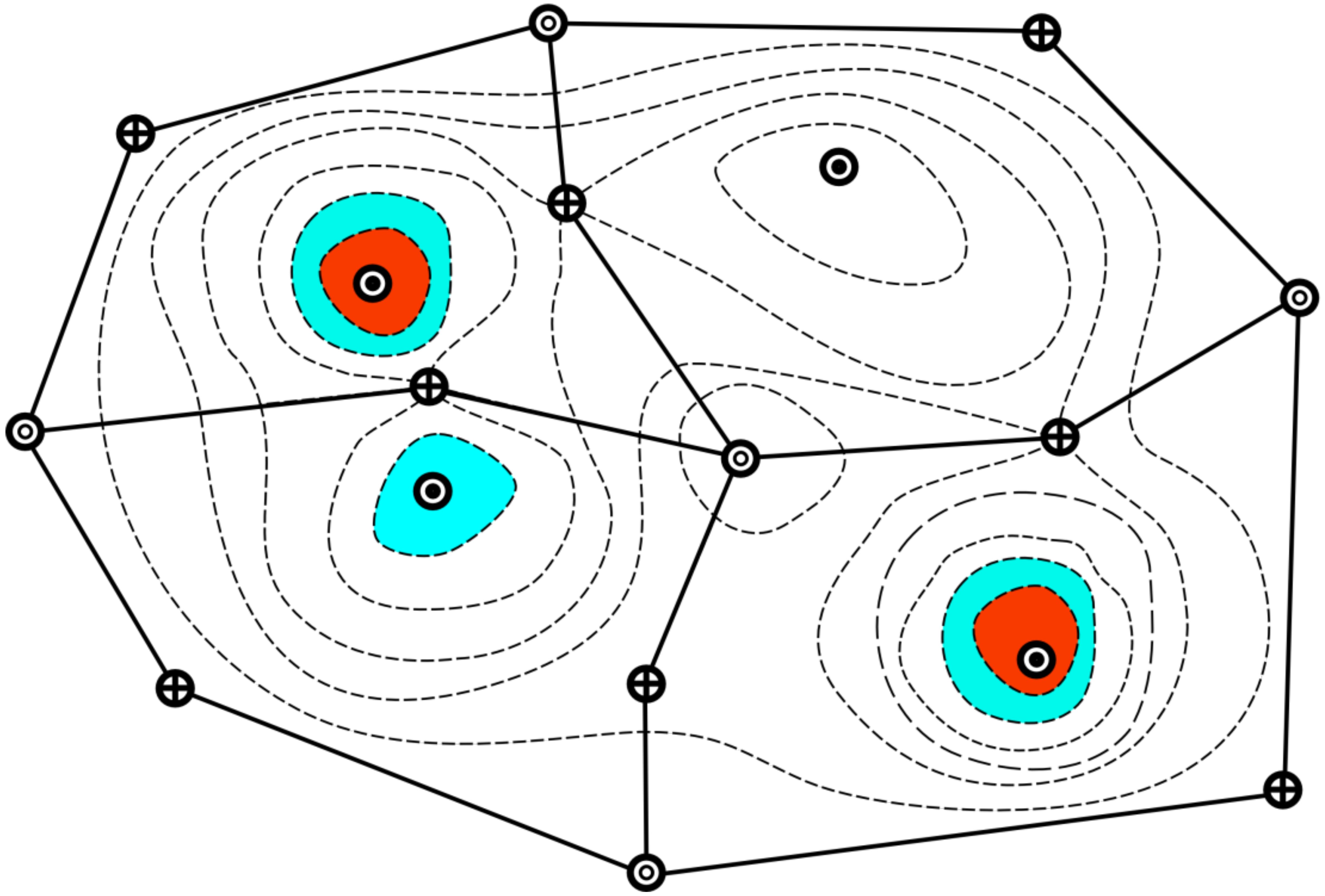
Contour tree revisited

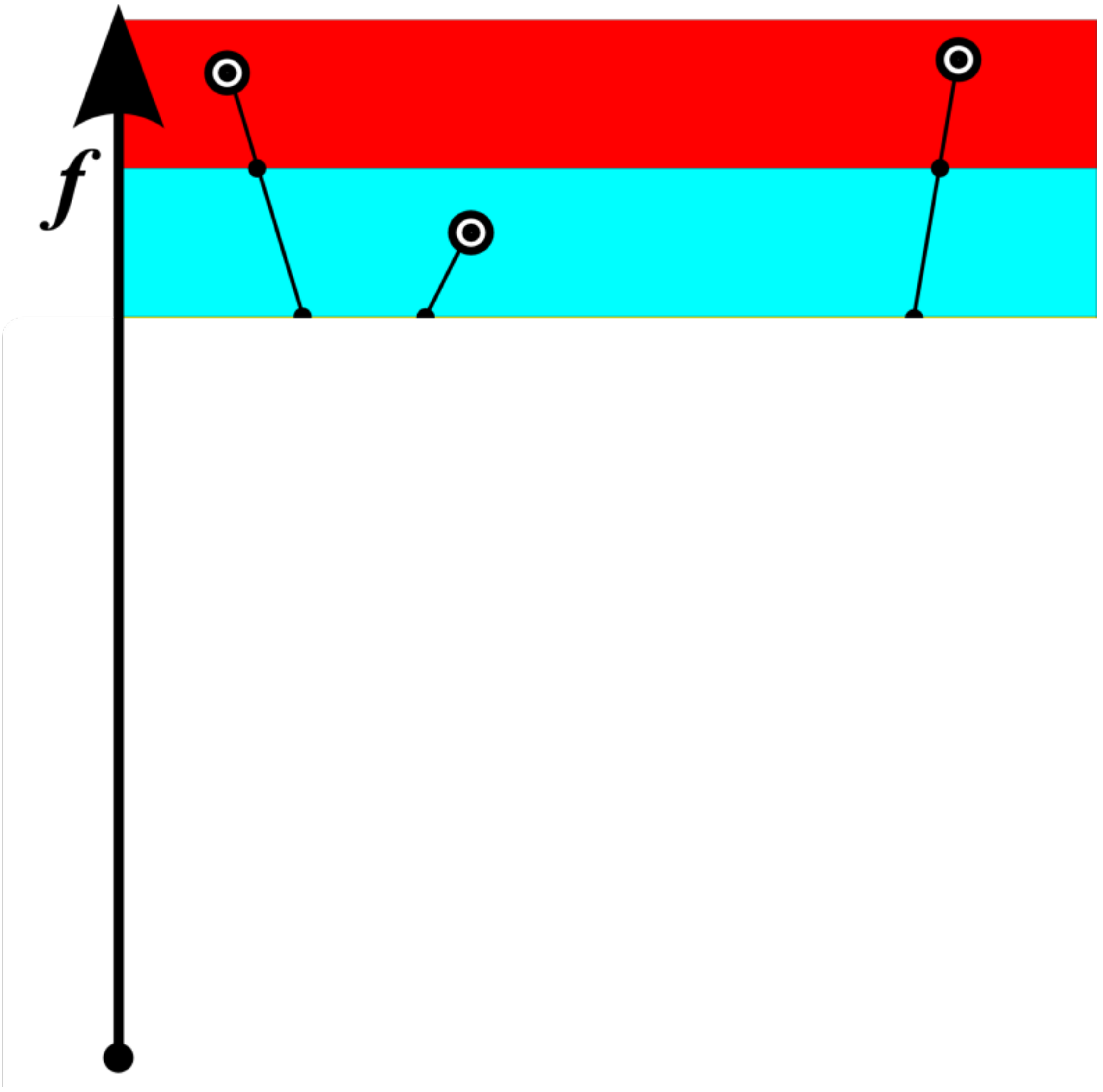
Elevation on a terrain: function on a 2D domain

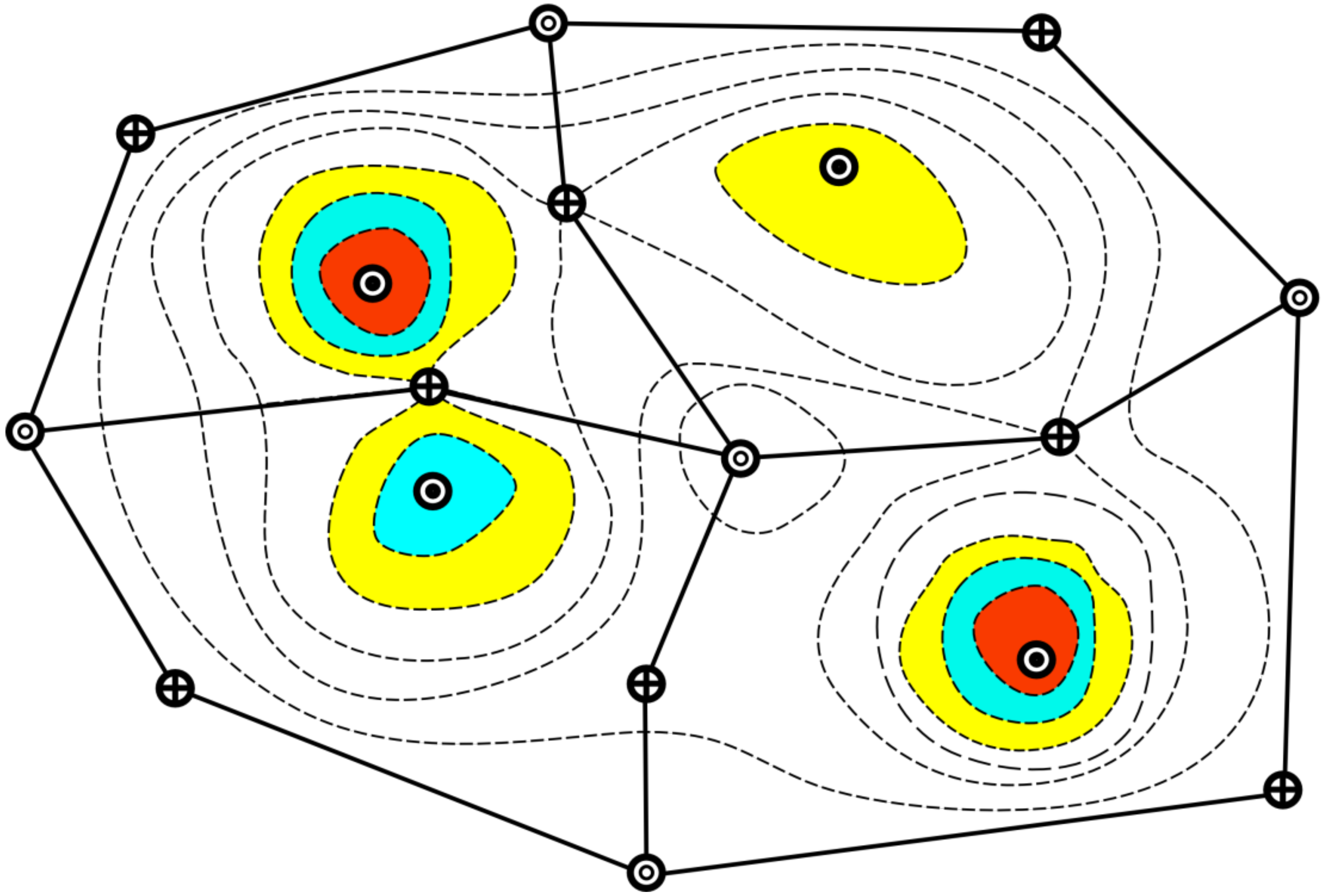


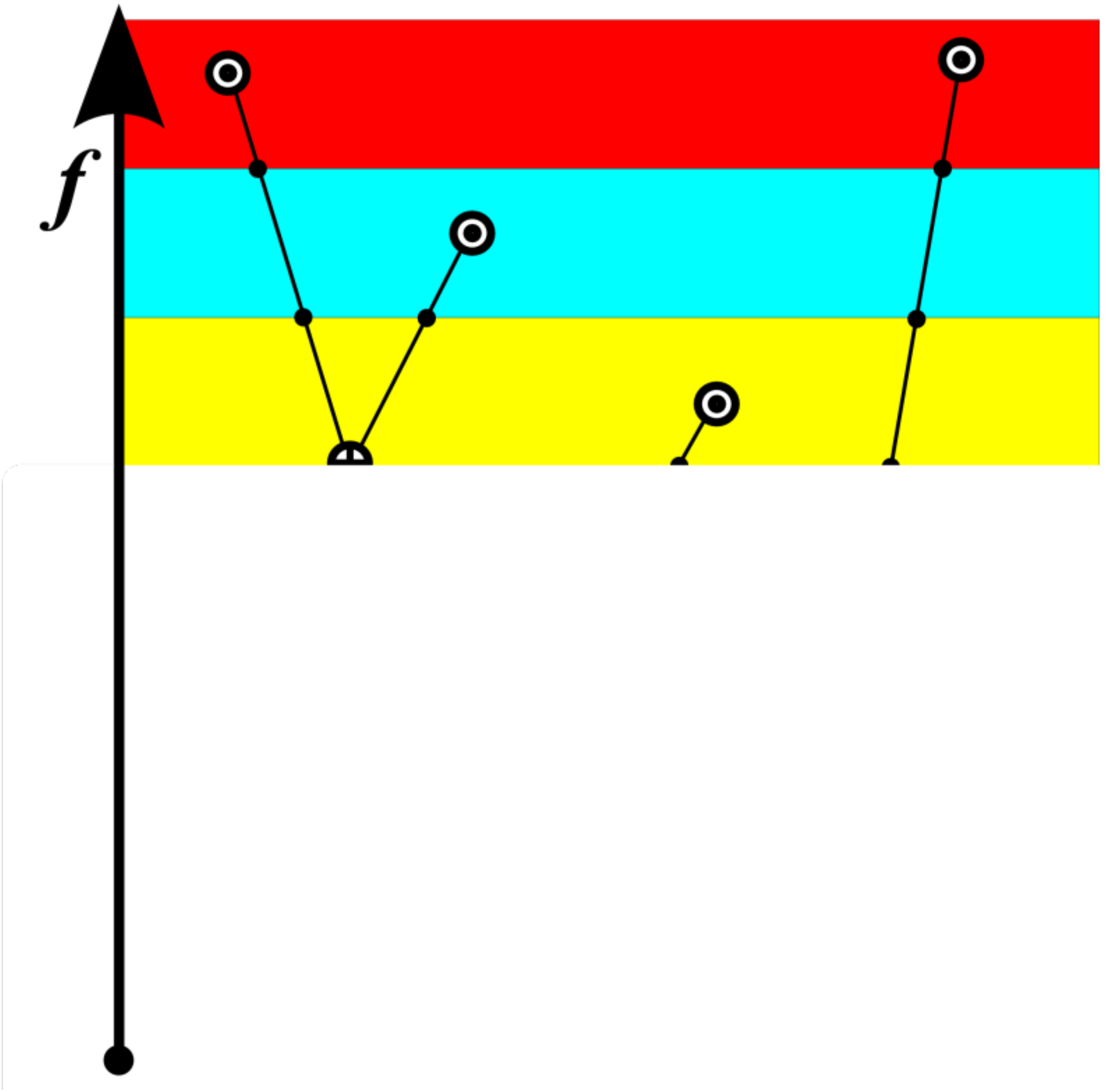


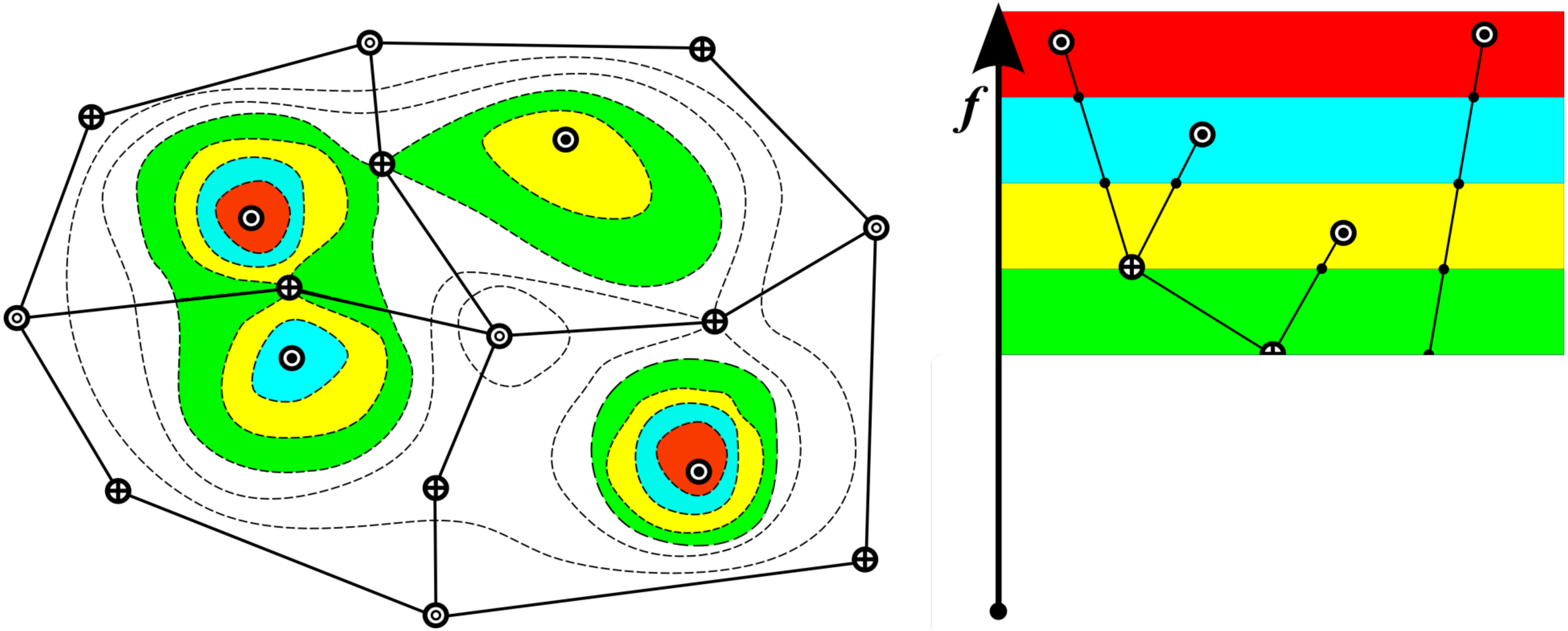
Source: Valerio Pascucci

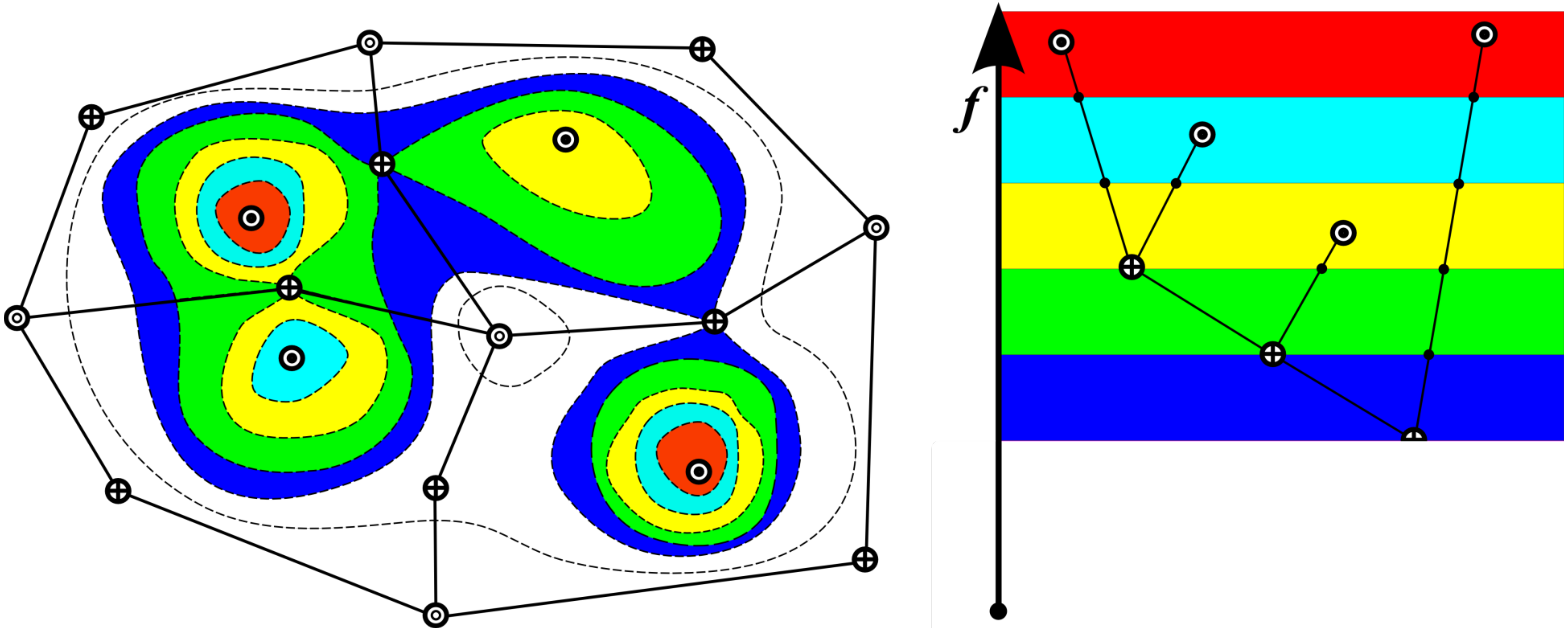


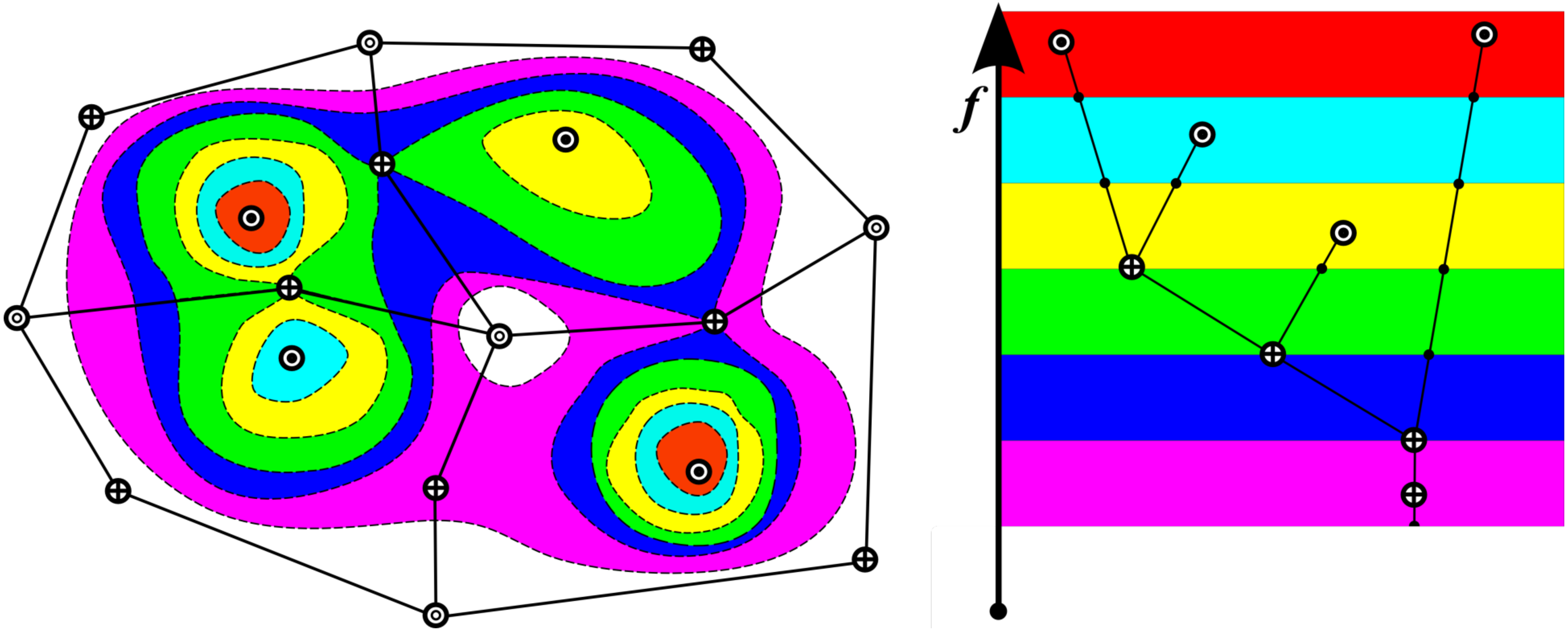

f

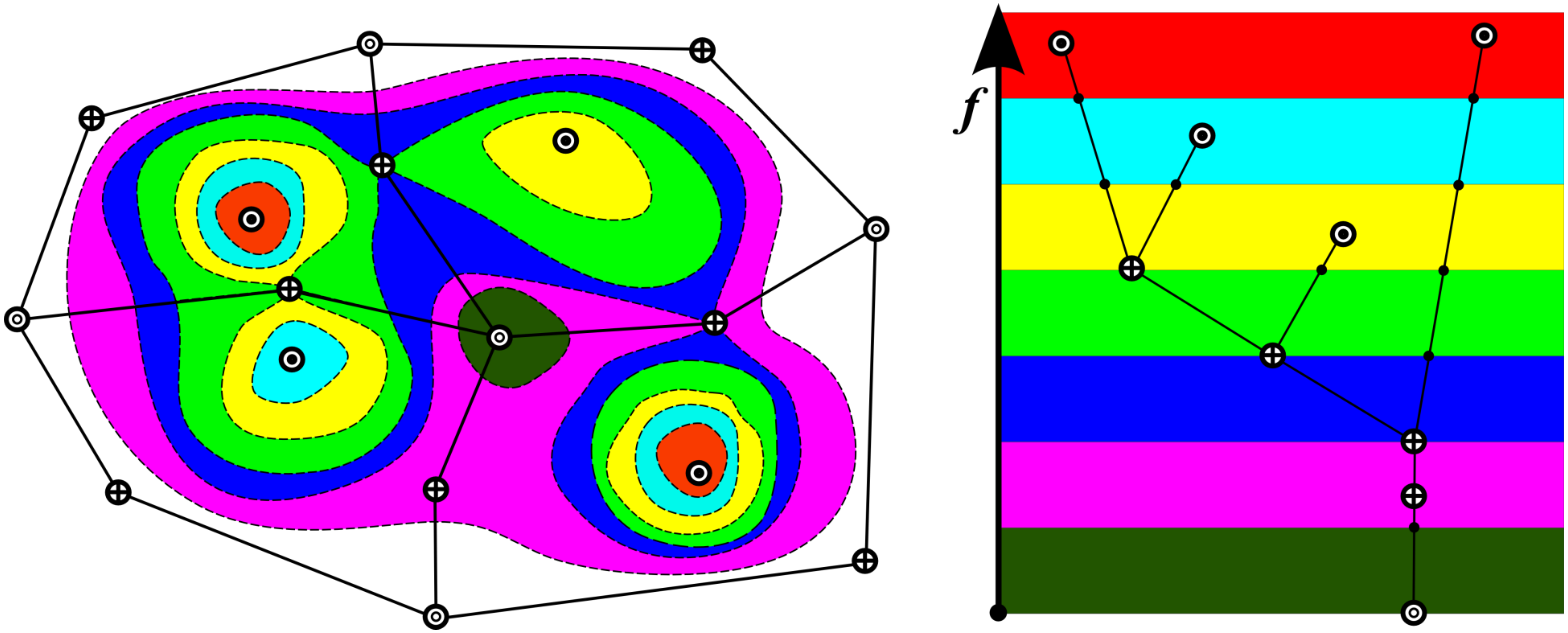



f


f


f

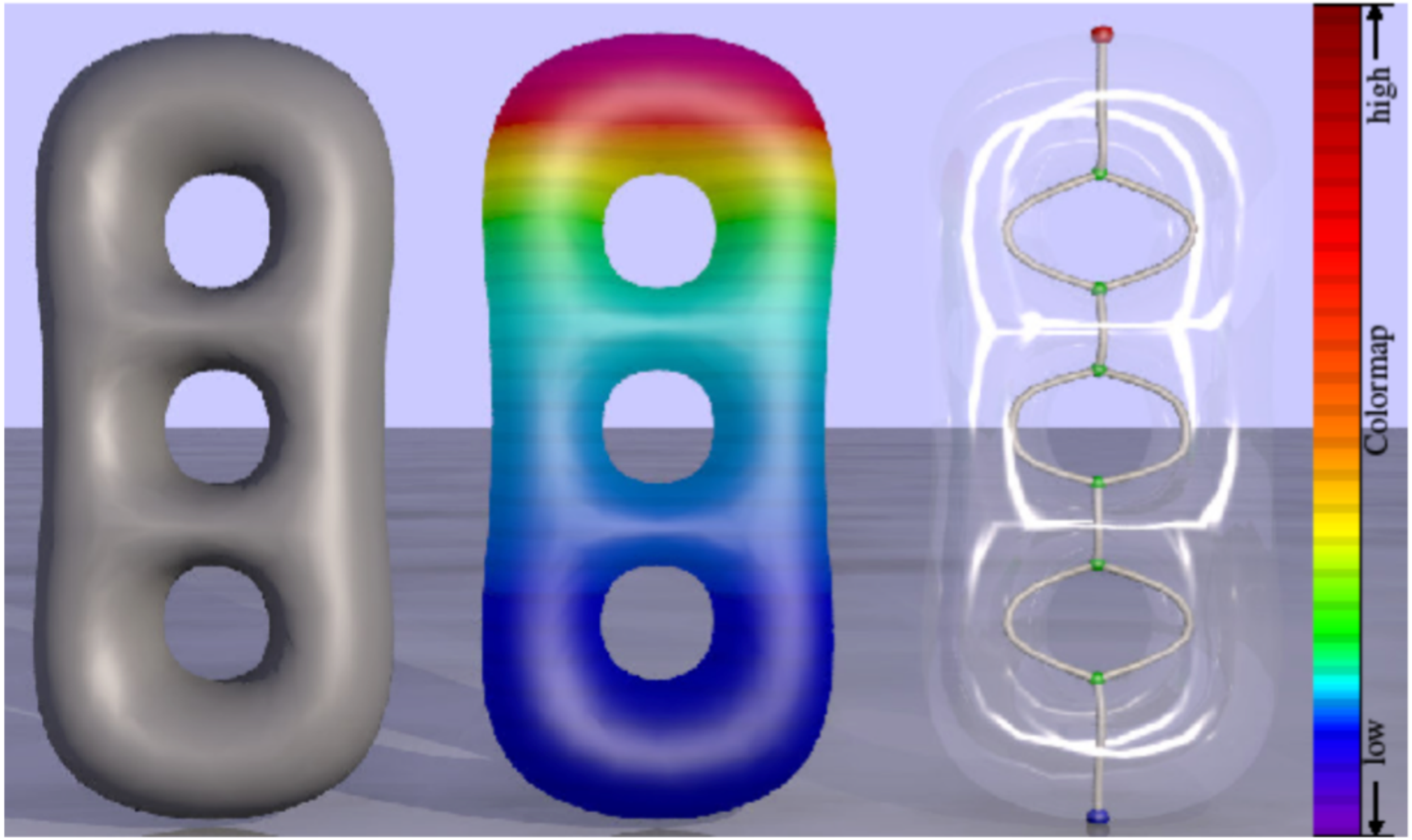

f


f

Reeb Graph

A generalization of contour tree

Graph obtained by continuos contraction of all the contours in a scalar field, where each contour is collapsed to a distinct point.

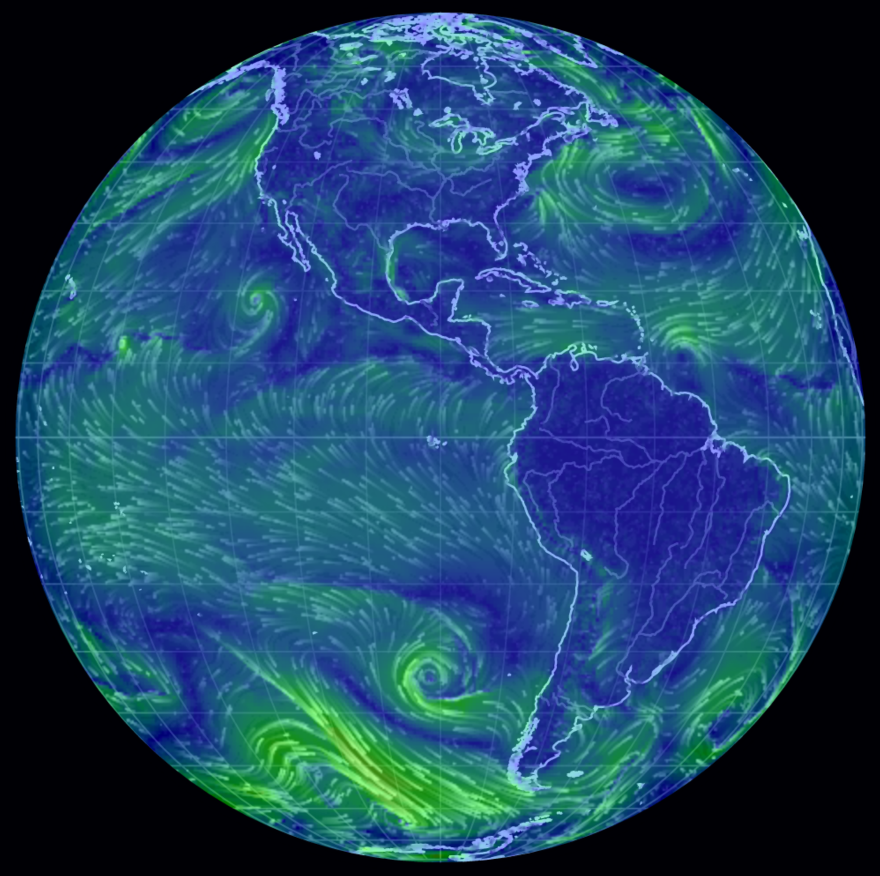


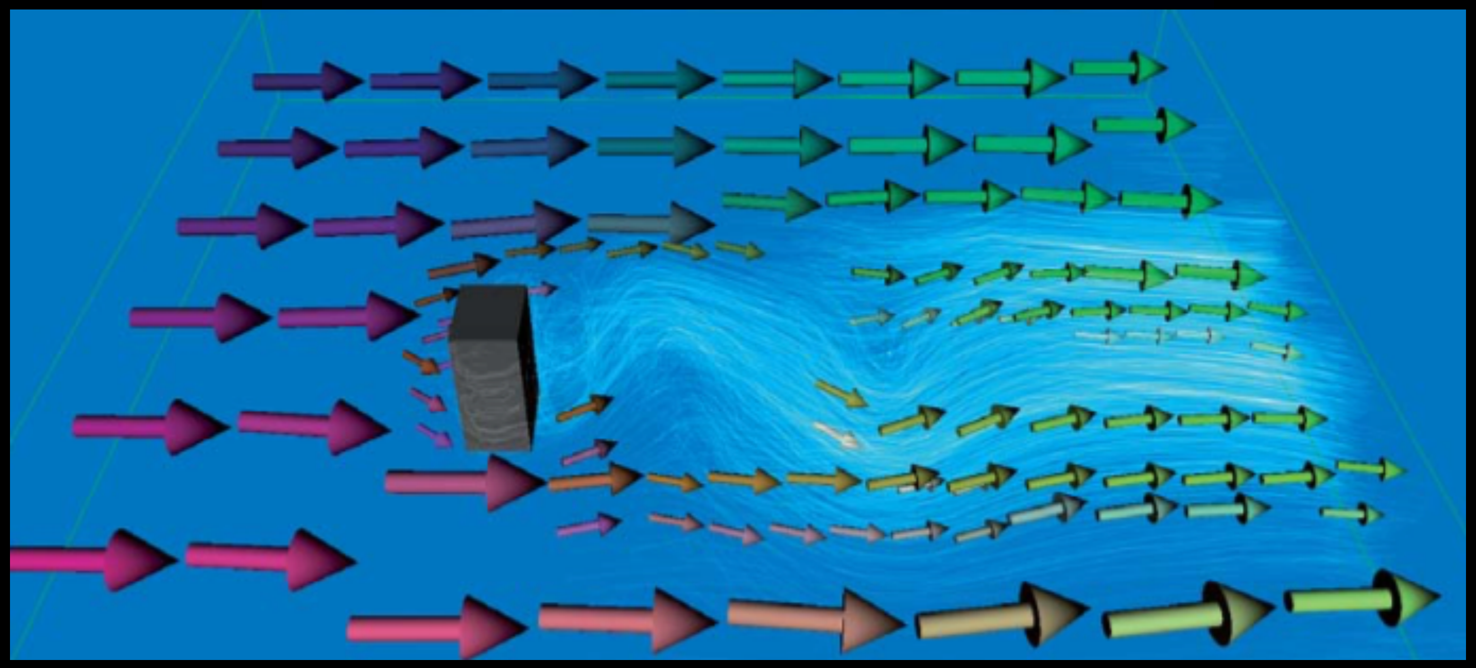
[K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan and V. Pascucci, 2004]

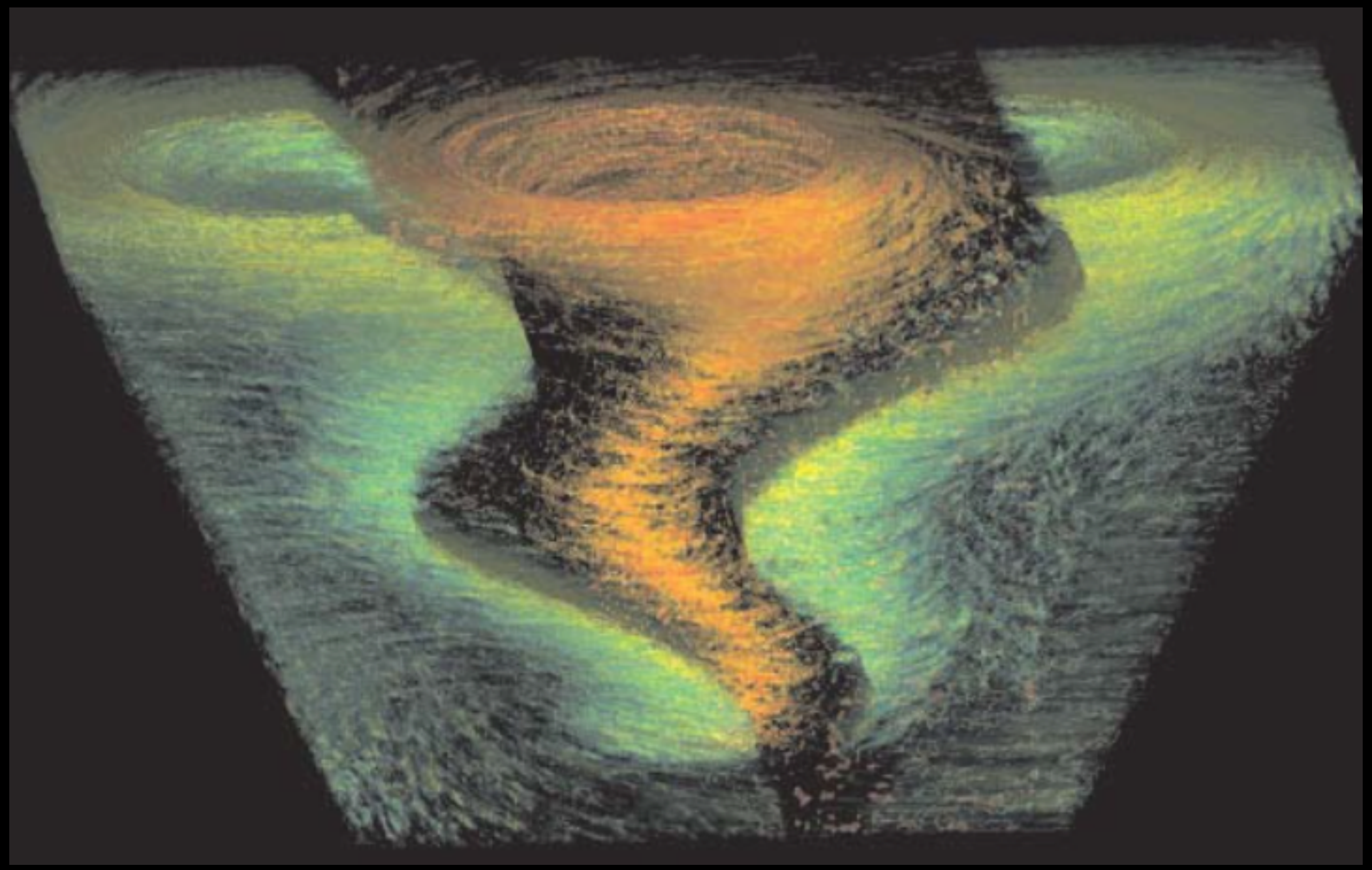
Case Study 1: Vector Fields
Combustion and Ocean

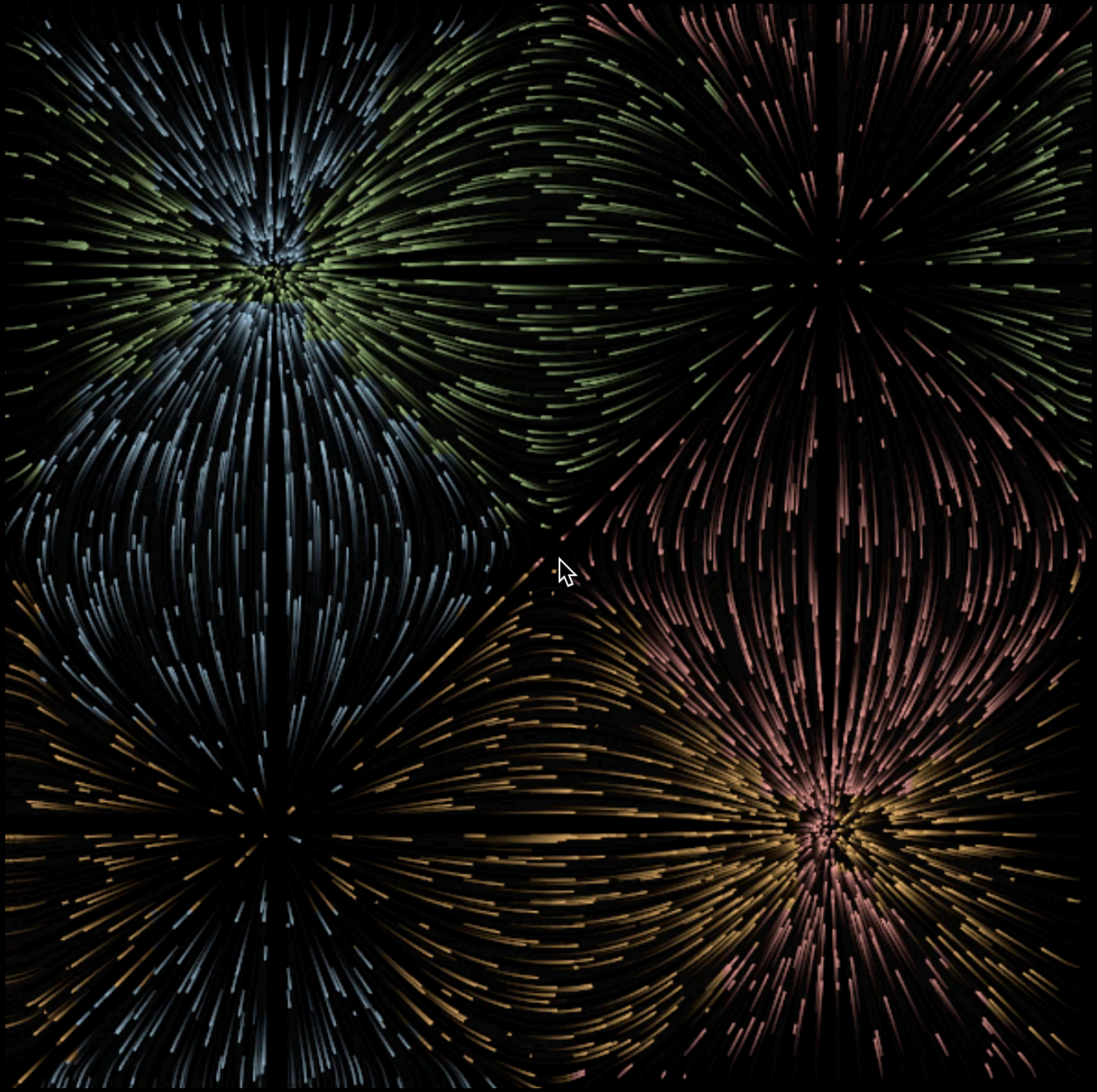
Application of contour tree

Make the flow patterns visible & Interpretable



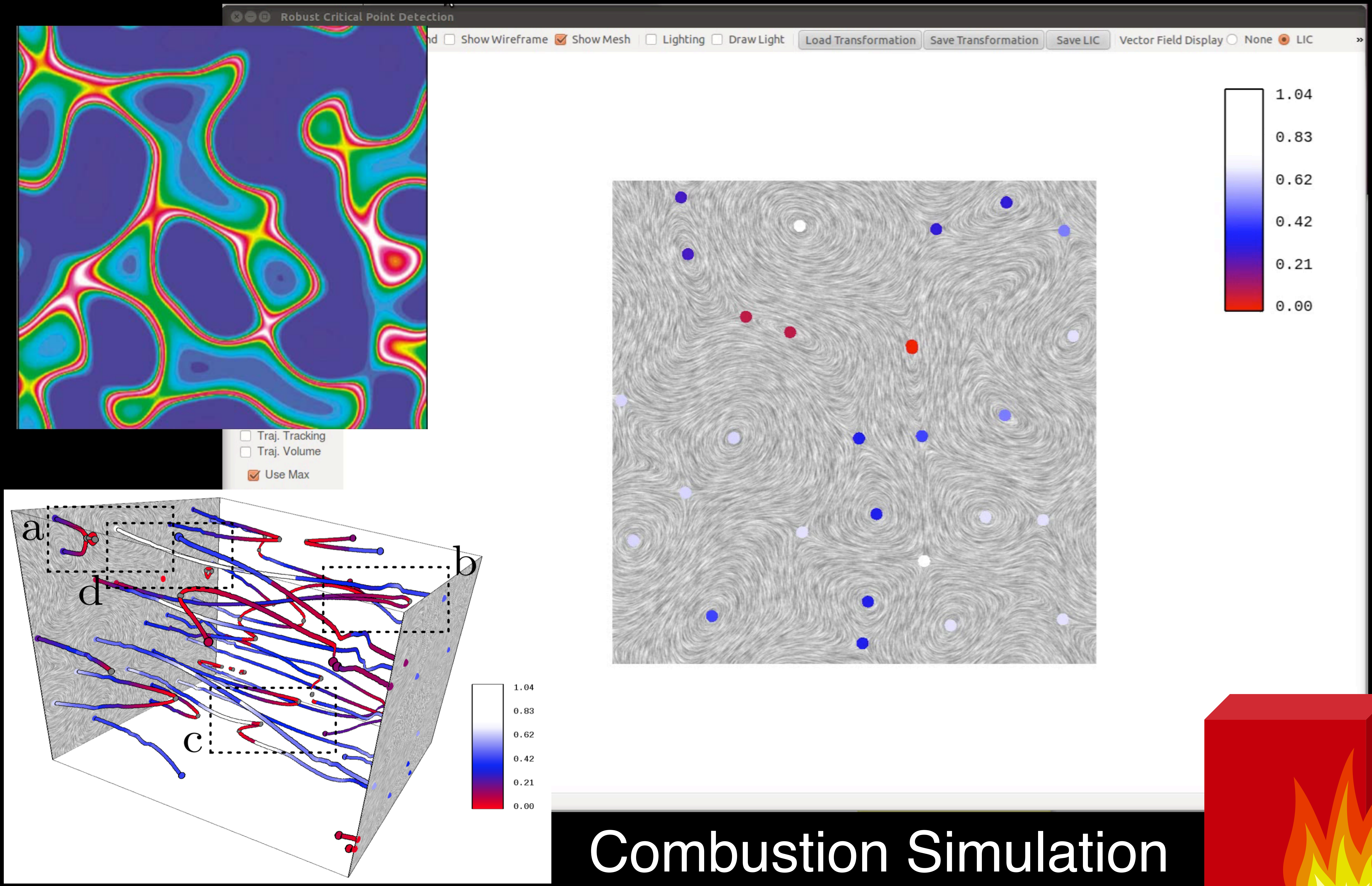






Sources: Dan Maljovec, Cameron Beccario, [Correa, Silver, Chen 2007]
[Burger, Kondratieva, Kruger, Westermann 2008]

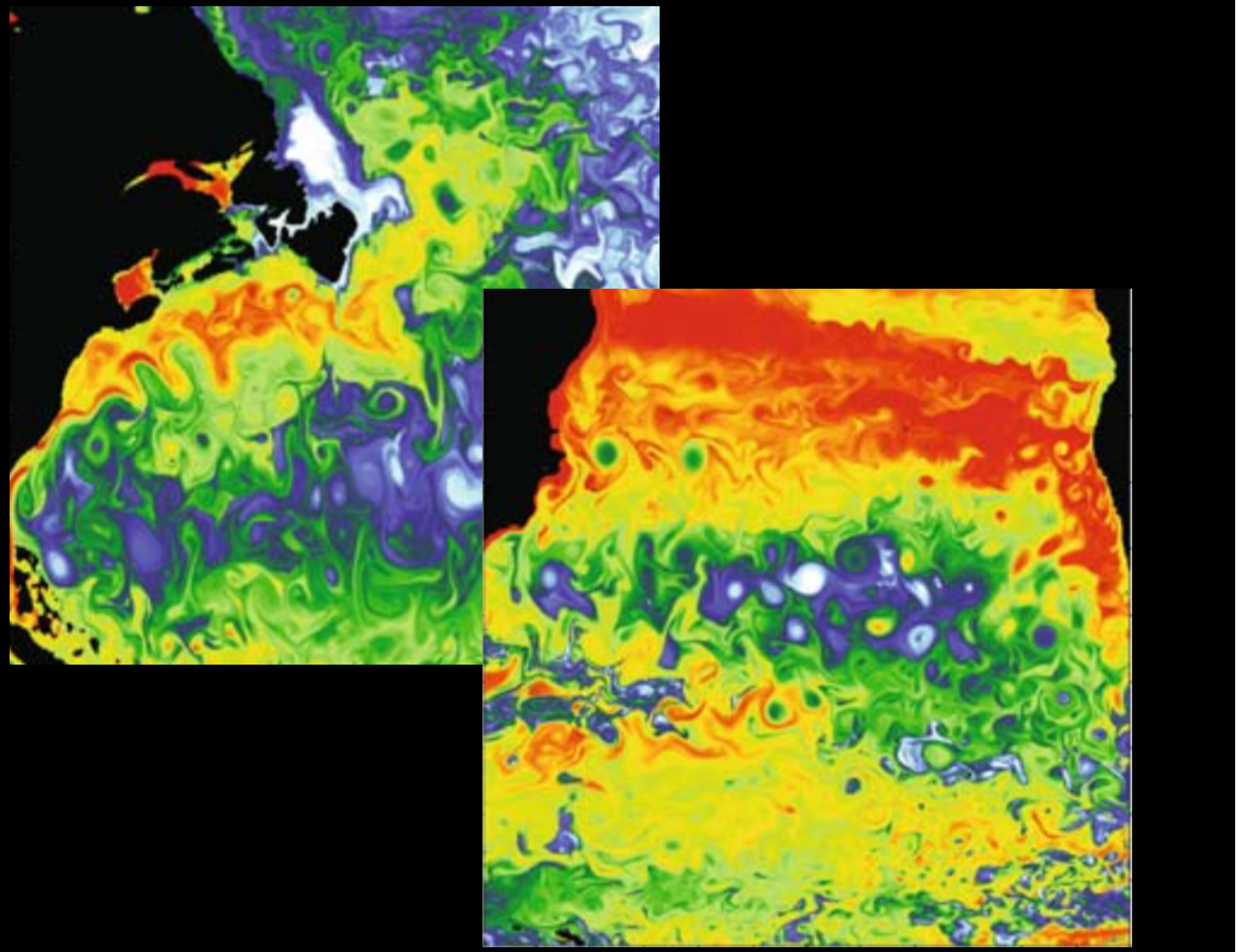
Quantify feature stability

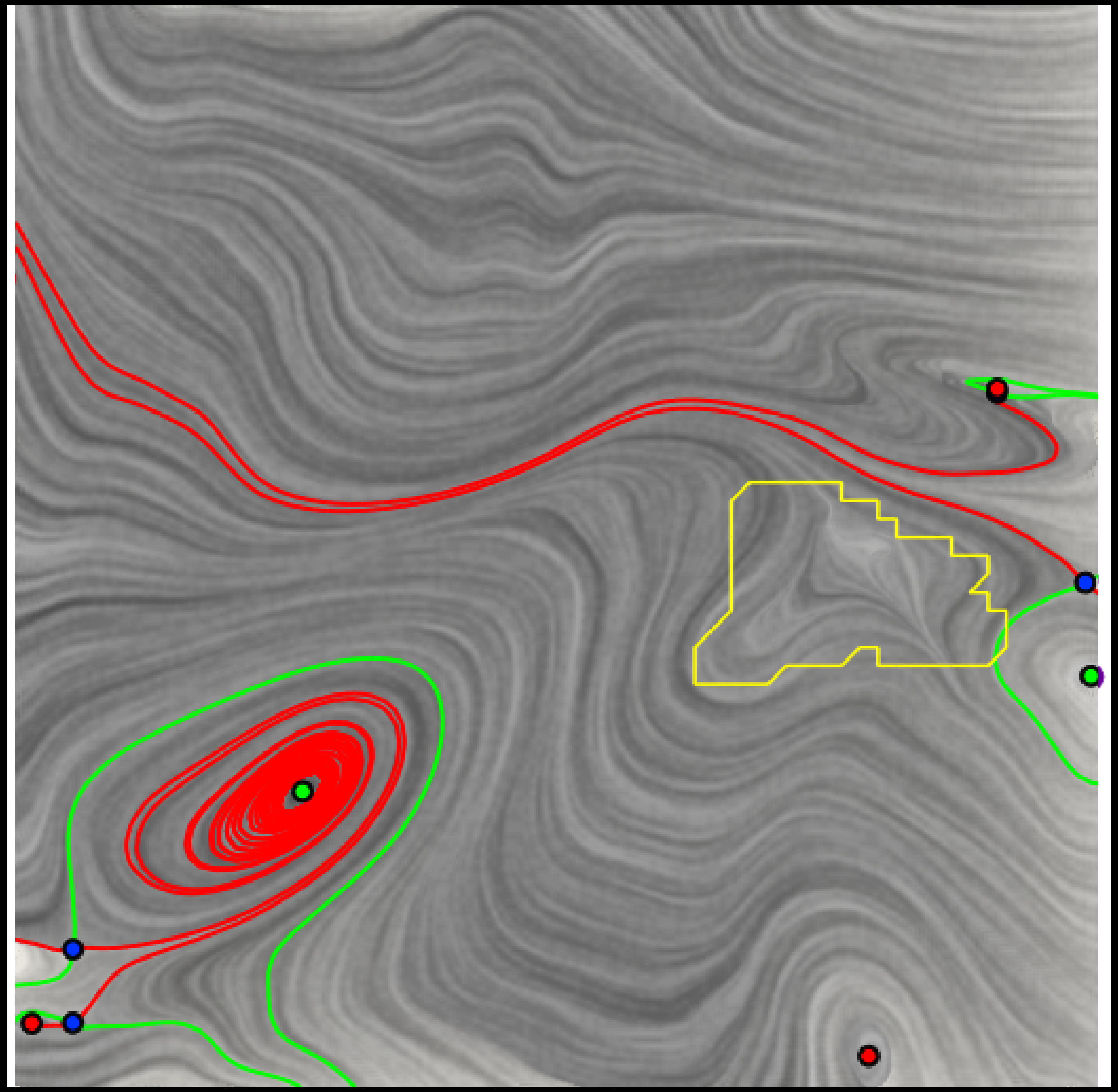


Combustion Simulation

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

Separate features from noise at multi-scale

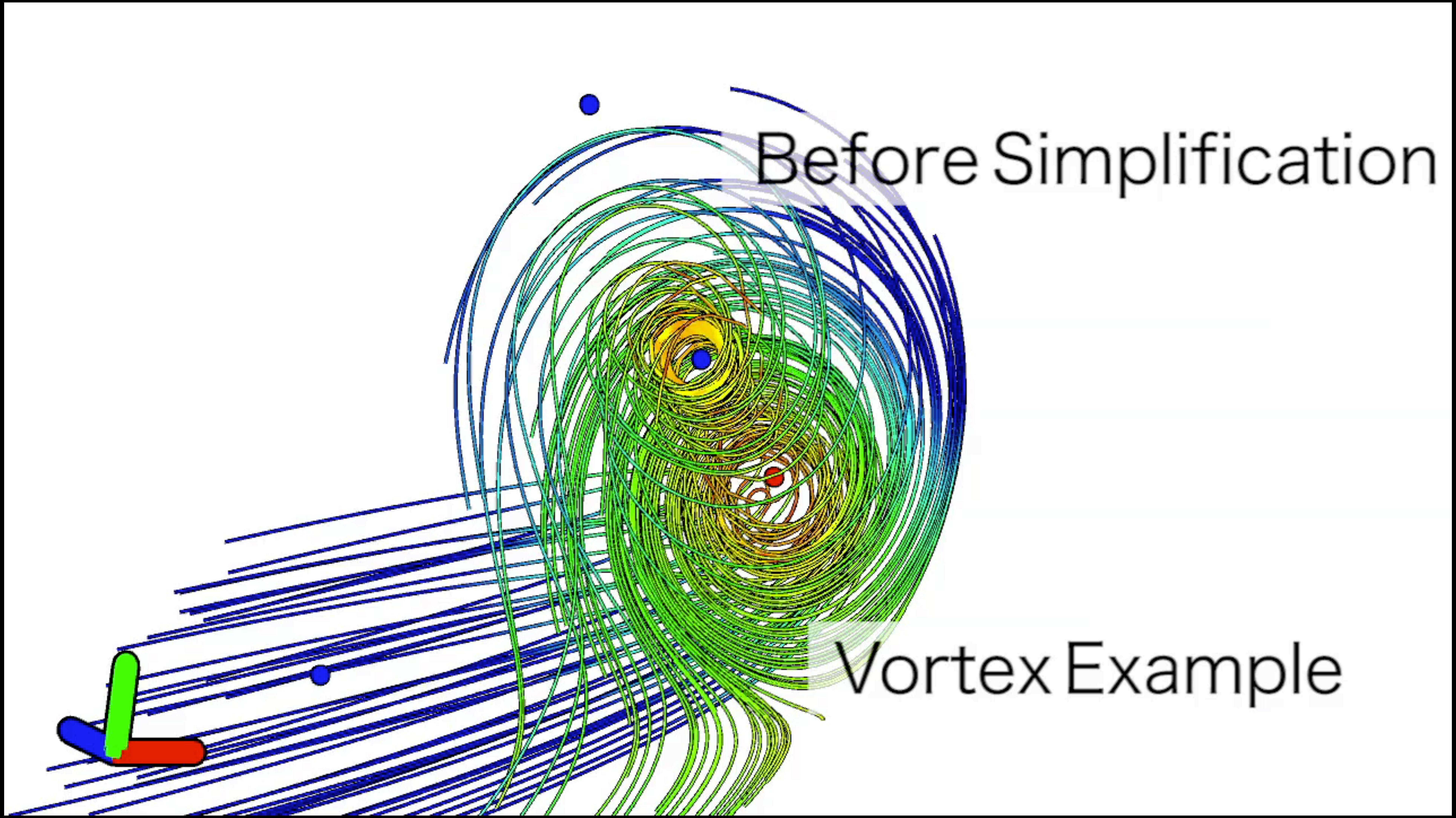




Ocean Eddy Simulation

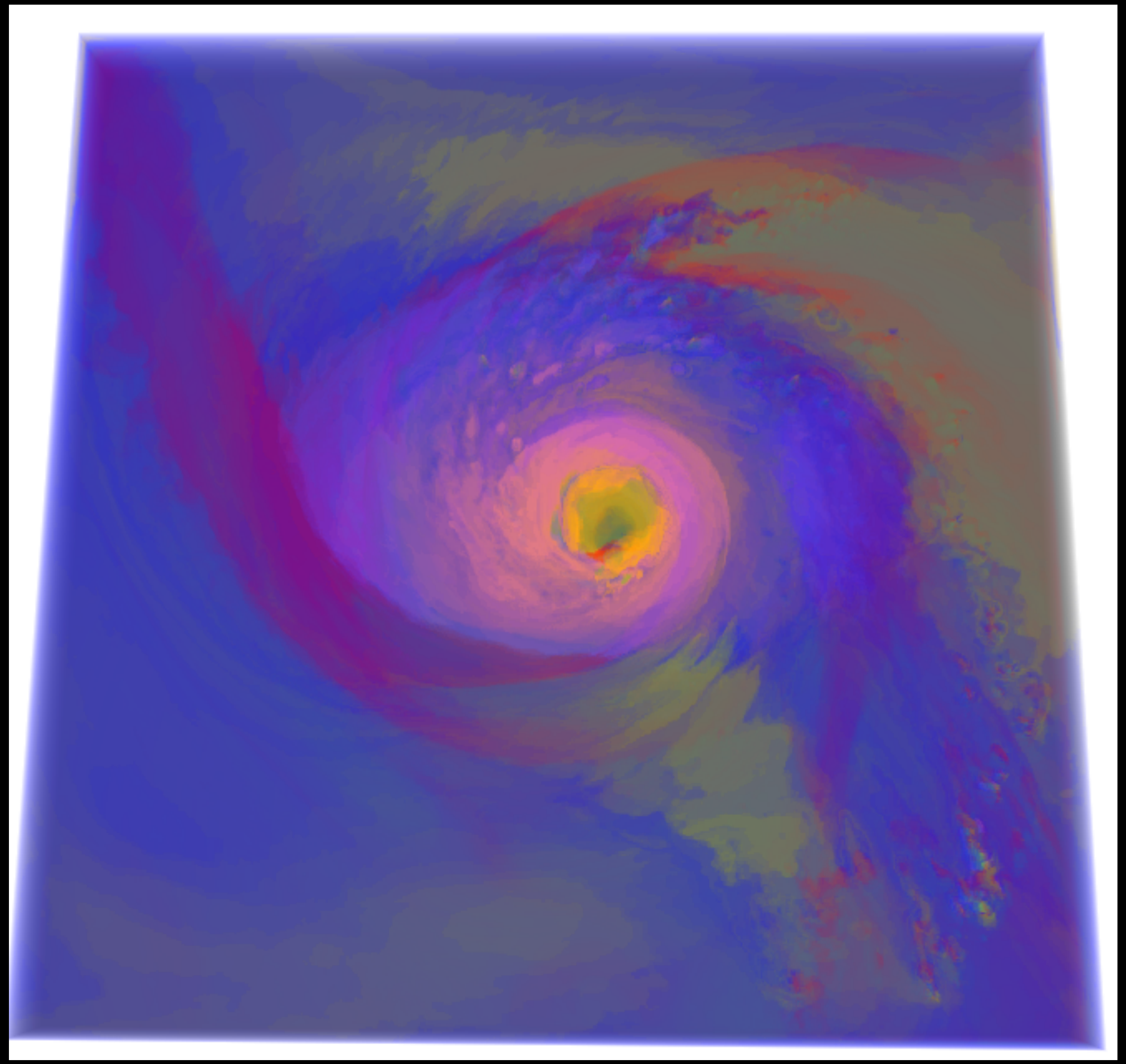
Map: Courtesy of SlidesCarnival & Unsplash
Simulation: [Maltrud, Bryan, Peacock 2010]

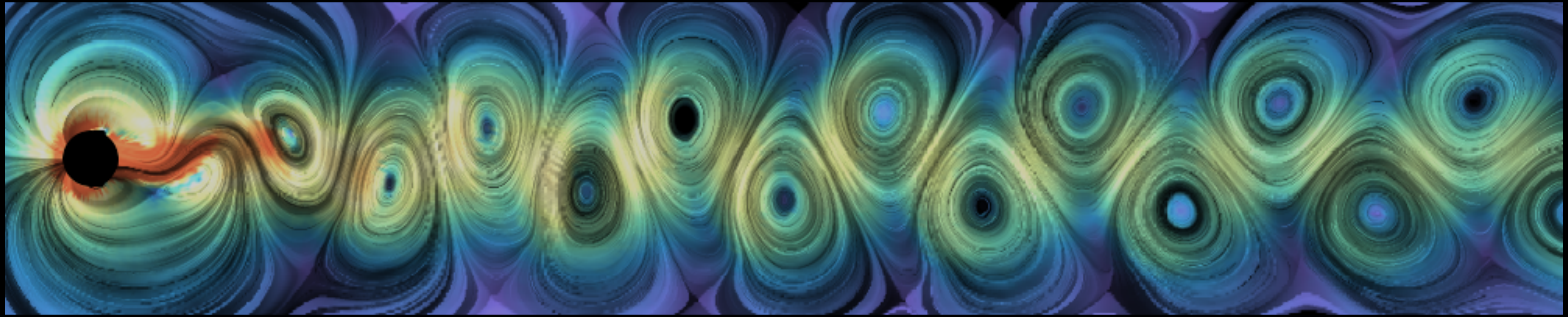
Visualize flow in 3D



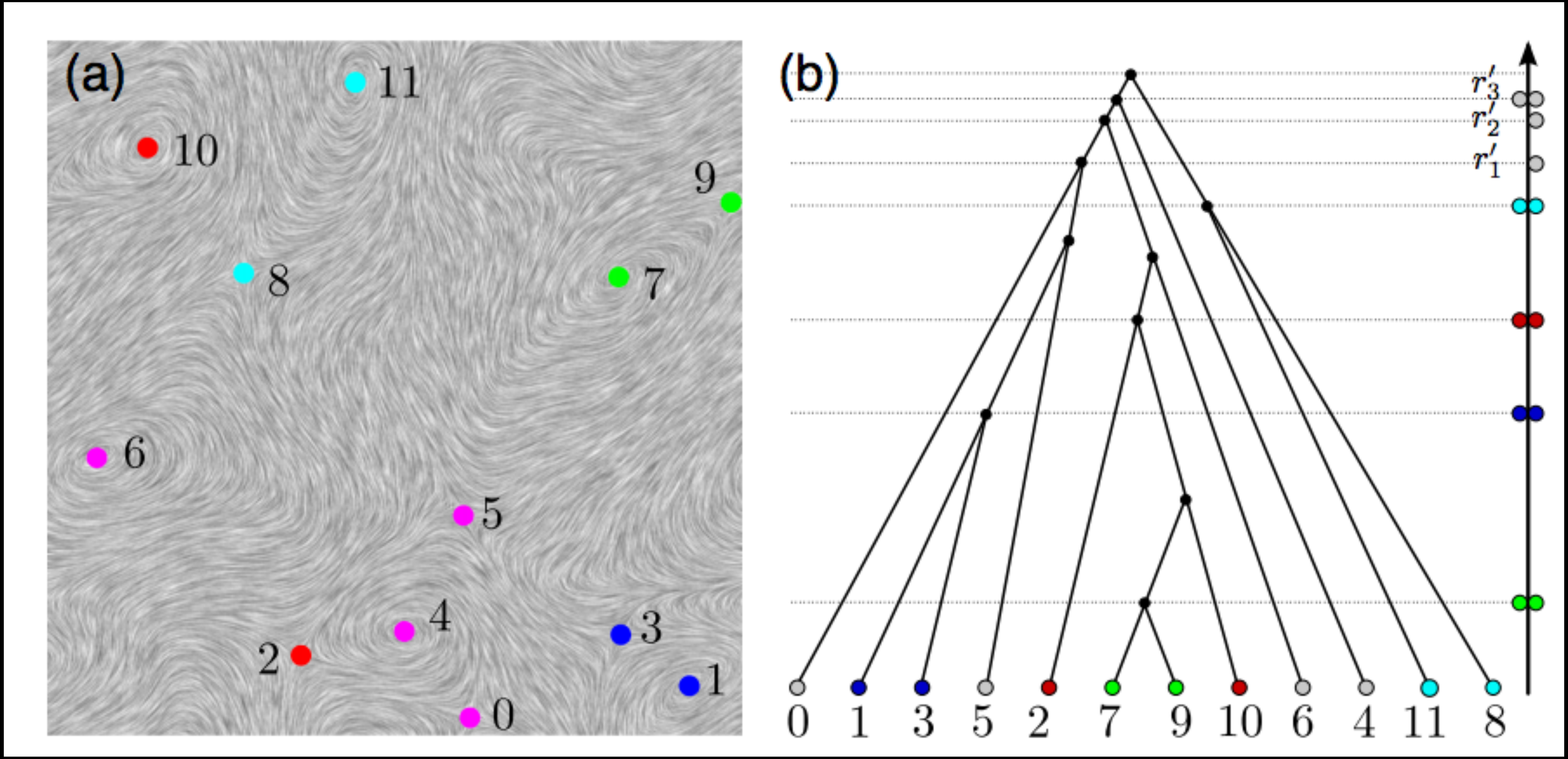
Understand turbulent flow

Source: NASA





Contour/Merge tree for VF data



http://www.sci.utah.edu/~beiwang/publications/EuroVis_robustness_BeiWang_2013.pdf

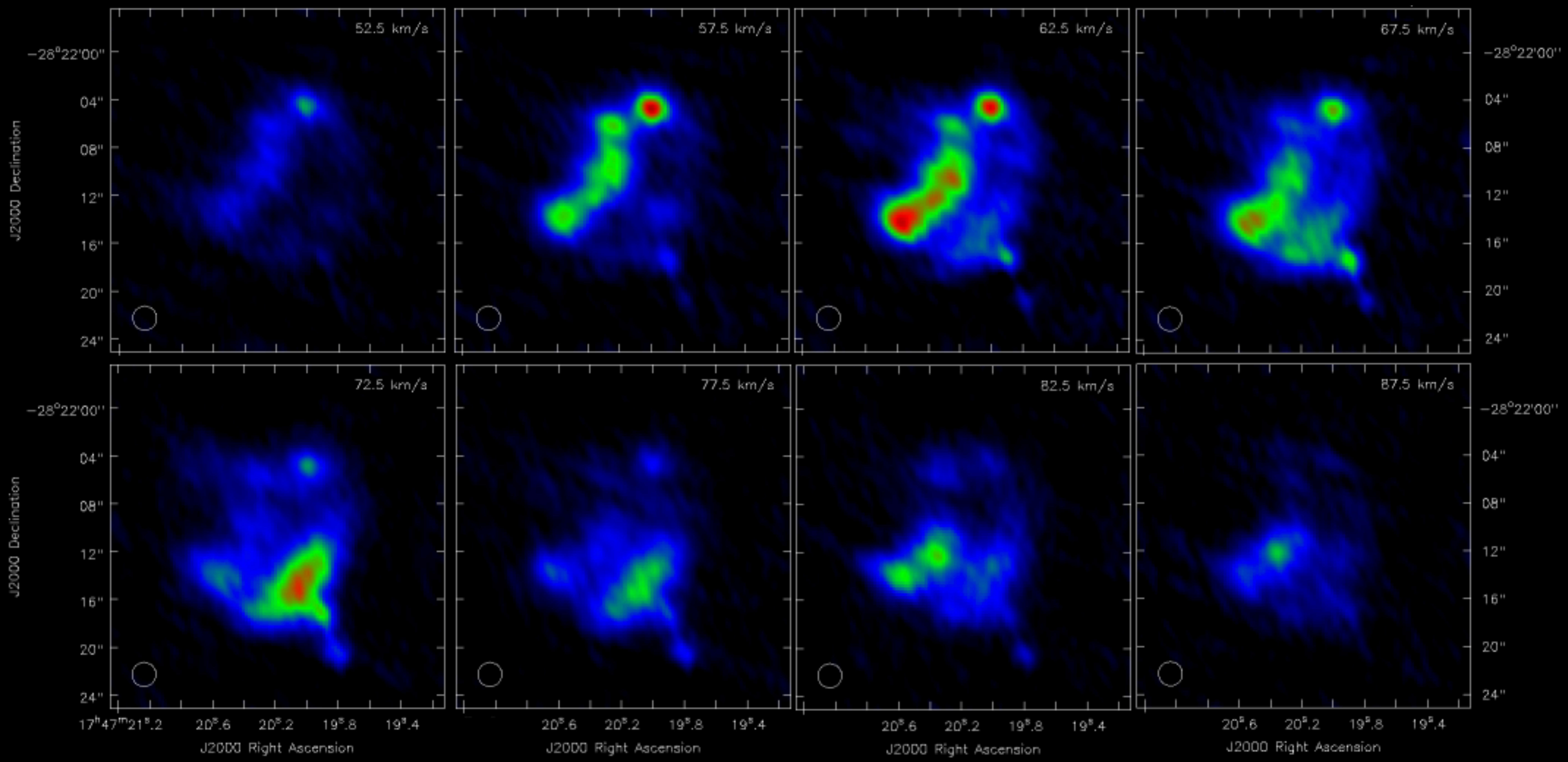
Case study 2: Astronomy
Telescopes and Black Holes

Application of Contour Tree

Largest radio telescopes in the world

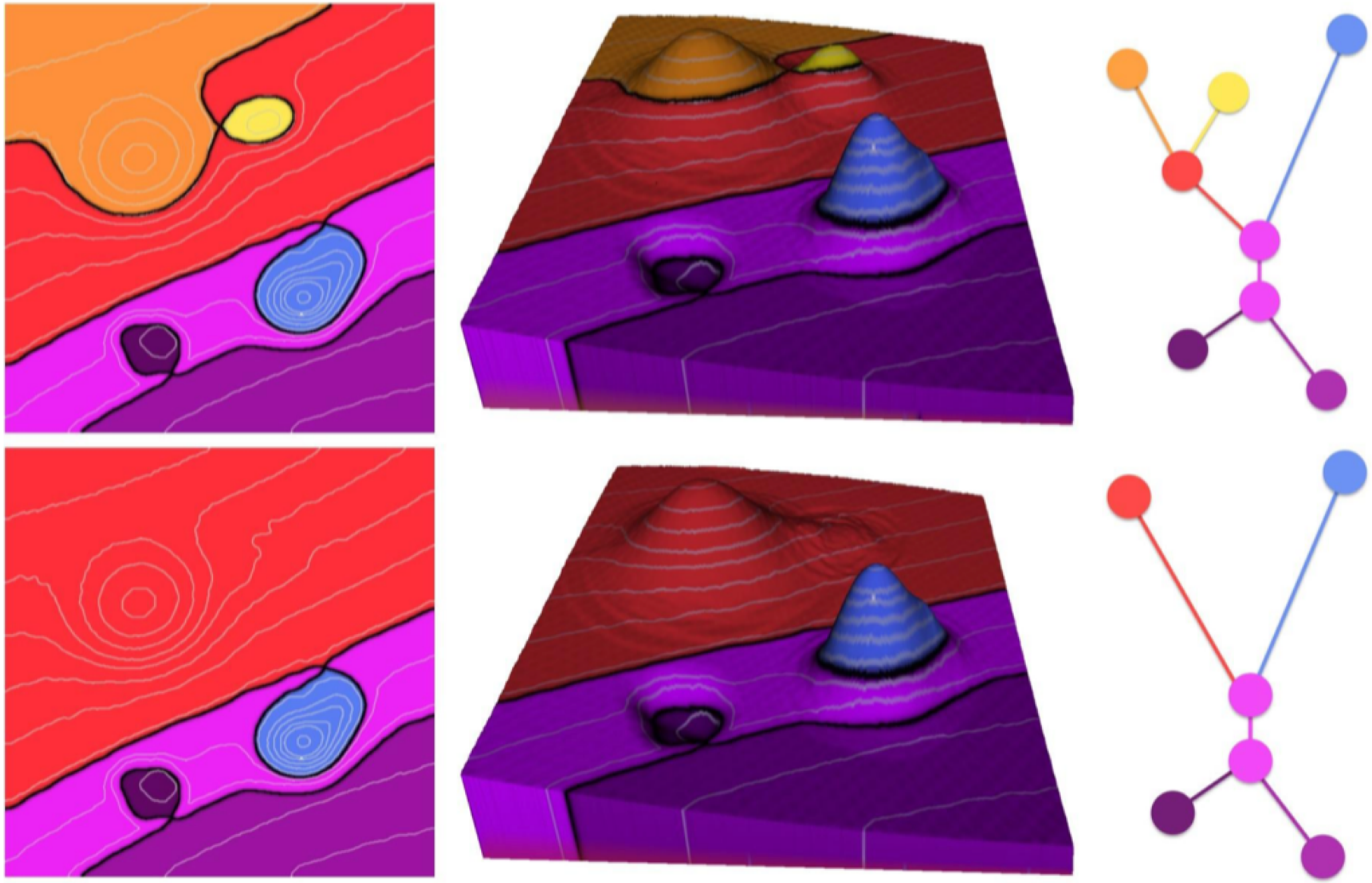
Credit: ALMA

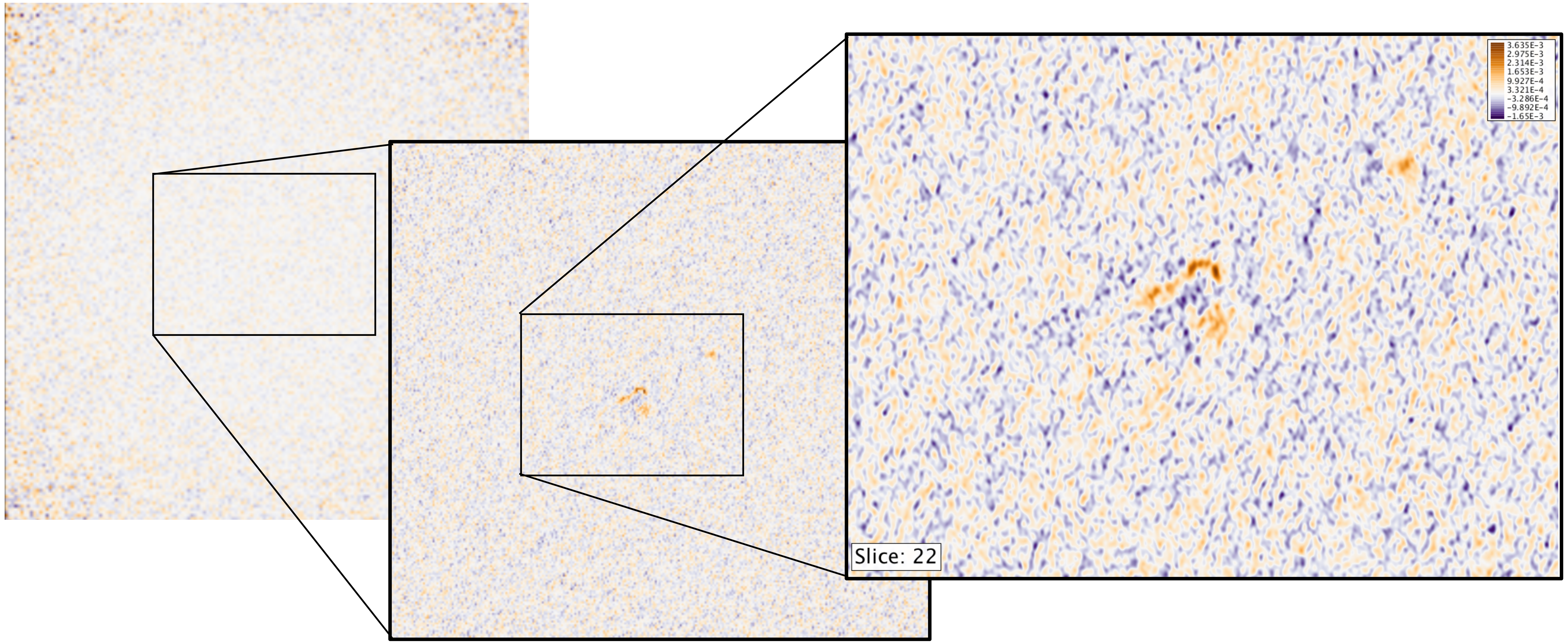
Radio telescope Data



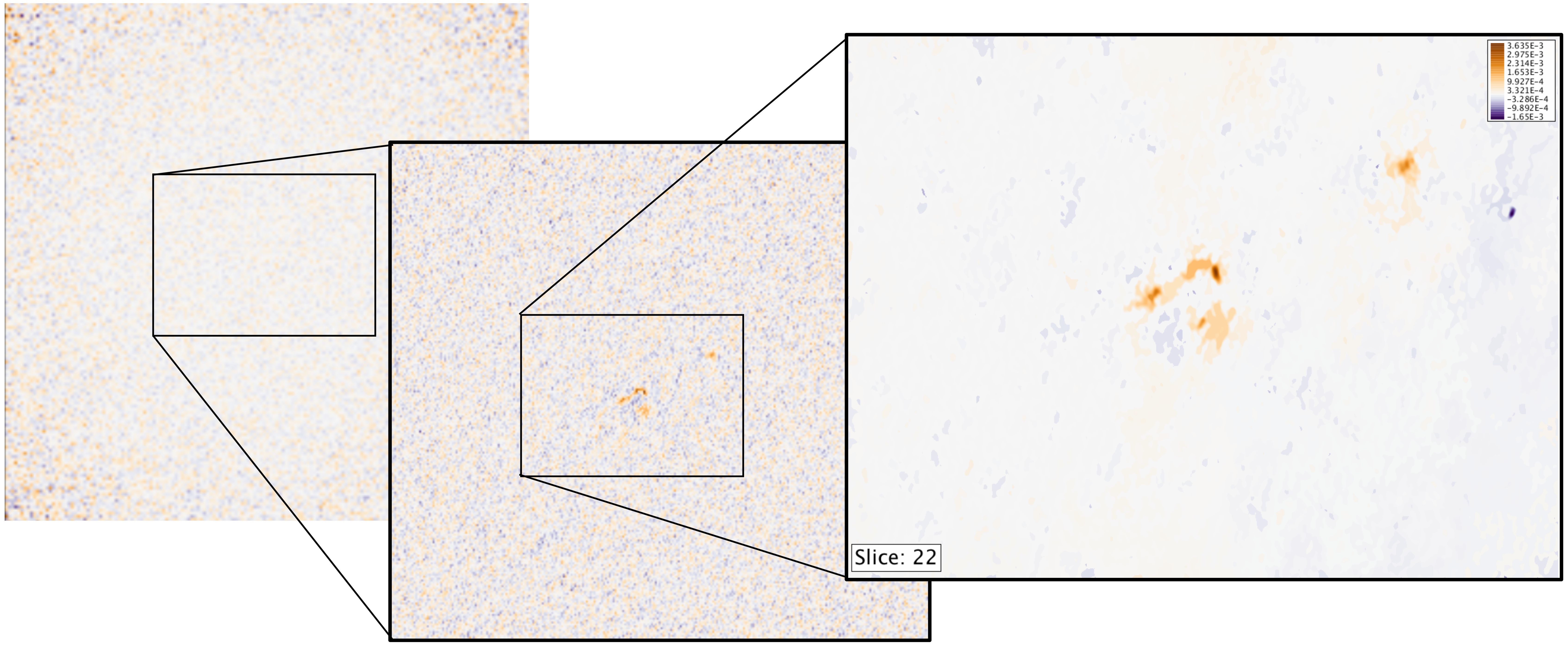
NGC 404: Mirach's Ghost Galaxy

Feature Denoting and Source Finding

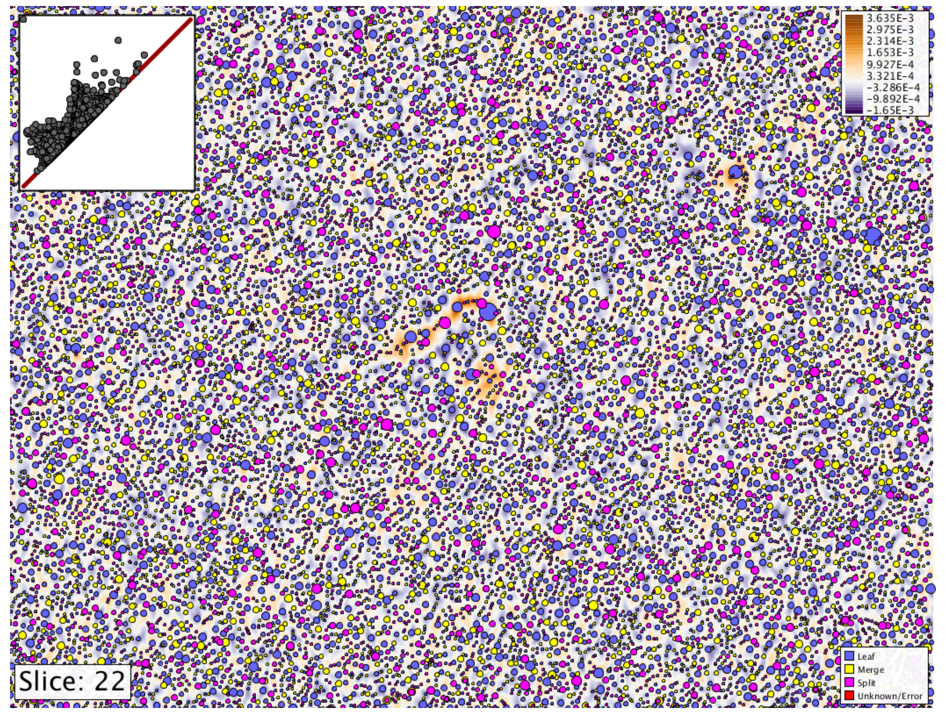


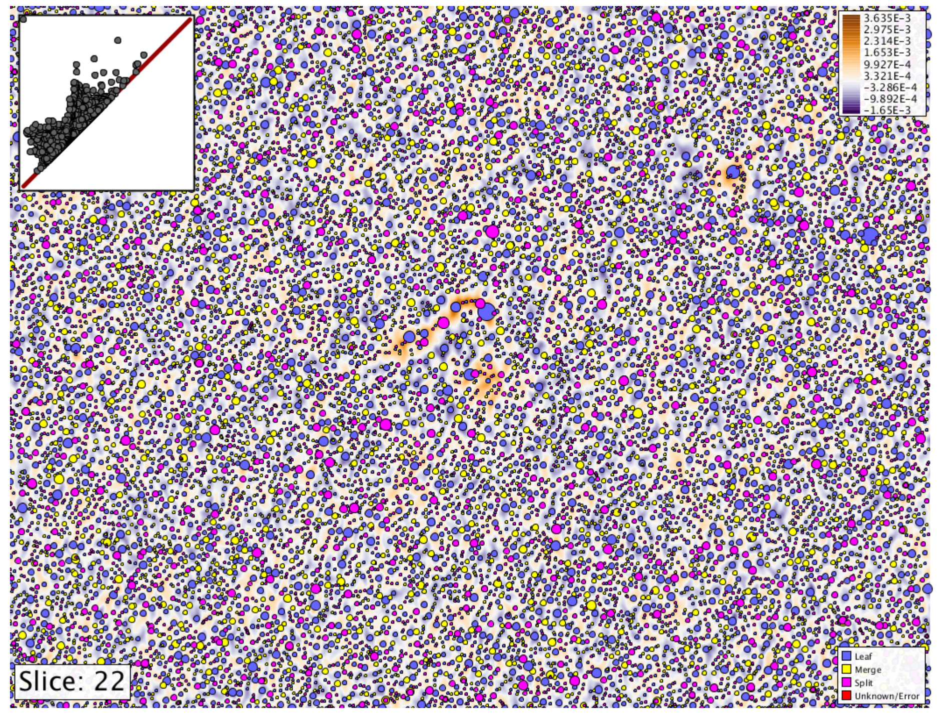


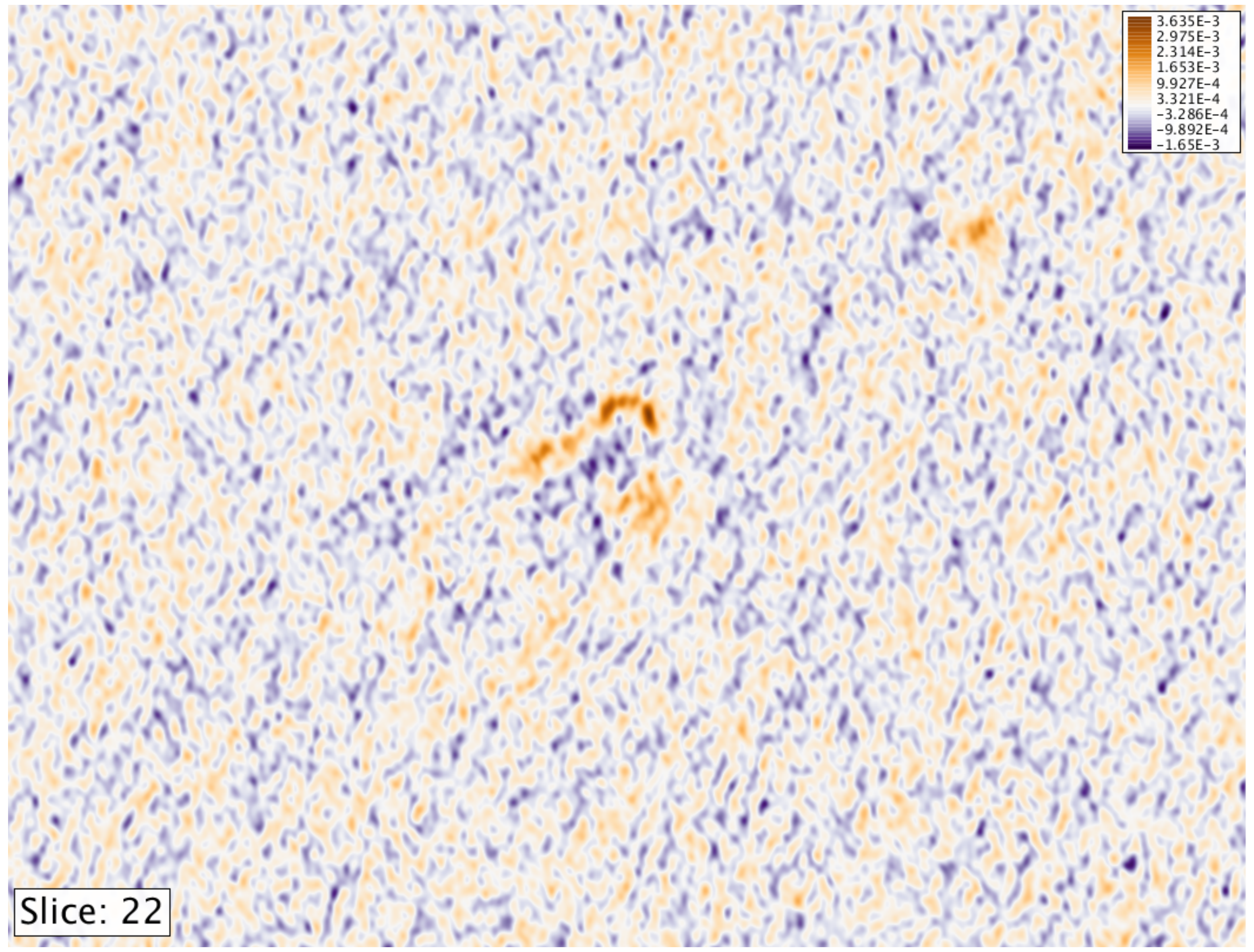
Paul Rosen, Bei Wang, Anil Seth, Betsy Mills, Adam Ginsburg, Julia Kamenetzky, Jeff Kern, Chris R. Johnson. Manuscript, 2017.

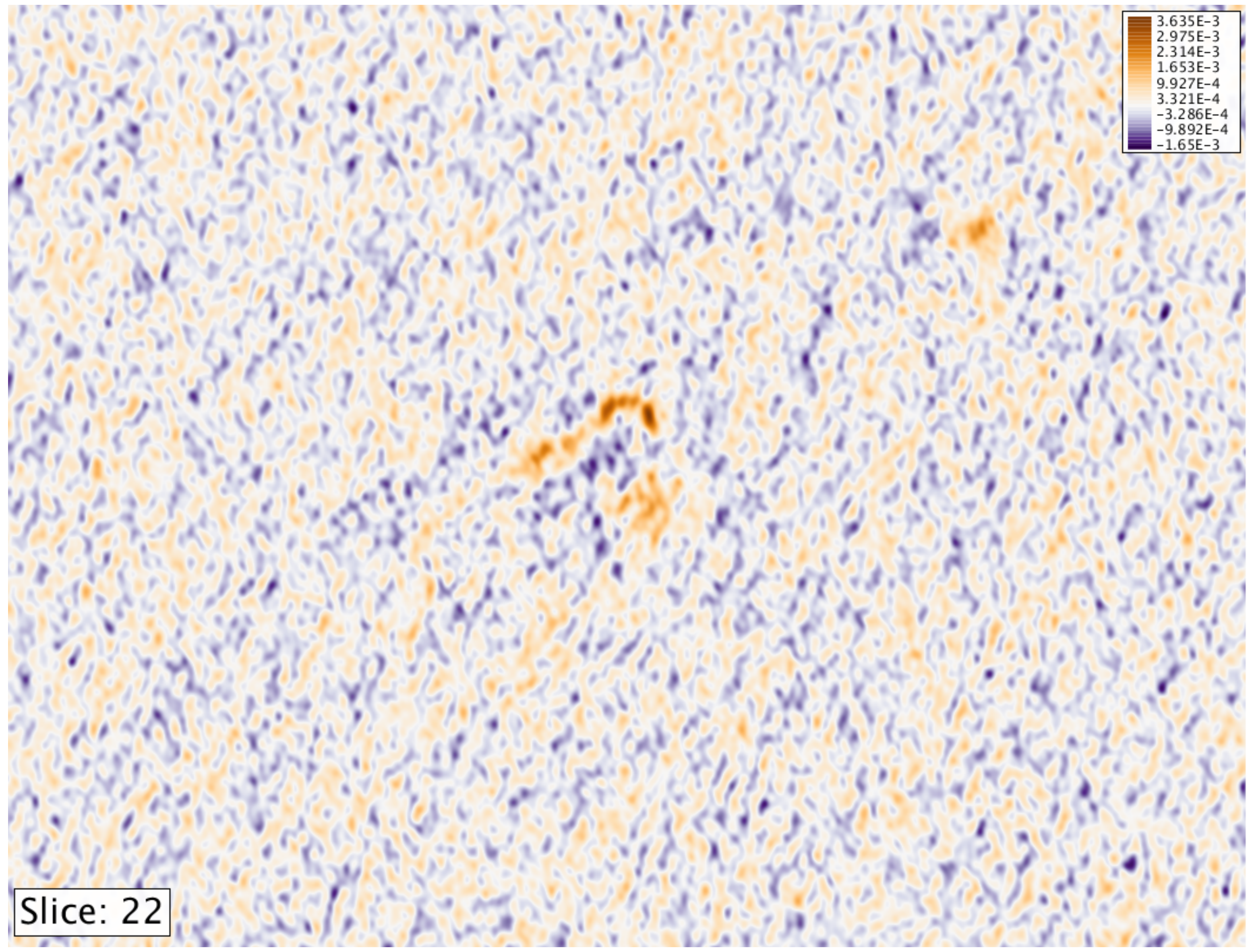

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22

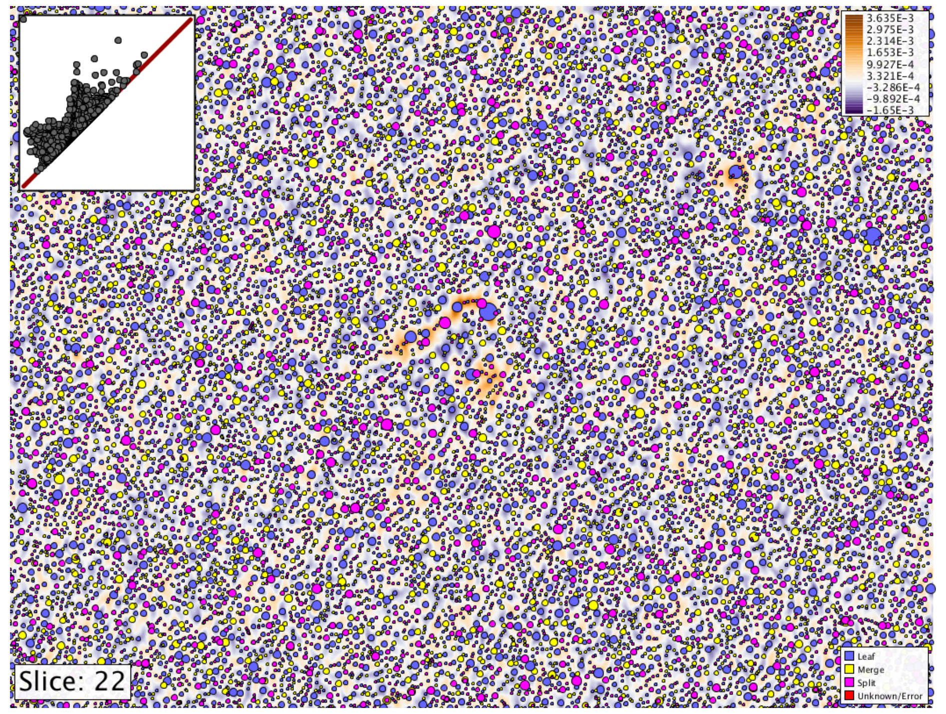
Denoising at Multi-scale and Source Finding

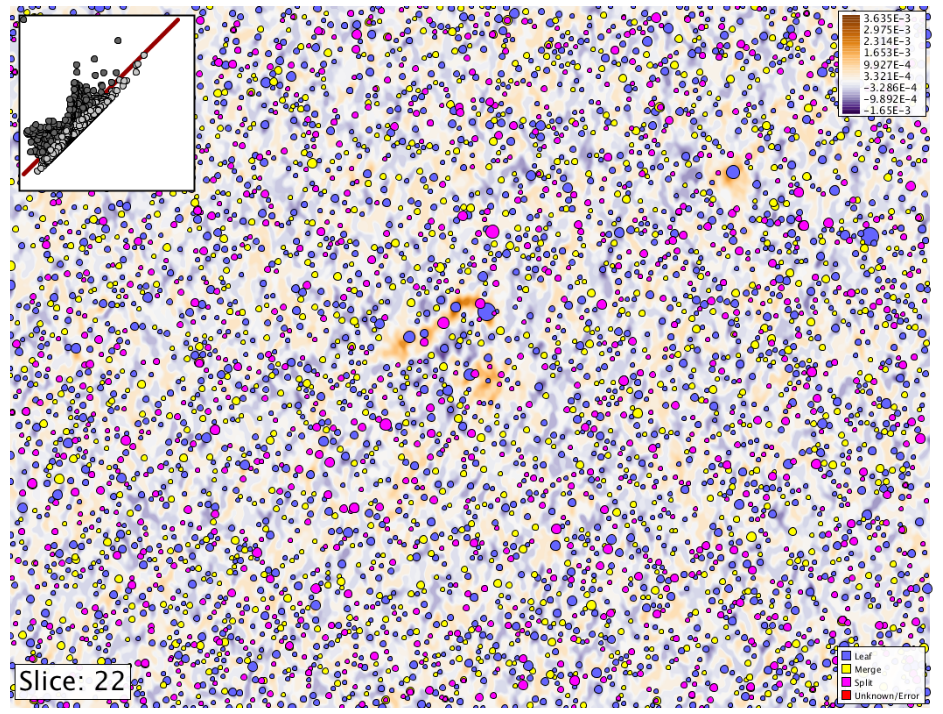

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
−3.286E-4
−9.892E-4
−1.65E-3
Slice: 22
Leaf
Merge
Split
Unknown/Error

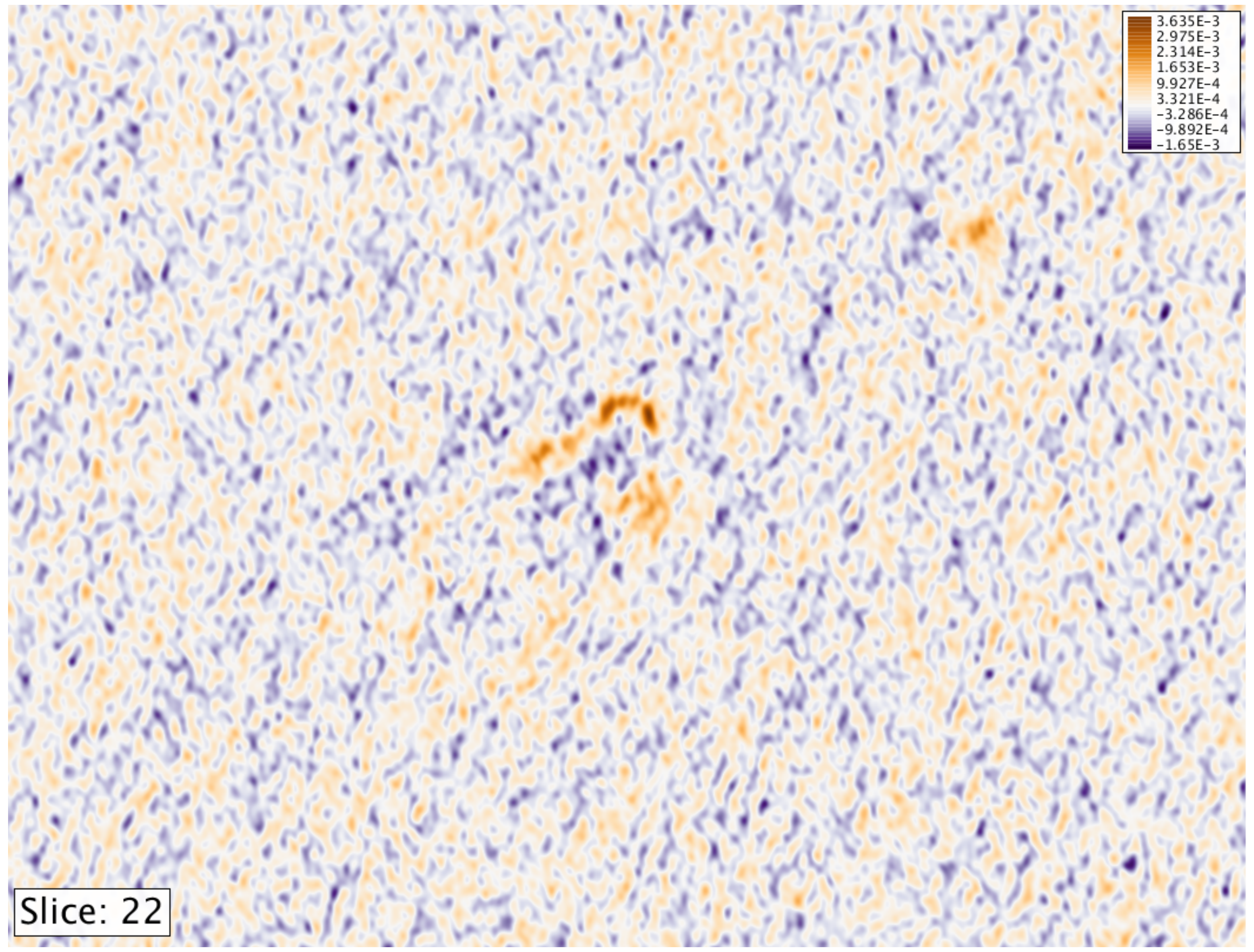

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
−3.286E-4
−9.892E-4
−1.65E-3
Slice: 22
Leaf
Merge
Split
Unknown/Error

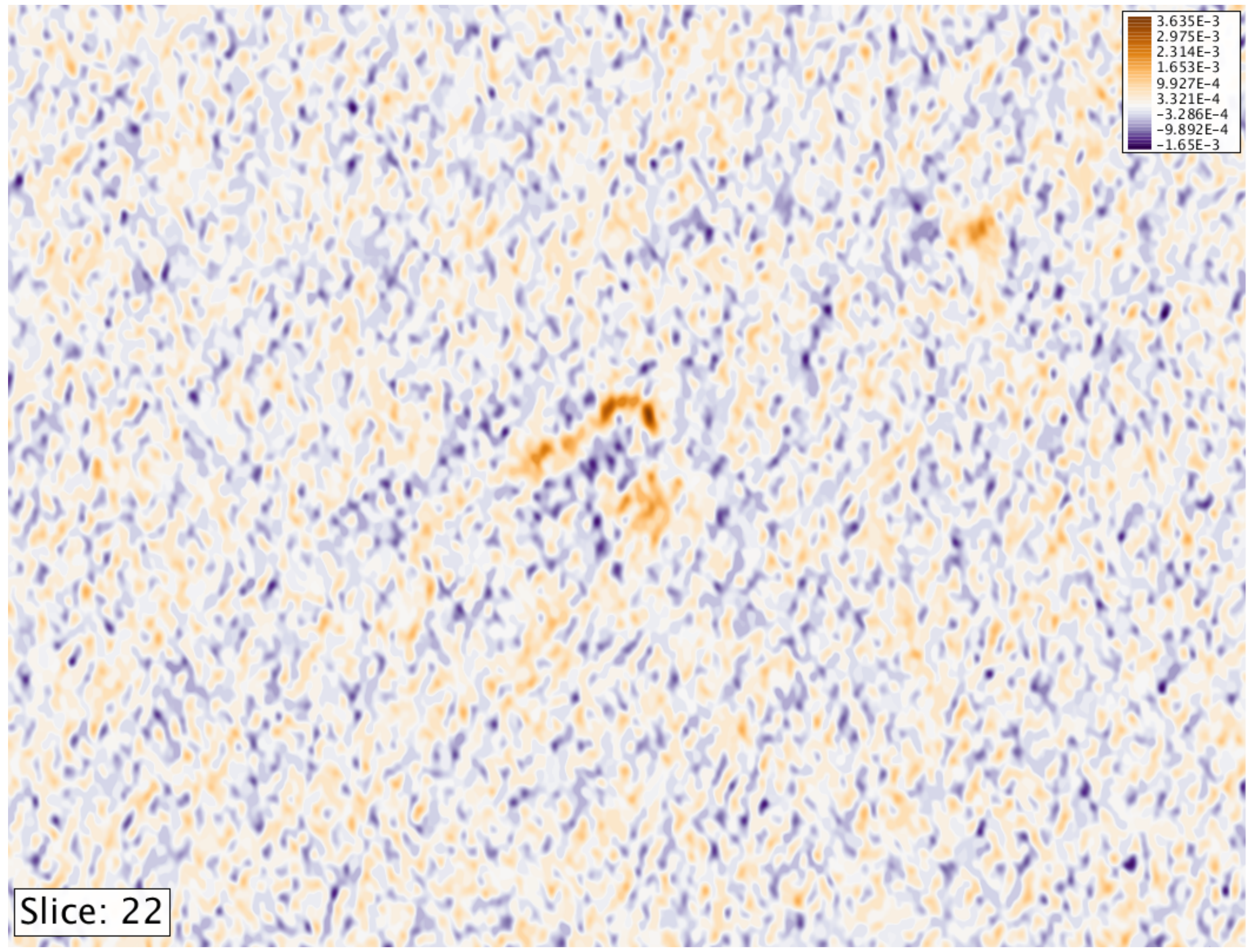

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
−3.286E-4
−9.892E-4
−1.65E-3
Slice: 22

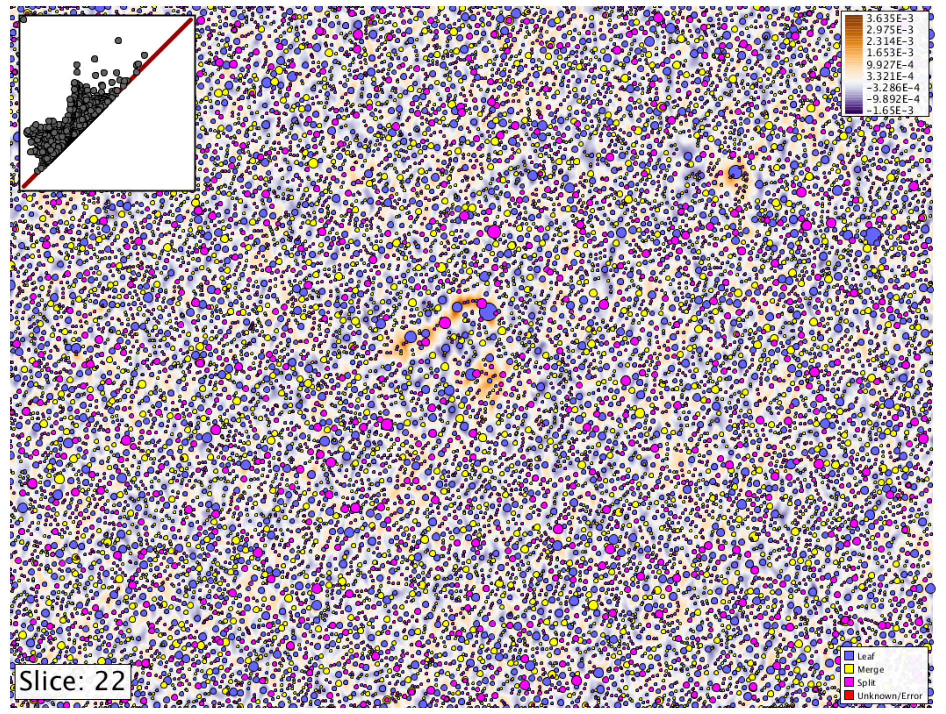

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
−3.286E-4
−9.892E-4
−1.65E-3
Slice: 22

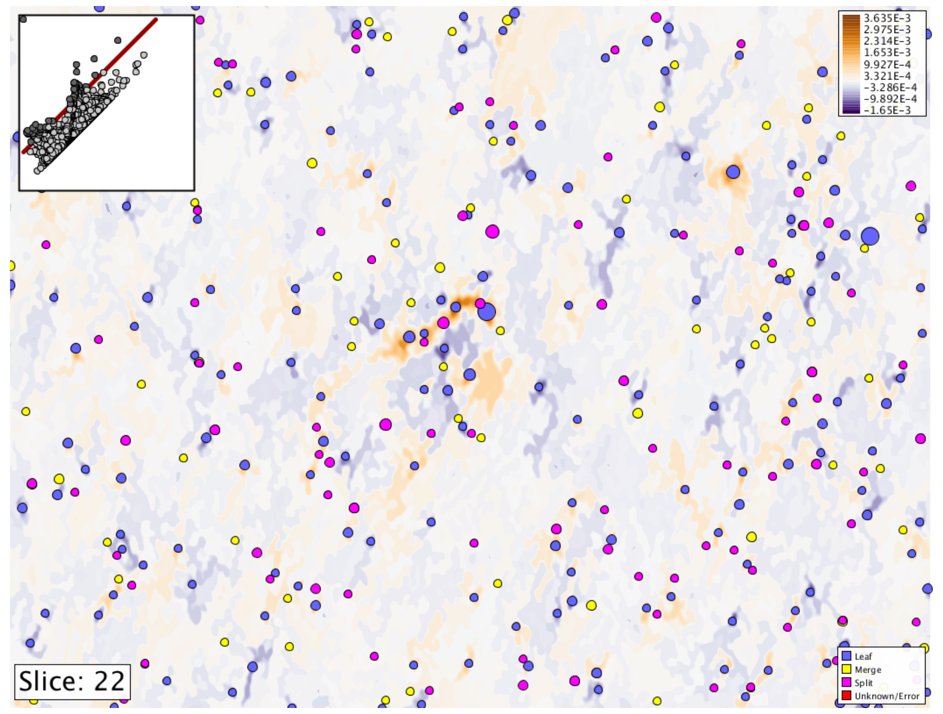

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Leaf
Merge
Split
Unknown/Error
Slice: 22

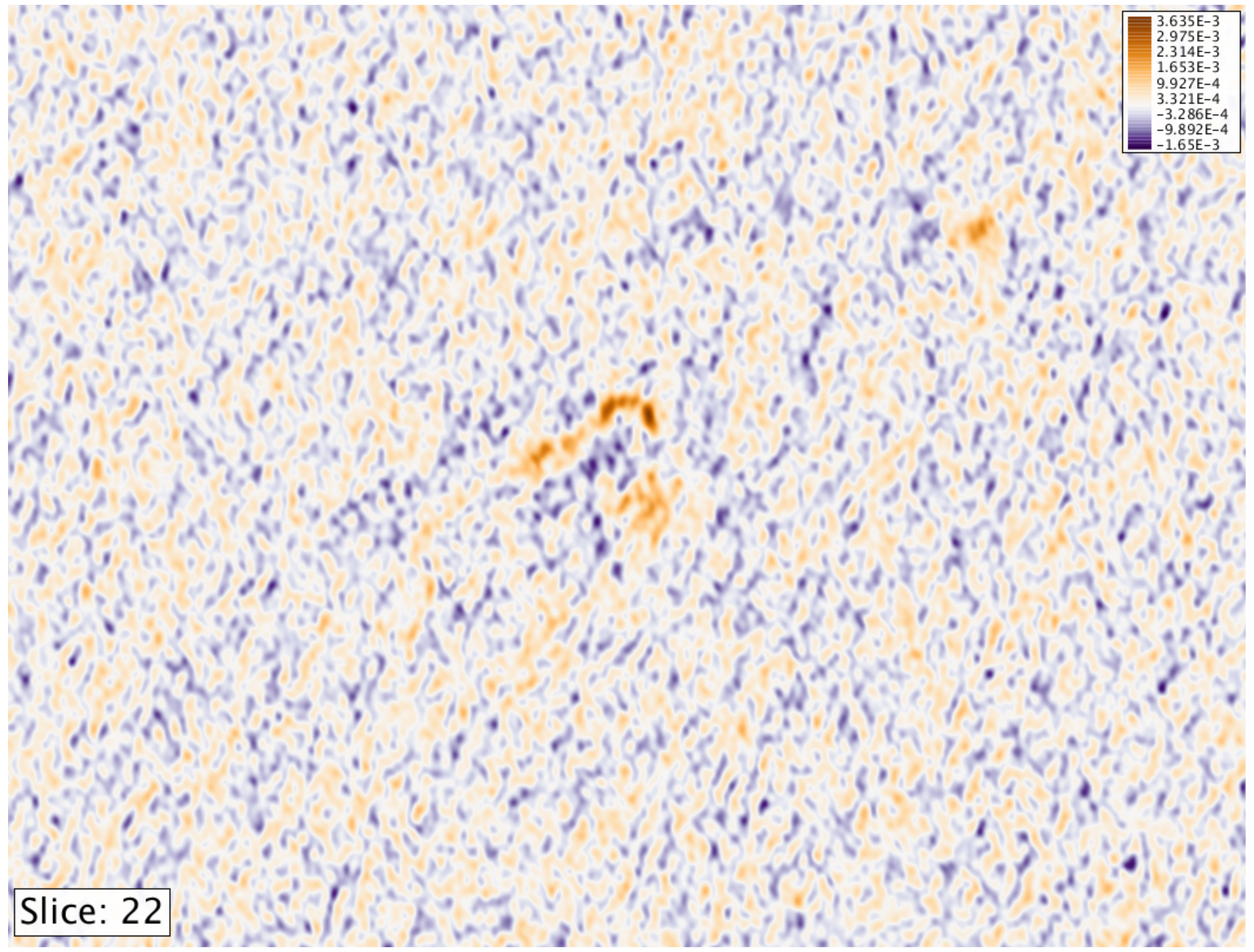

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Leaf
Merge
Split
Unknown/Error
Slice: 22

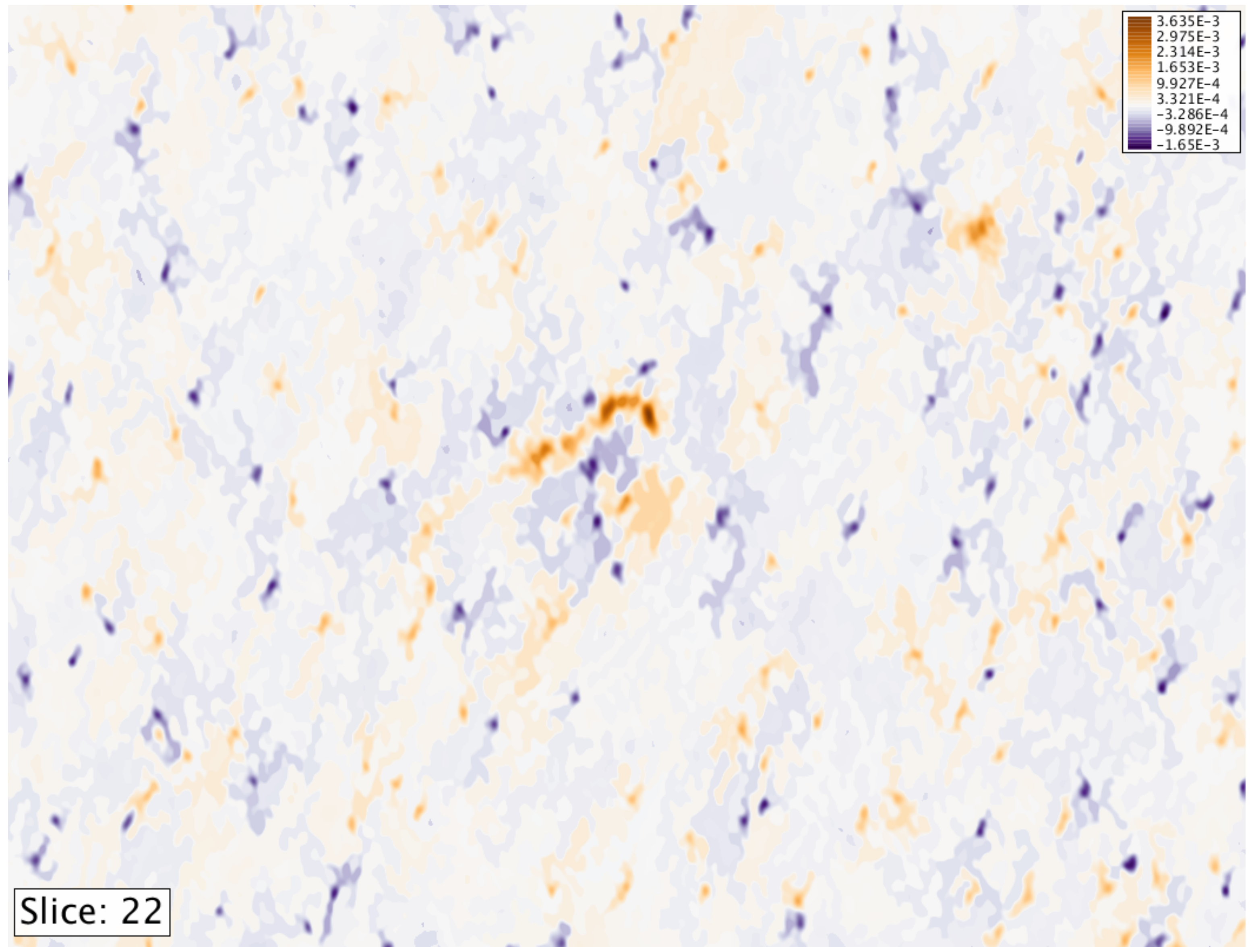

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22

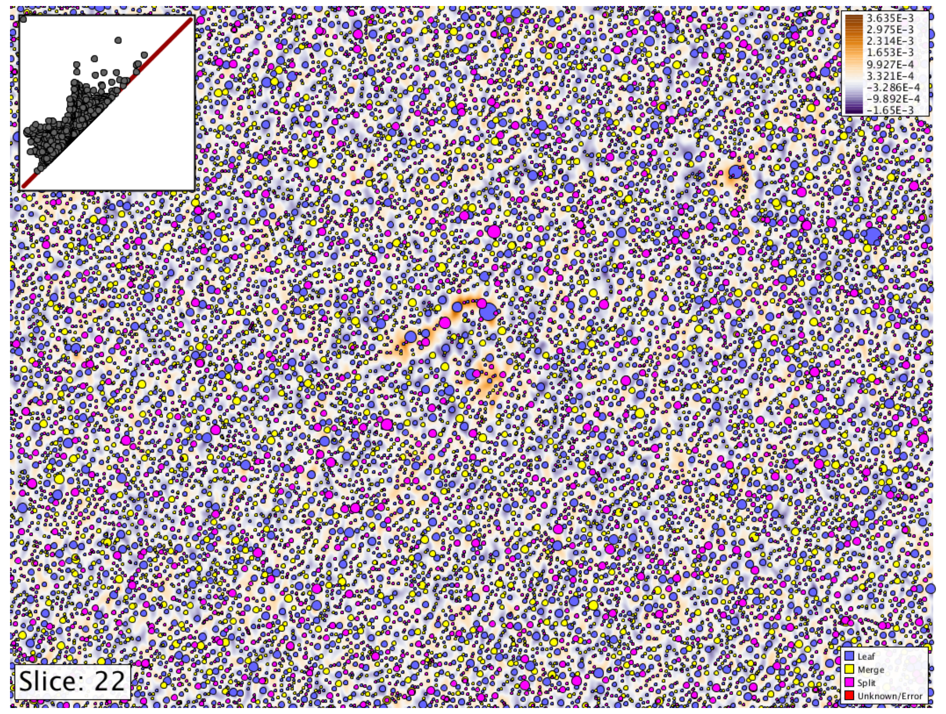

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22

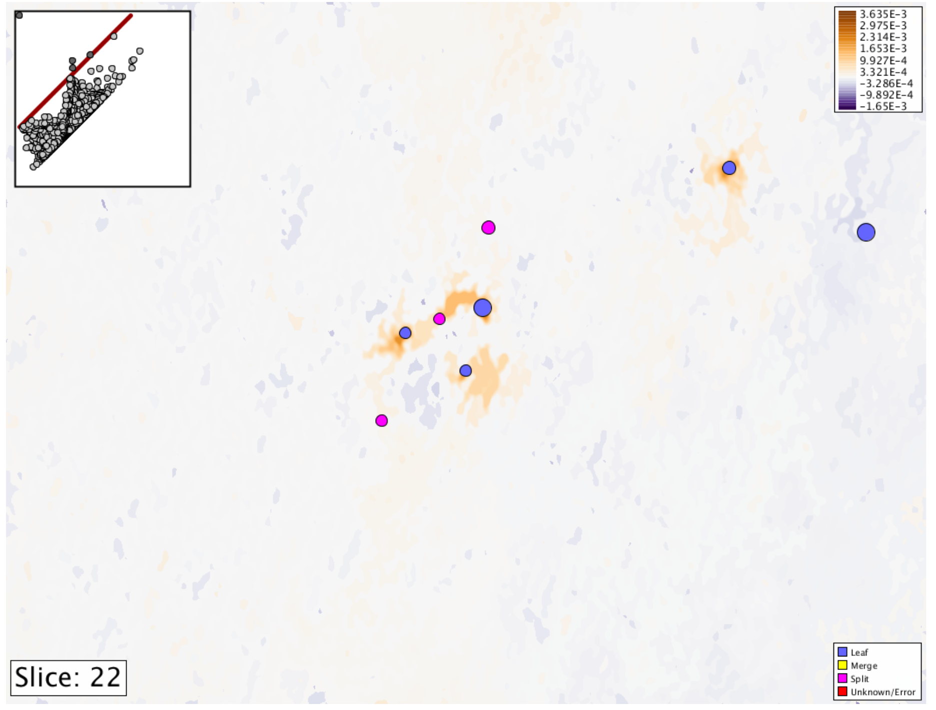

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Leaf
Merge
Split
Unknown/Error
Slice: 22

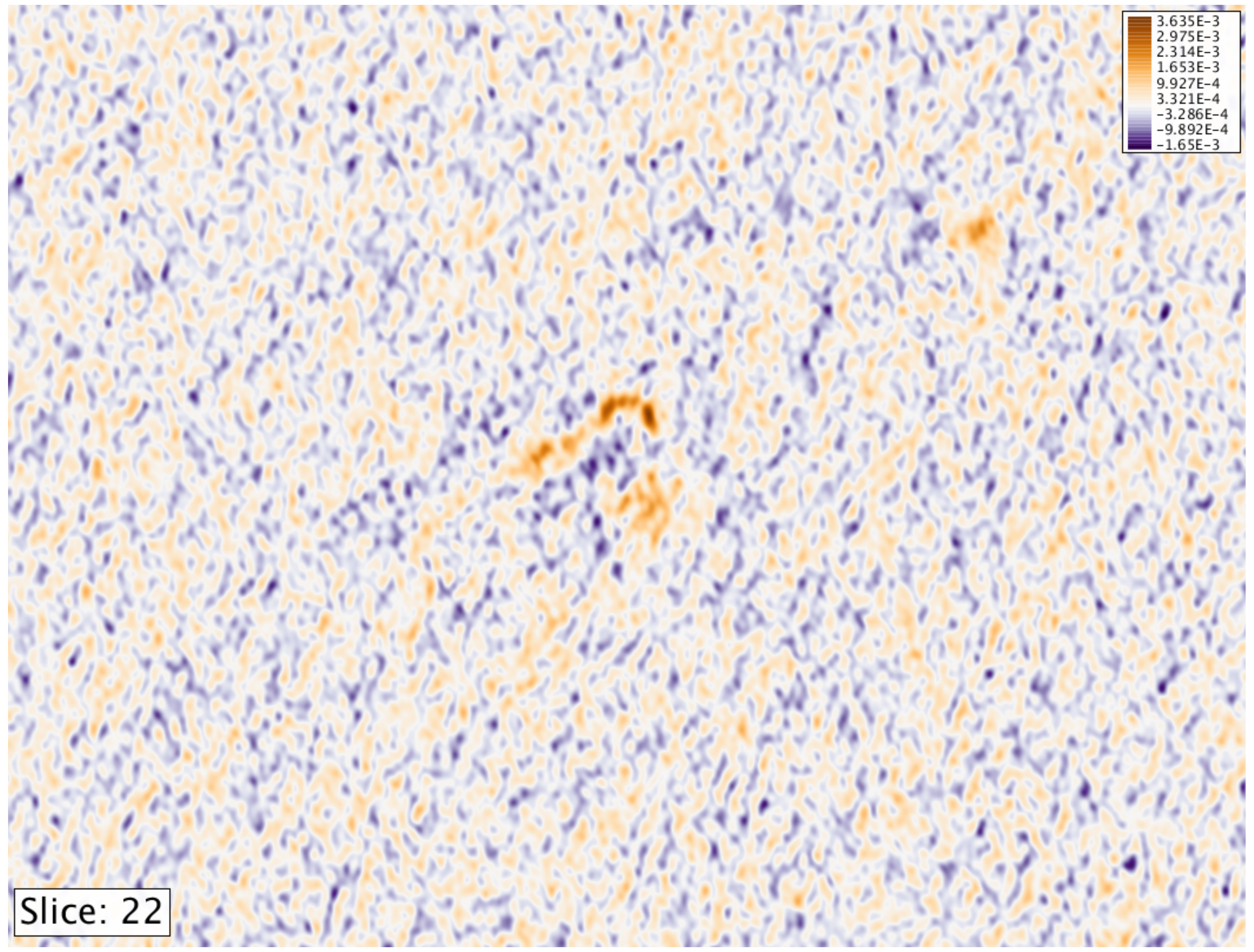

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Leaf
Merge
Split
Unknown/Error
Slice: 22

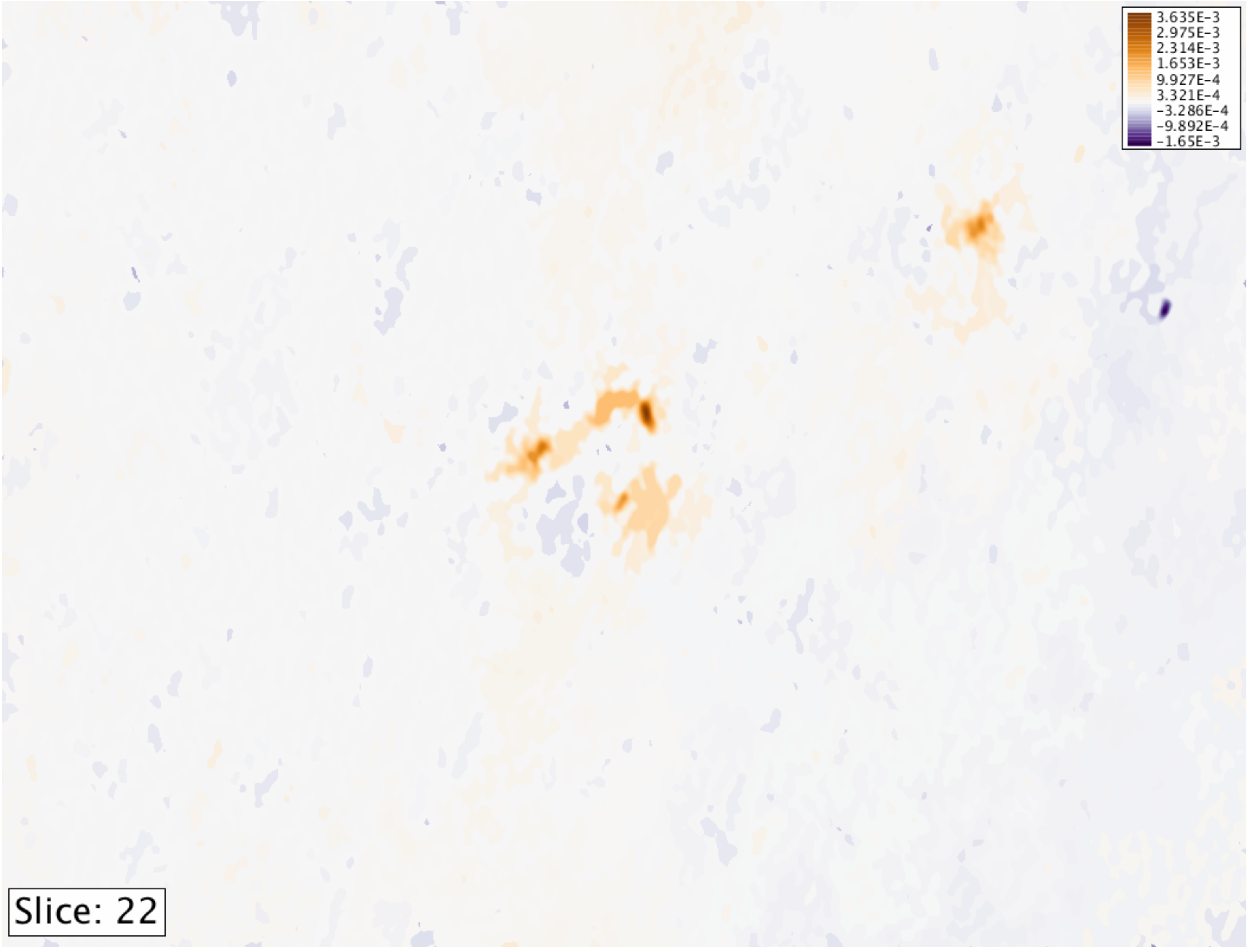

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22

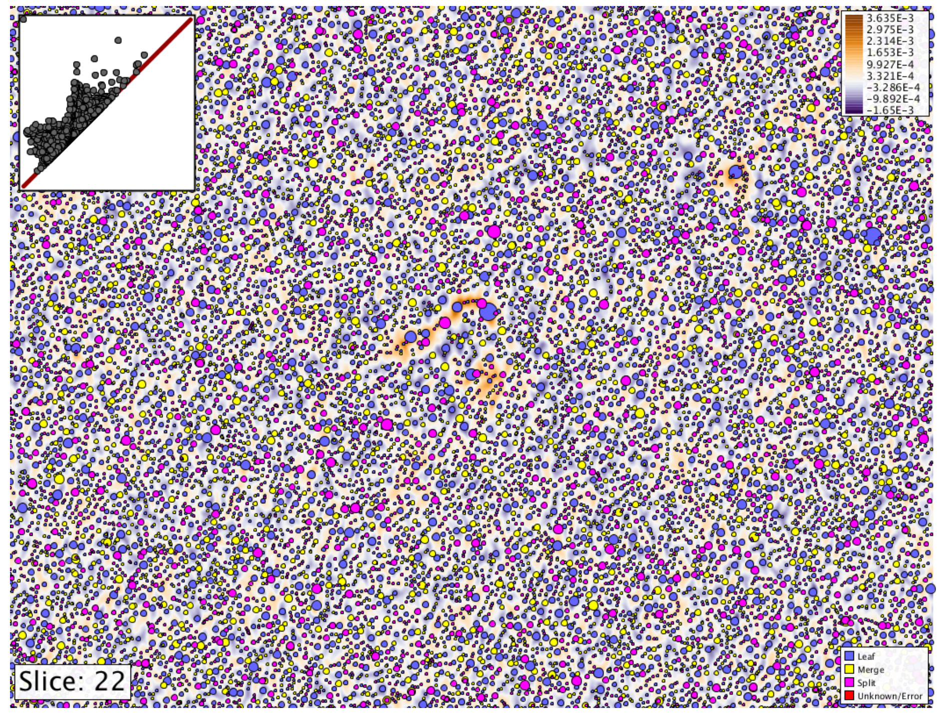

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22

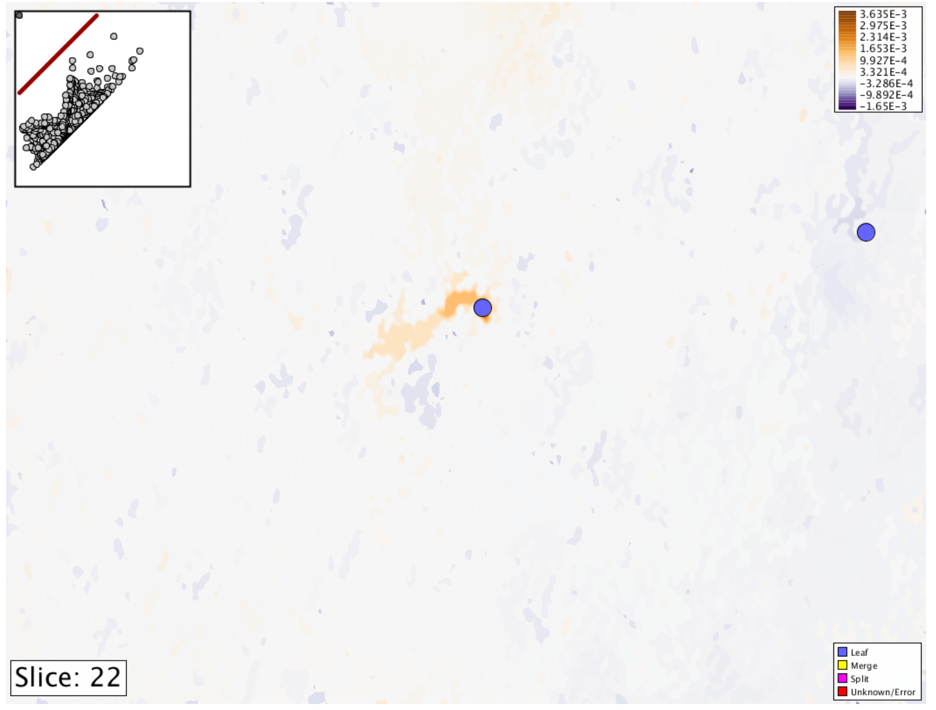

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Leaf
Merge
Split
Unknown/Error
Slice: 22

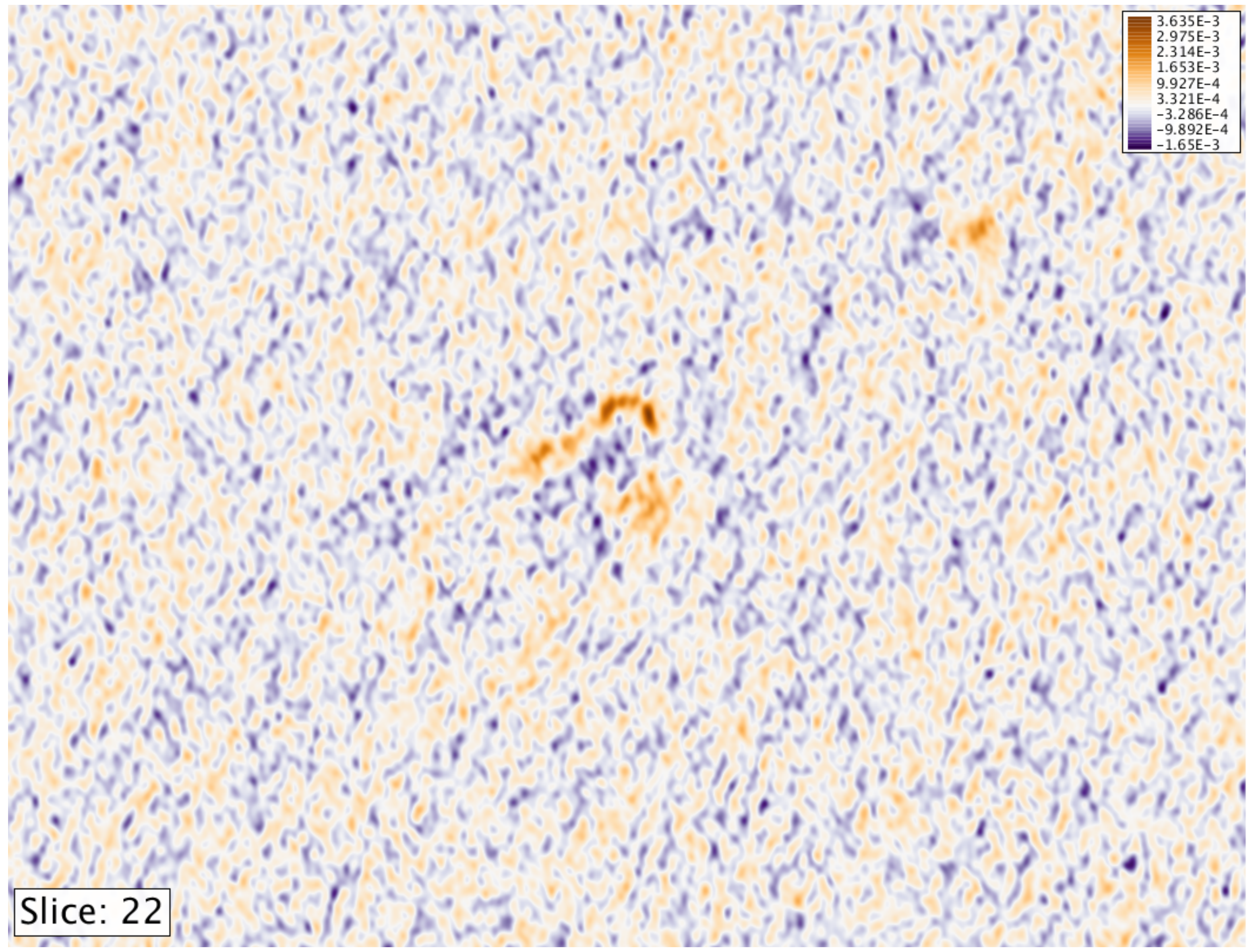

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Leaf
Merge
Split
Unknown/Error
Slice: 22

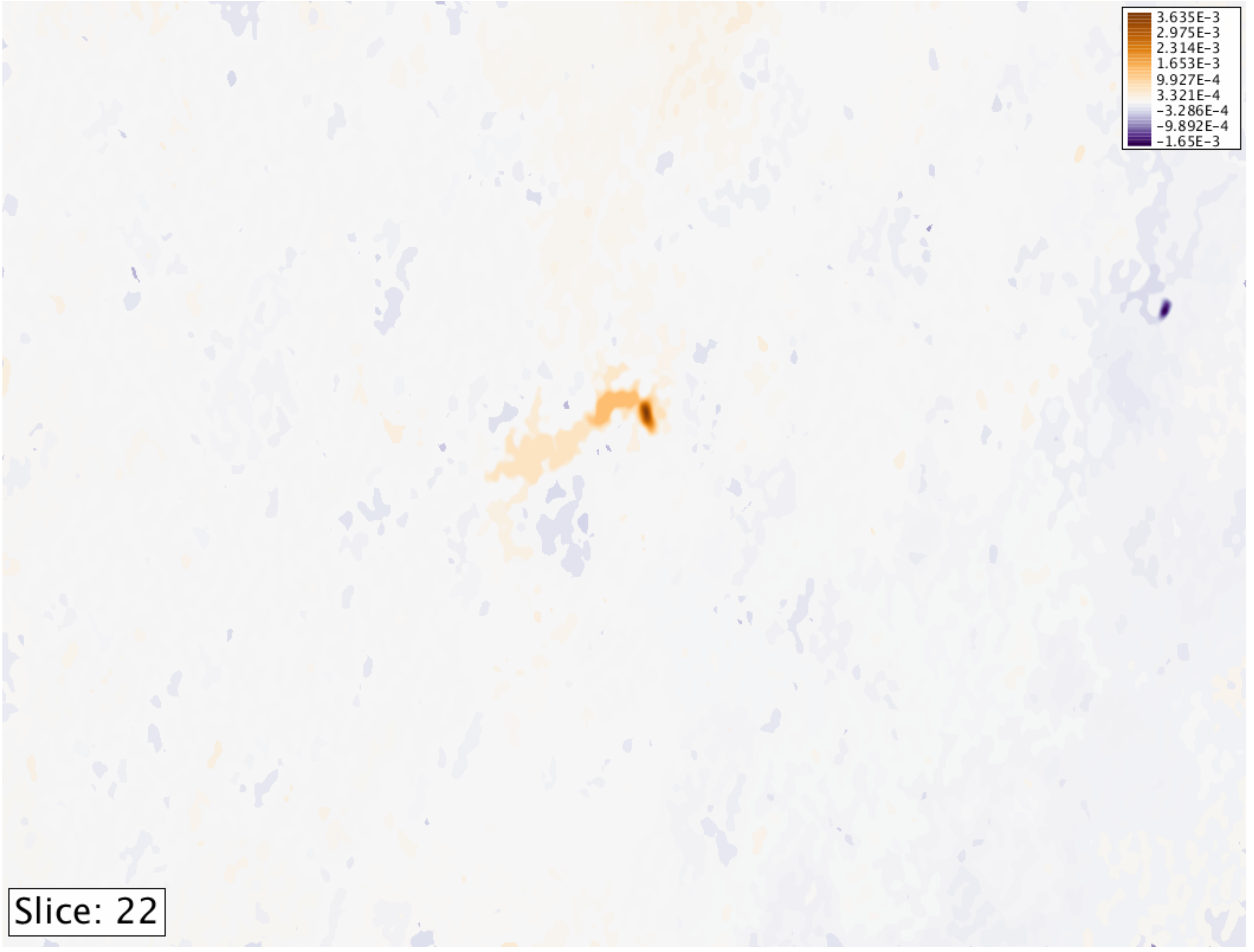

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22


3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22


3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Leaf
Merge
Split
Unknown/Error
Slice: 22

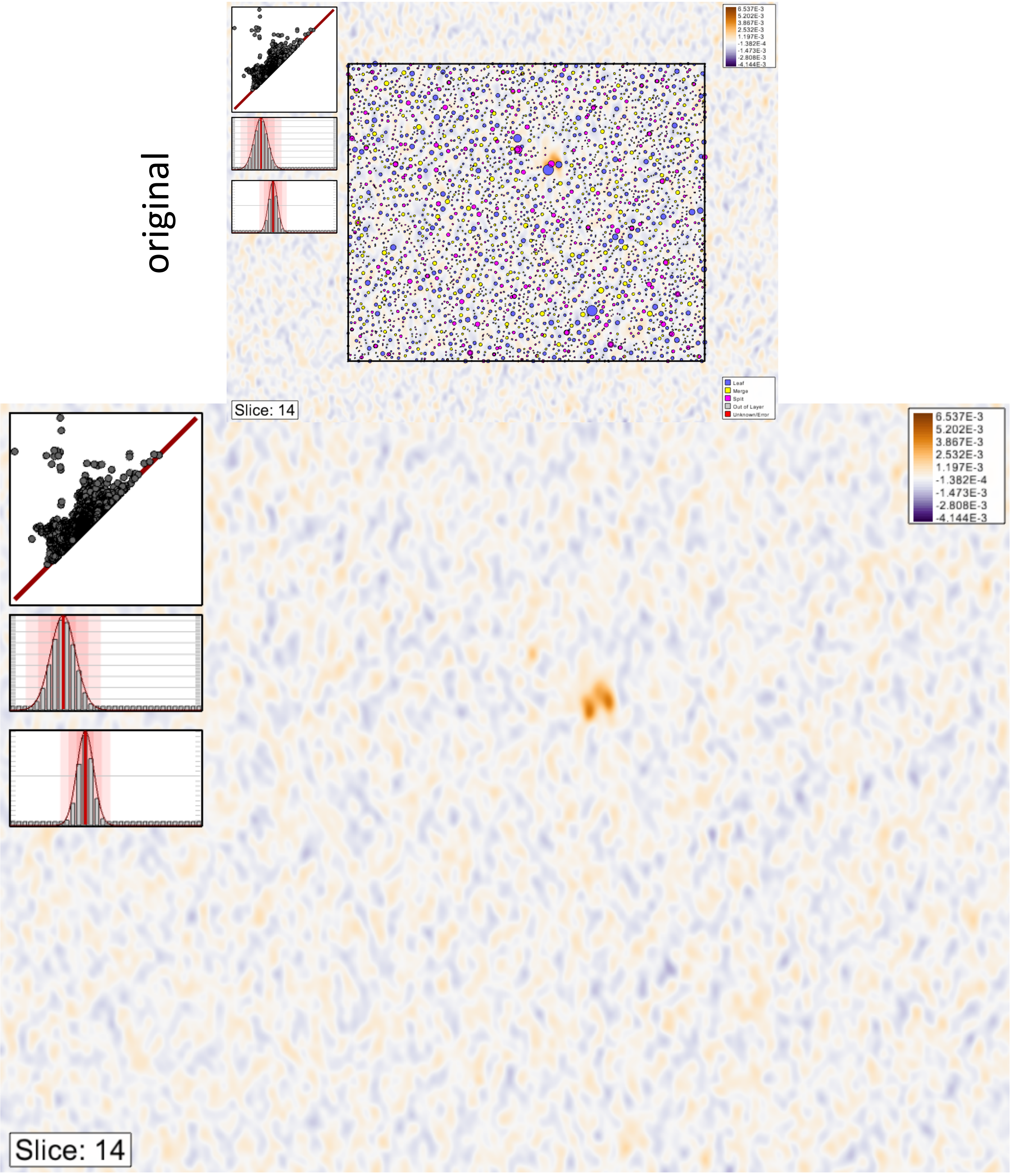

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Leaf
Merge
Split
Unknown/Error
Slice: 22


3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22

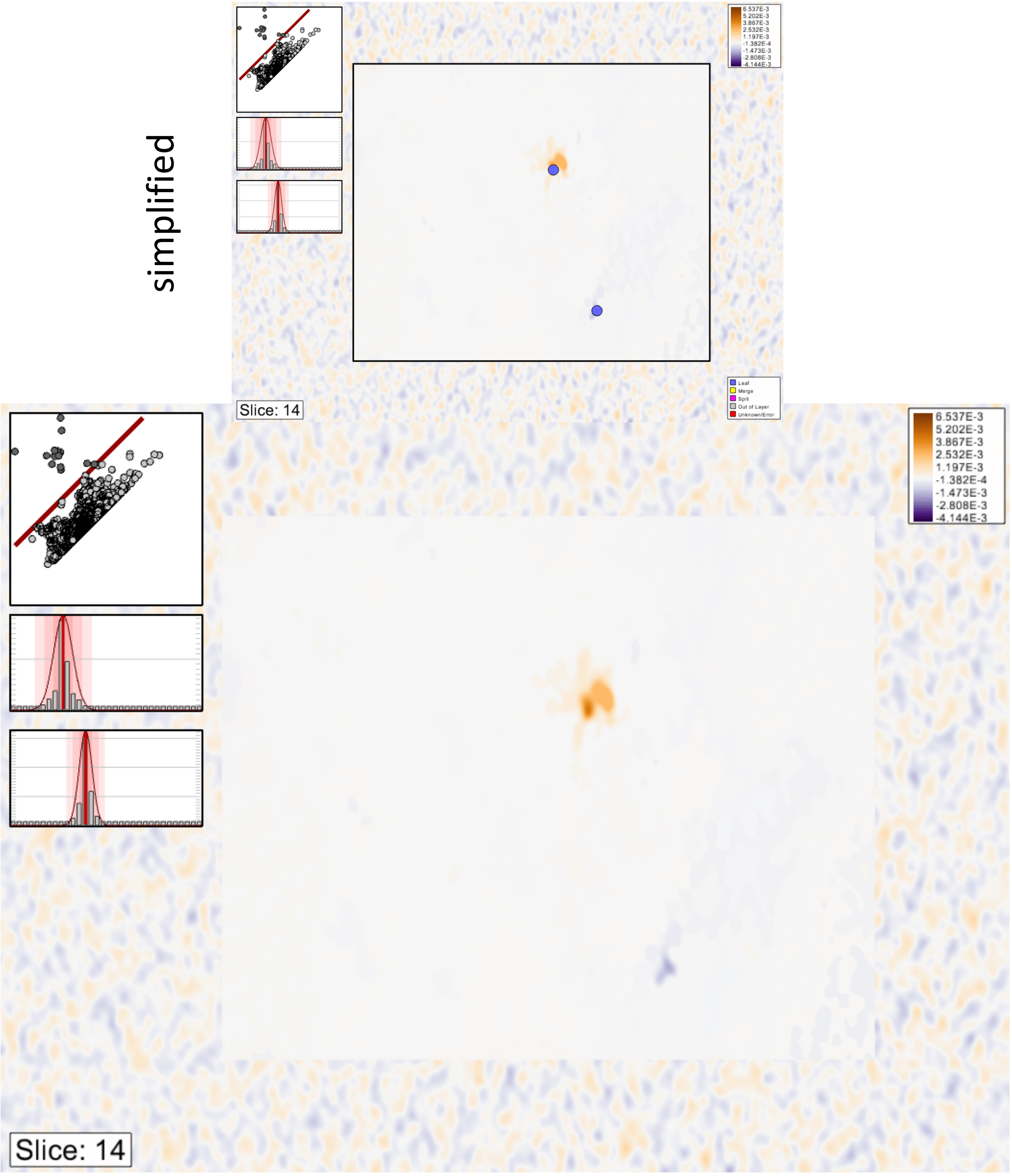

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22

Stepping Through Slices

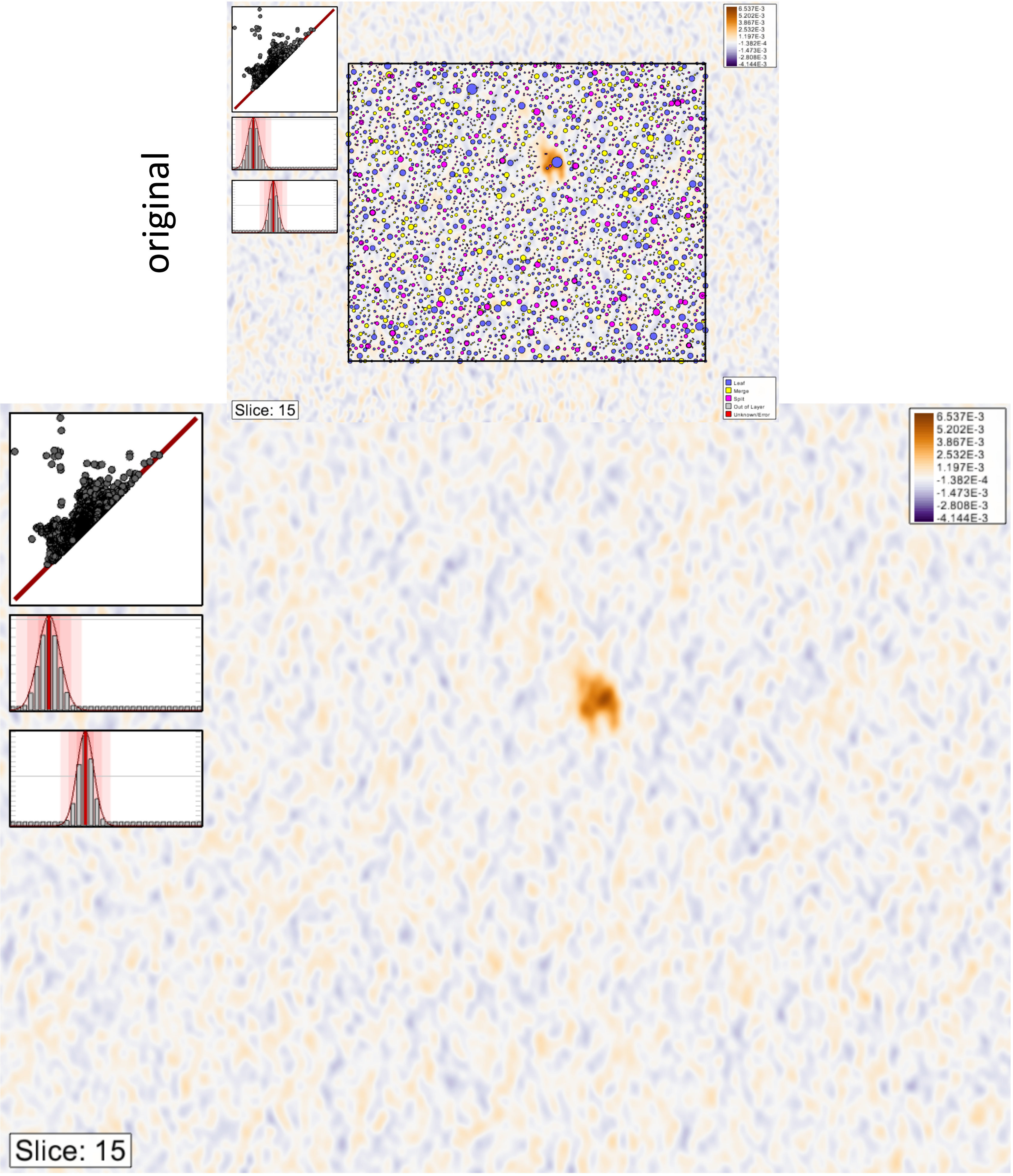
original



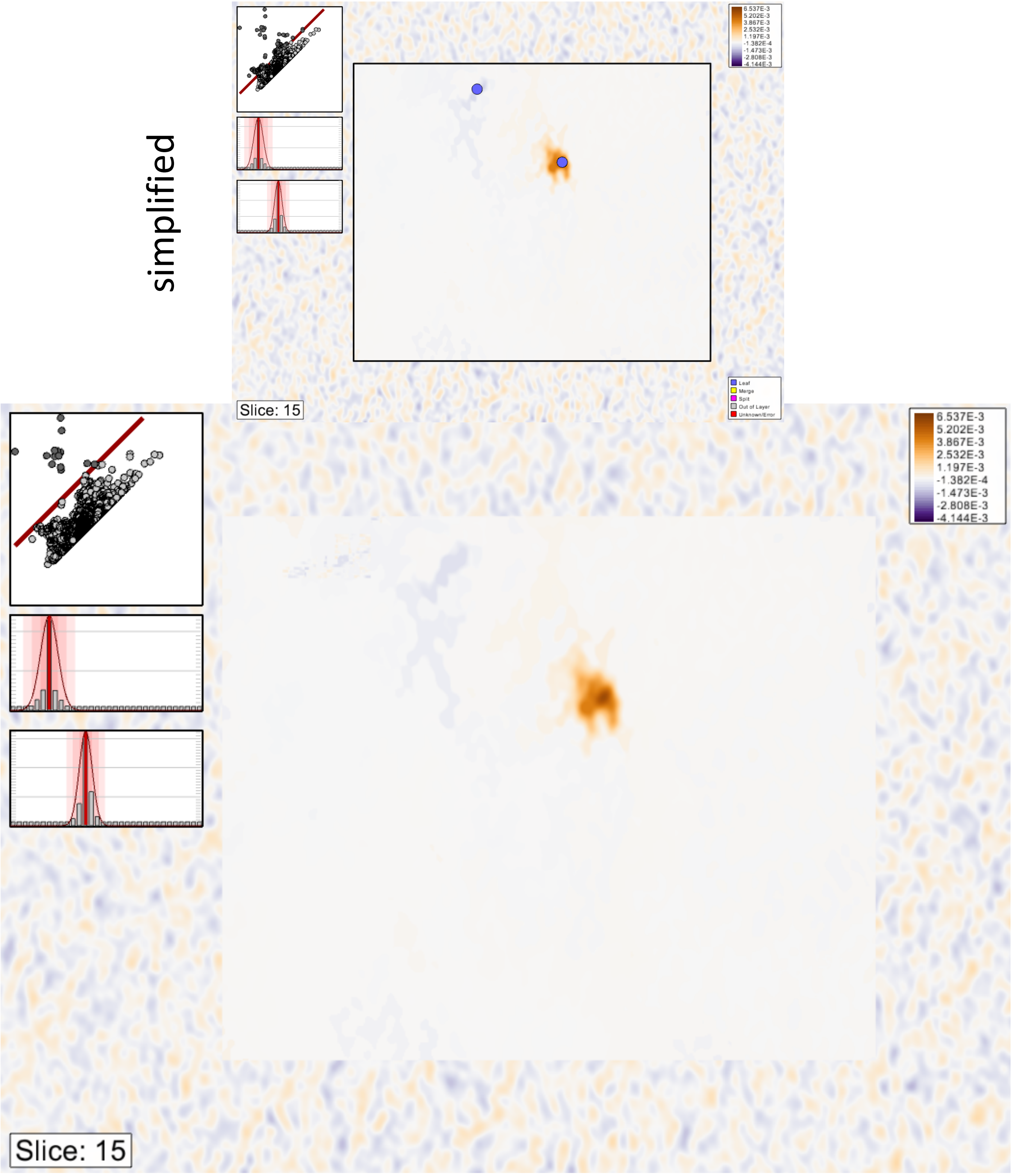
simplified



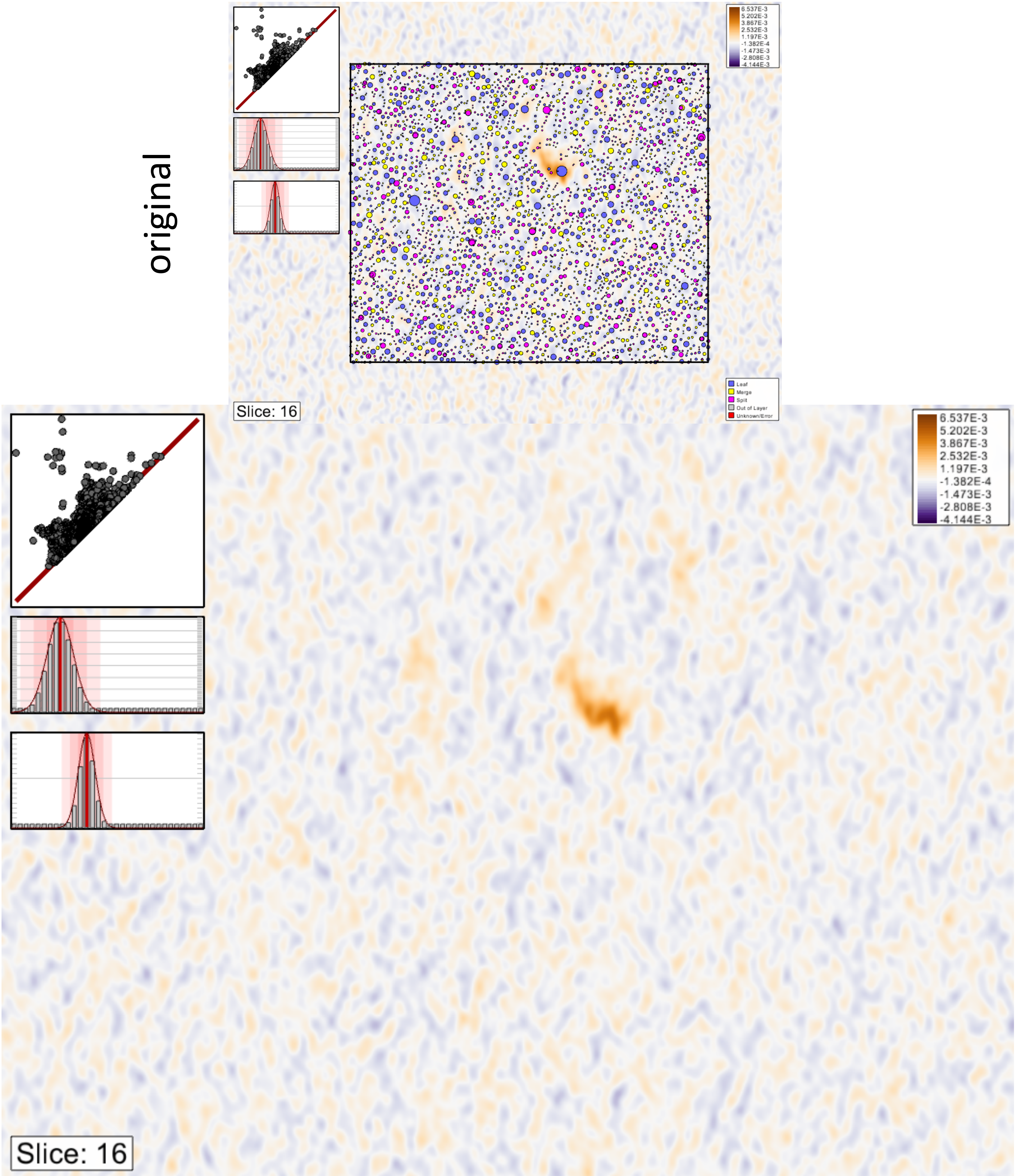
original



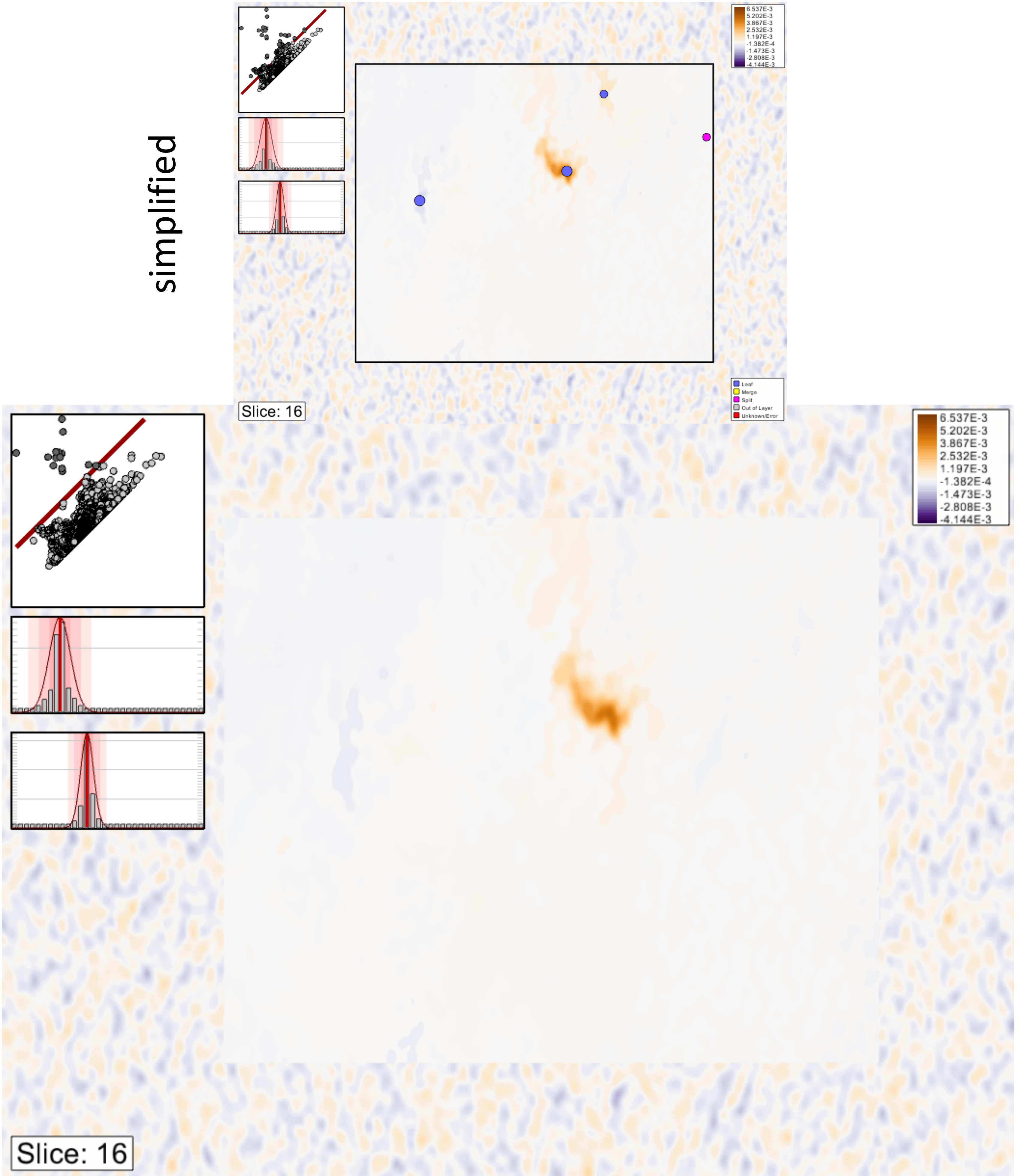
simplified



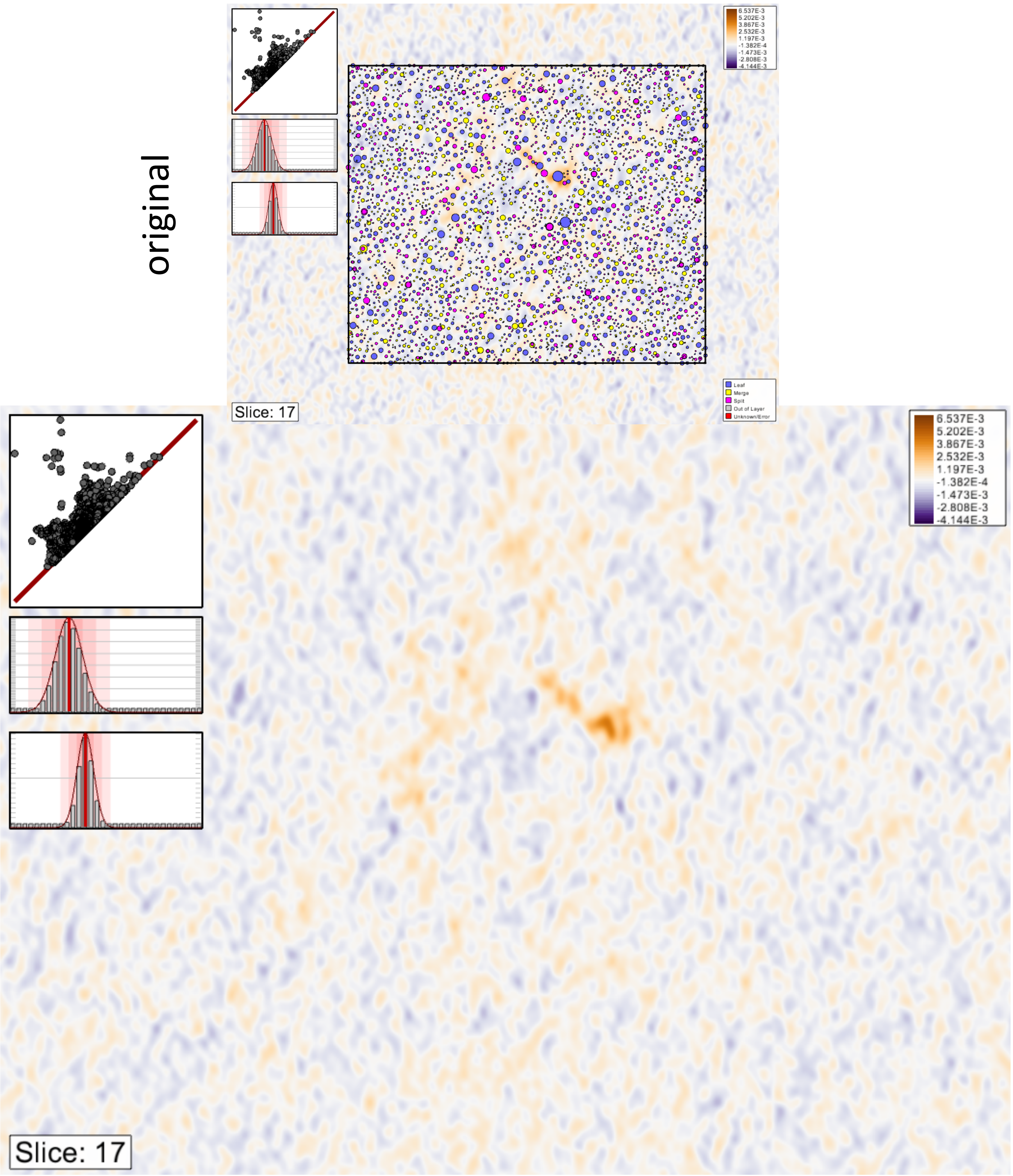
original



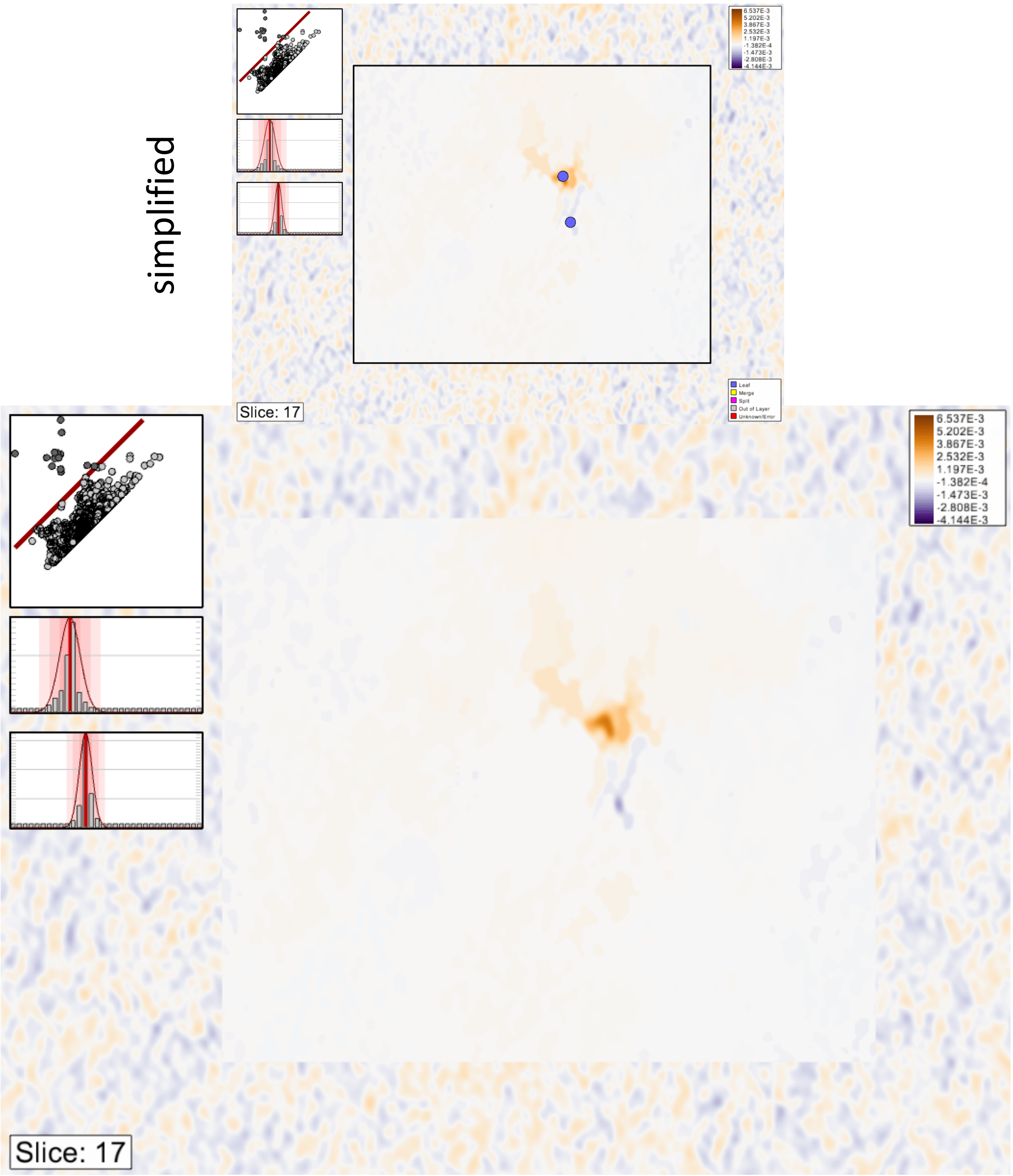
simplified



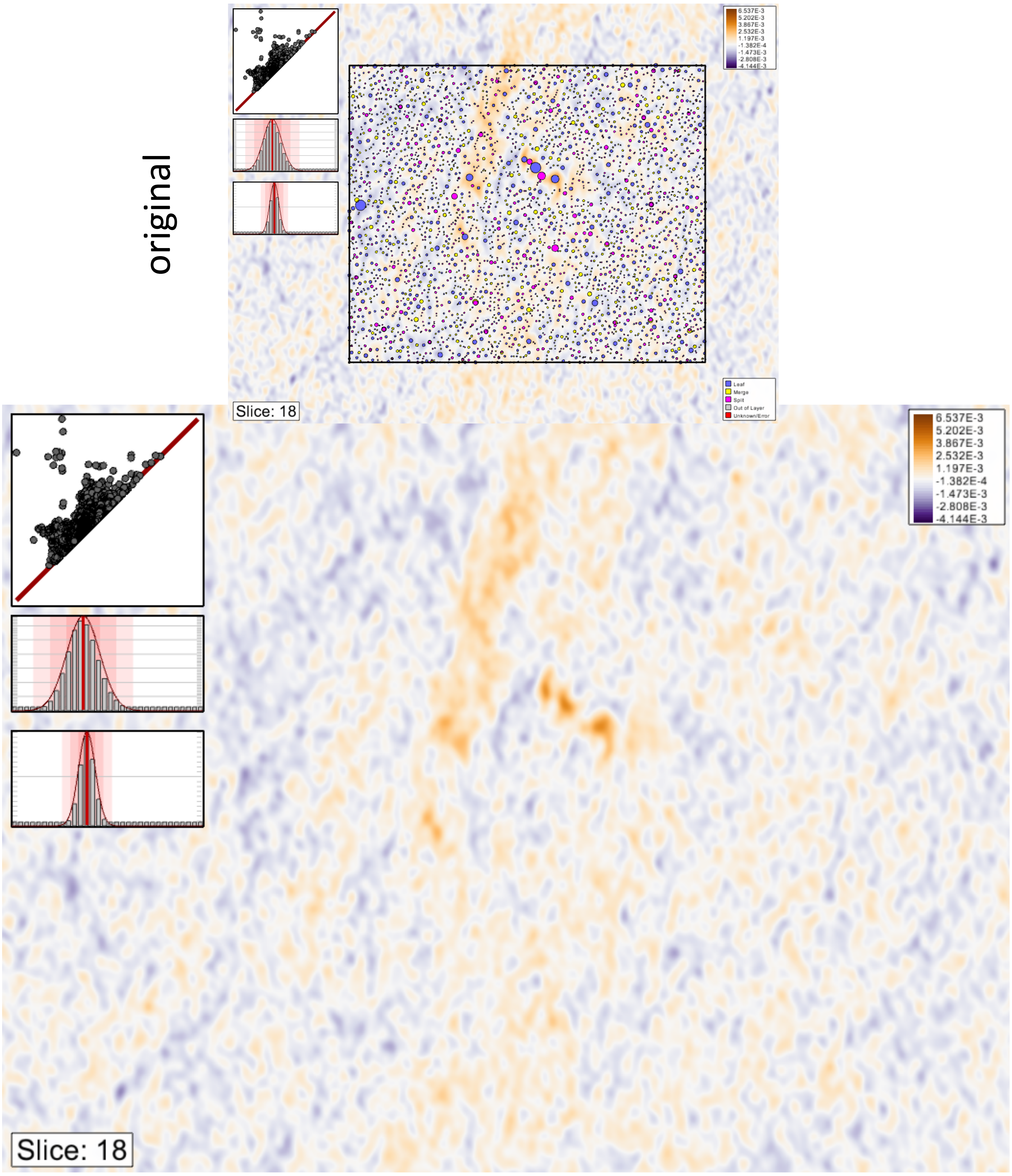
original



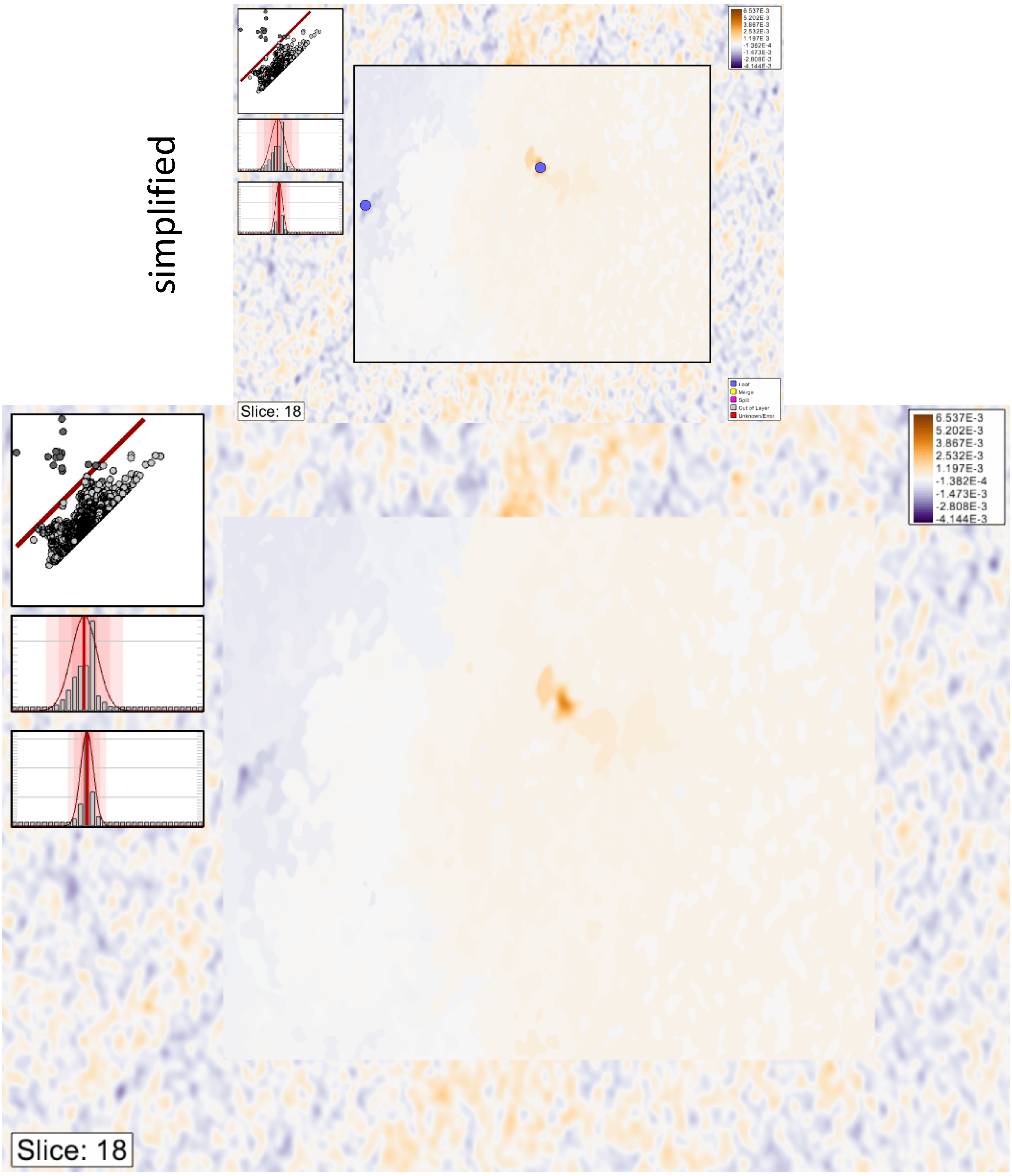
simplified



original

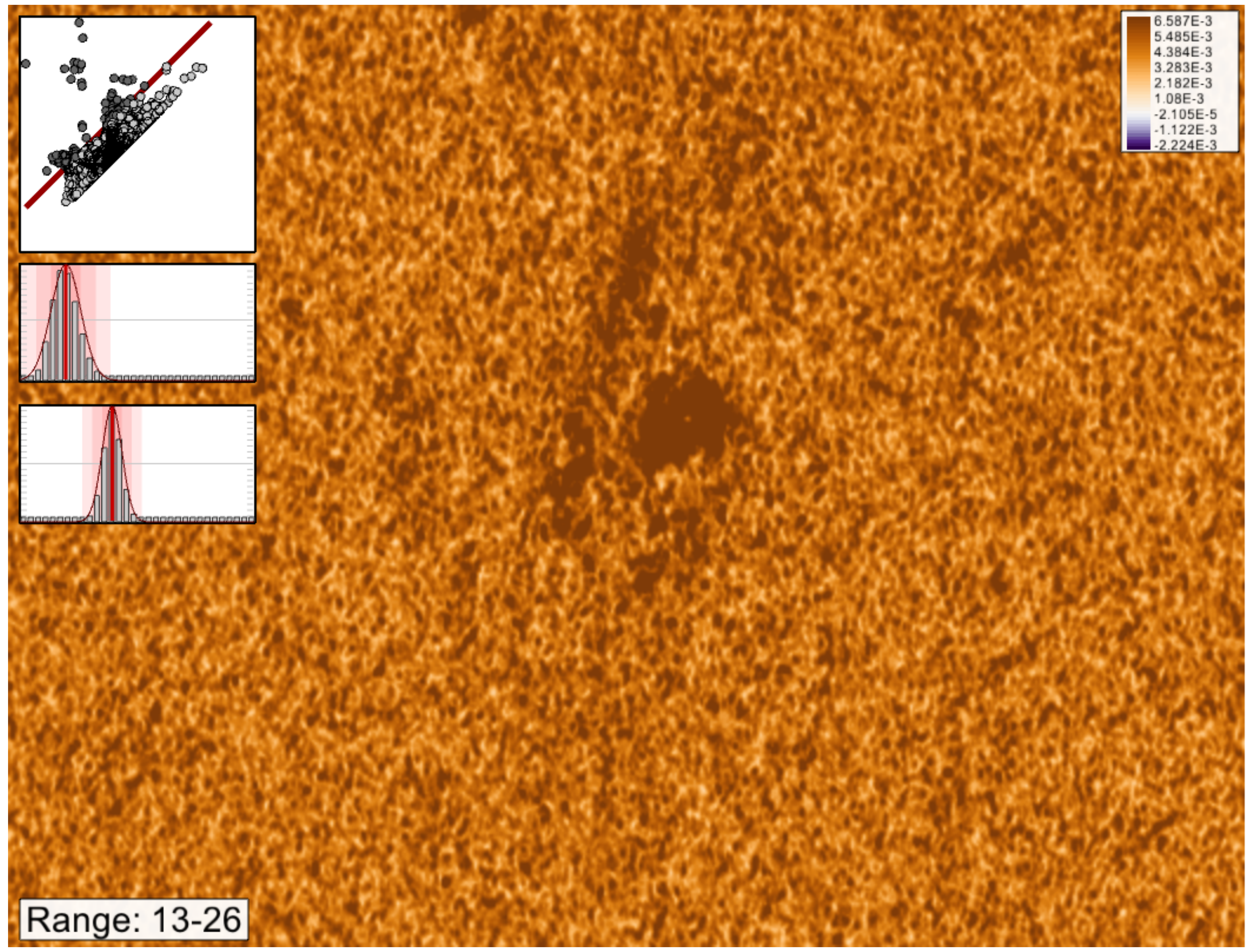


simplified

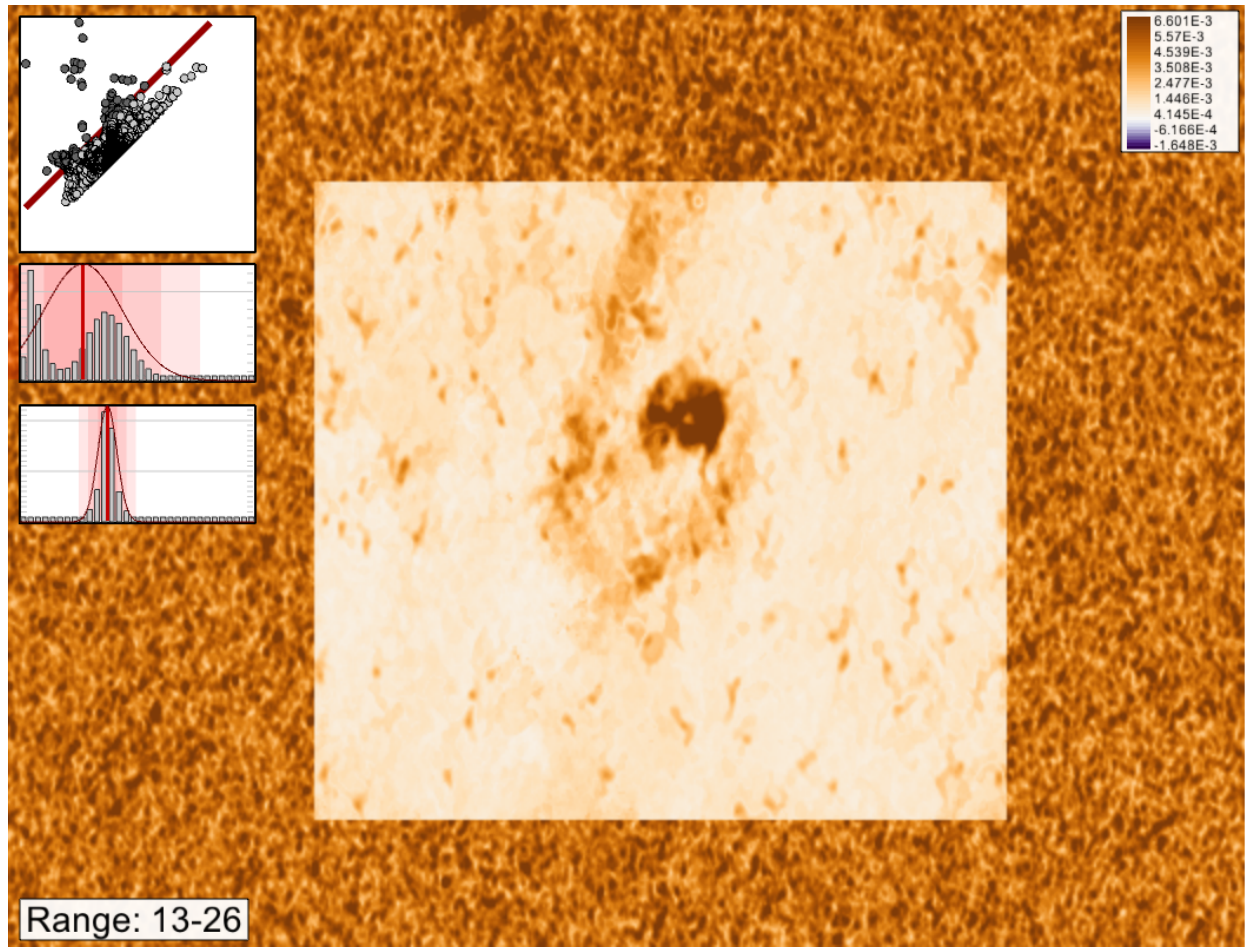


Moment 0 Analysis

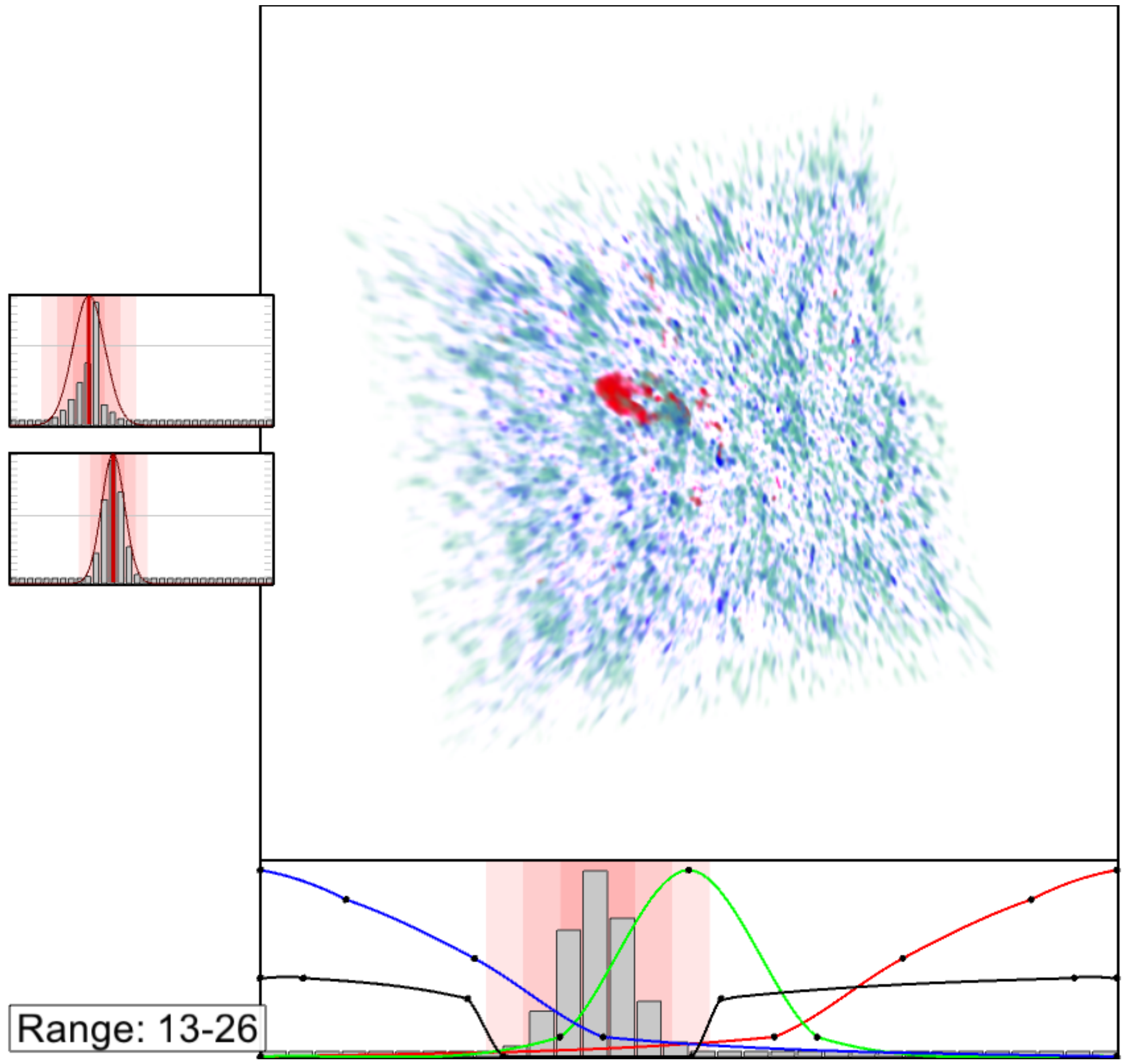
original

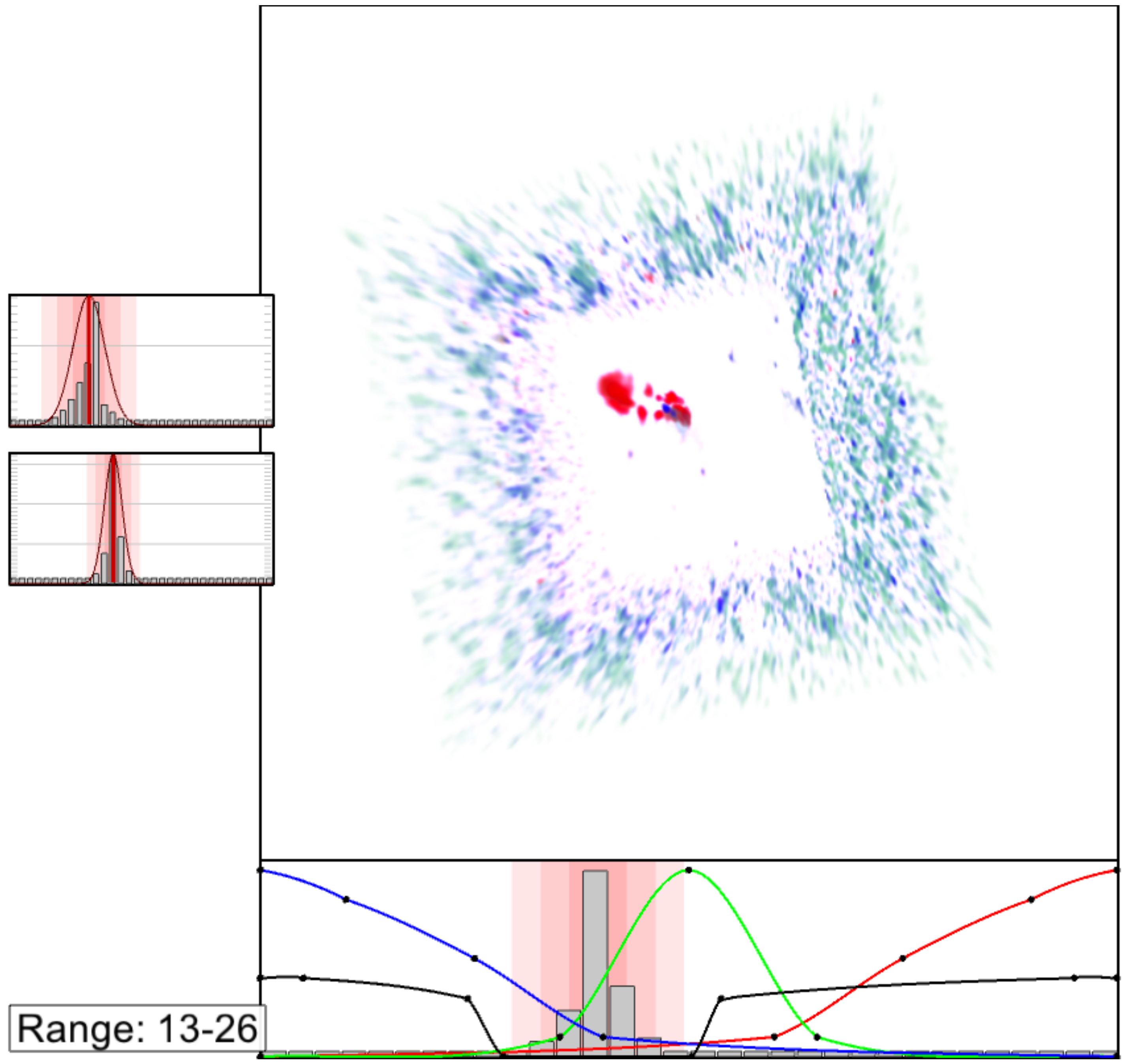


simplified

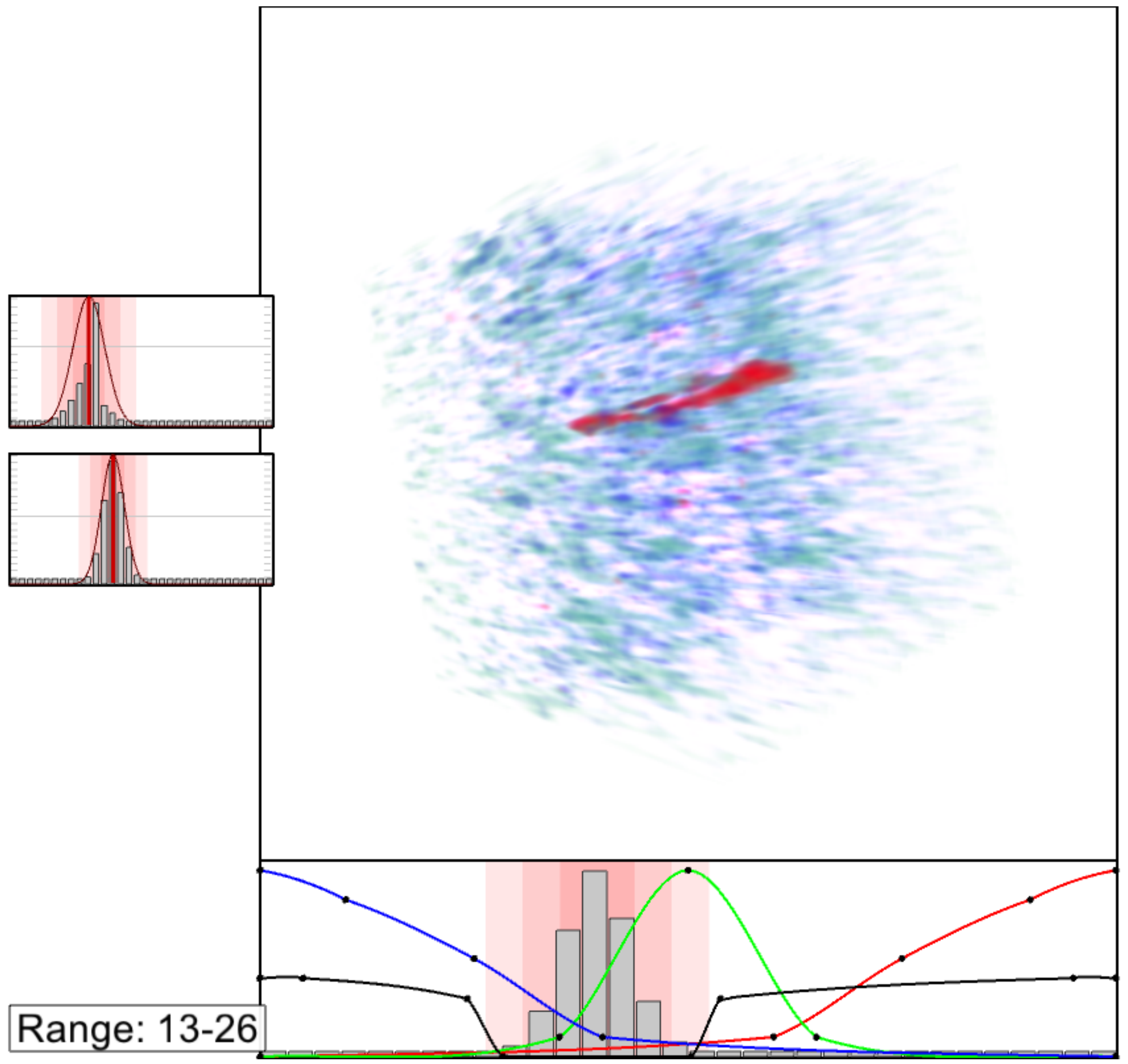


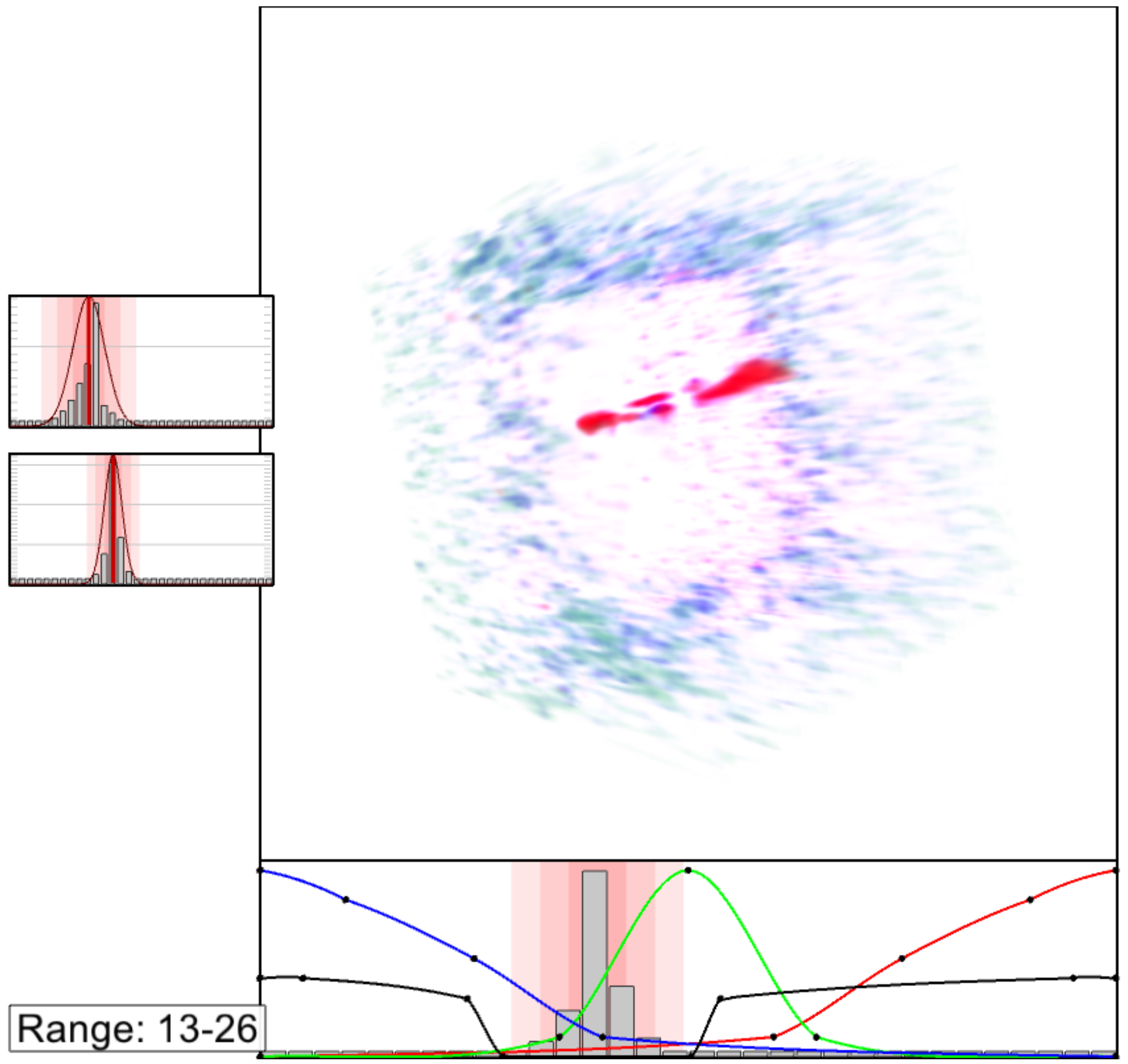
Observing the red shift



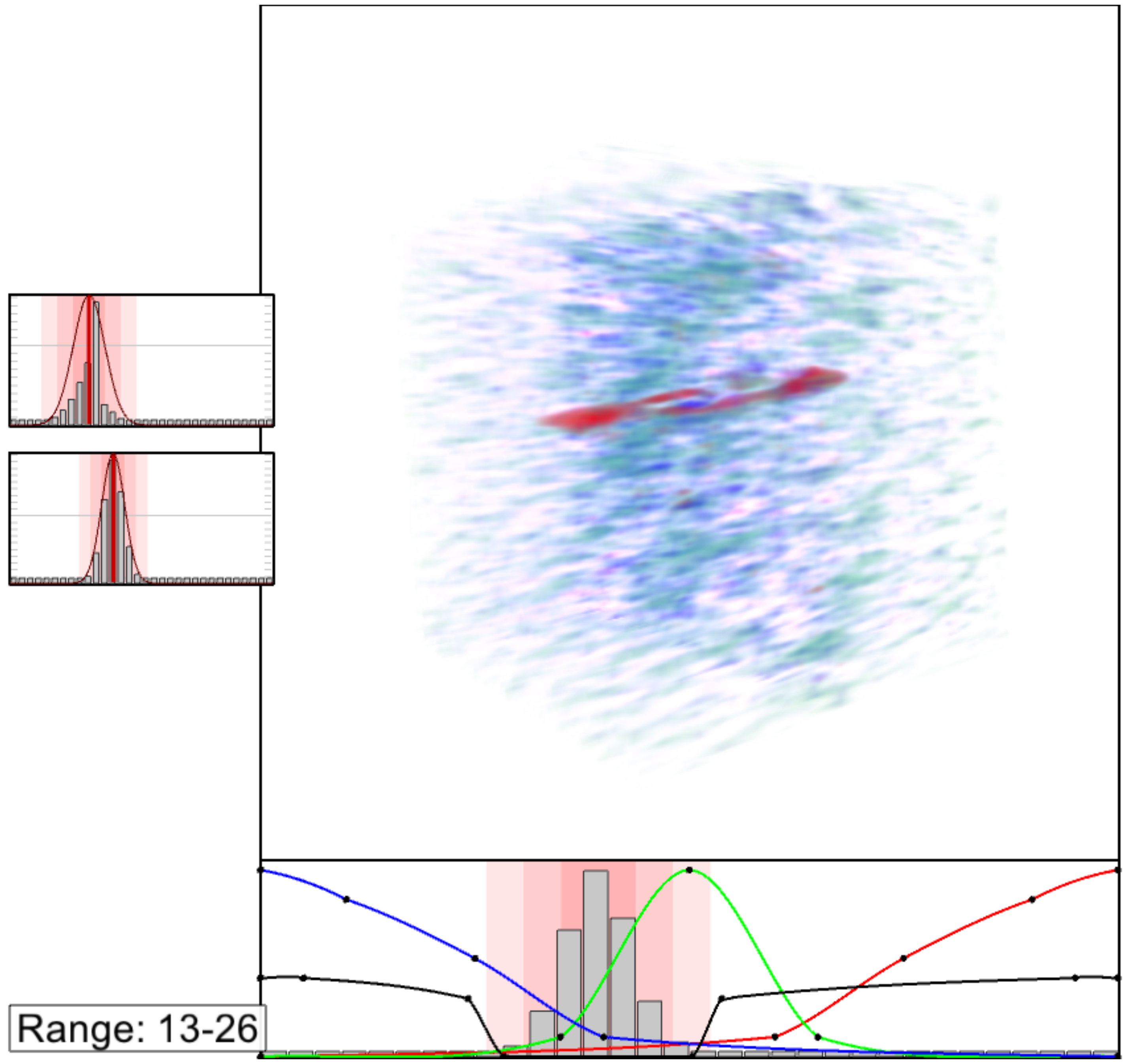


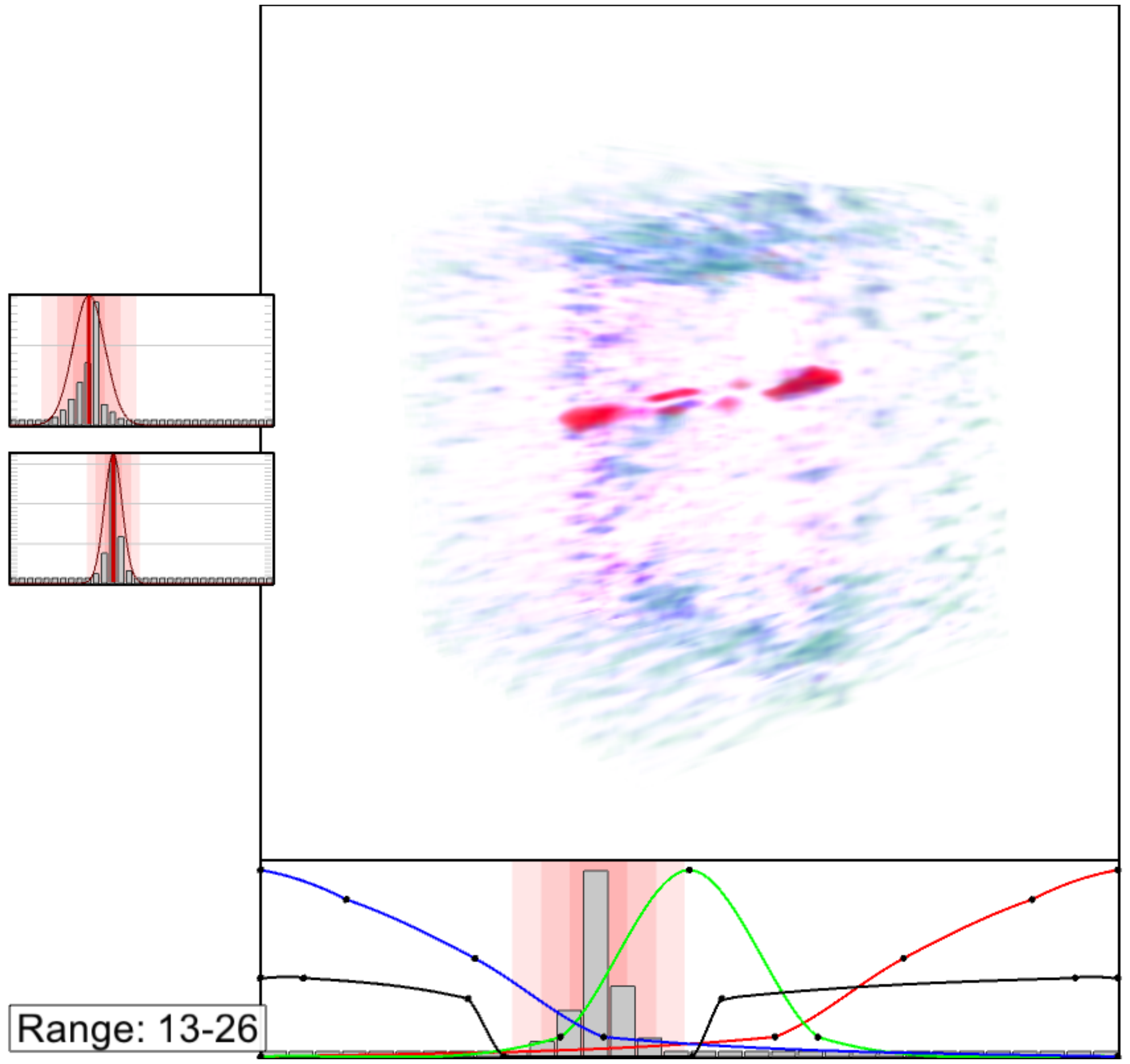
Observing the red shift





Observing the red shift



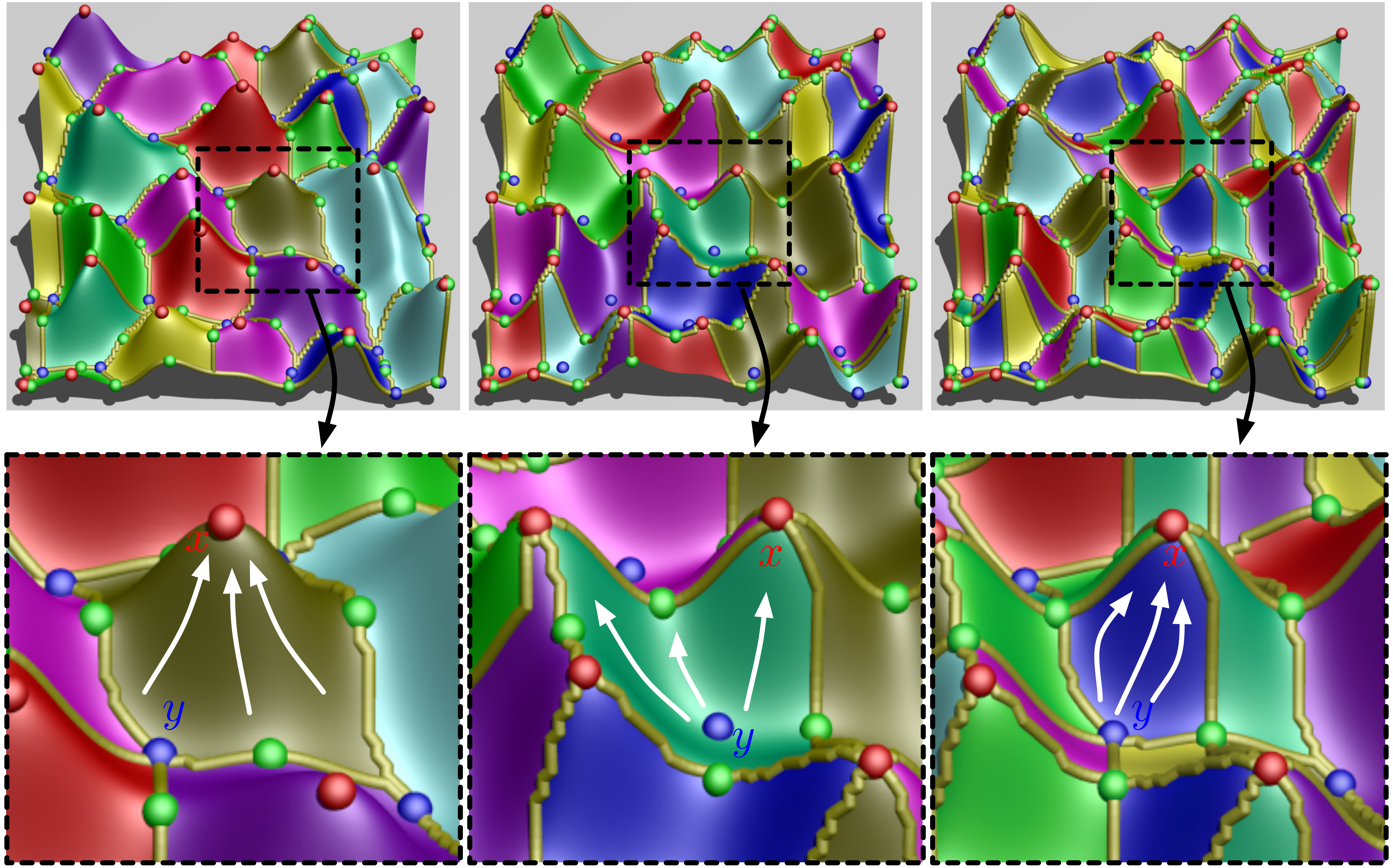


Morse-Smale Complex (MSC)

A review and application stories

MSC

Elevation on a terrain: function on a 2D domain



(a) (b) (c)

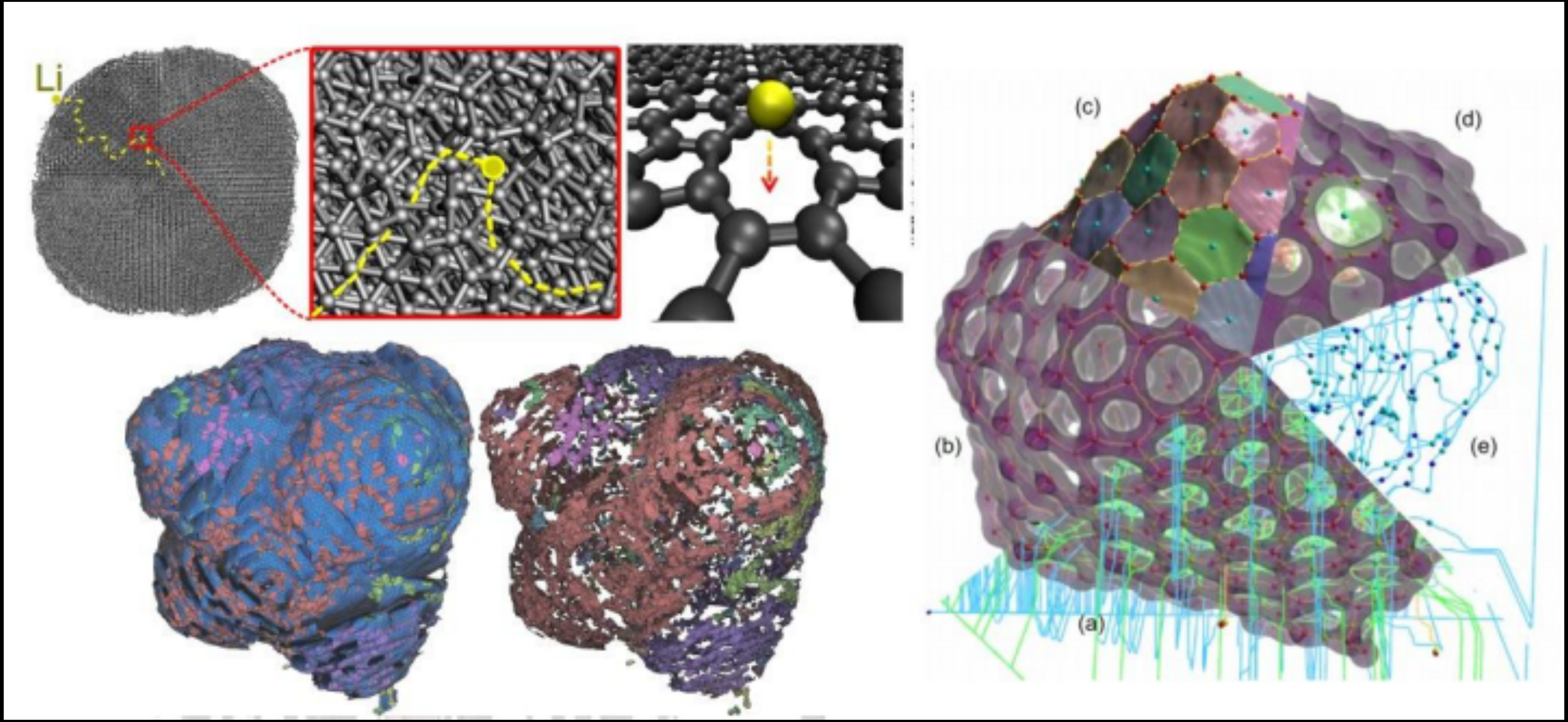
Case Study 1: Material Science

Battery

Application of MSC

How long can
your battery last?

Ion diffusion geometry extraction in battery



Case Study 2:
Data simplification, and more

Application of MSC

Simplify terrain data

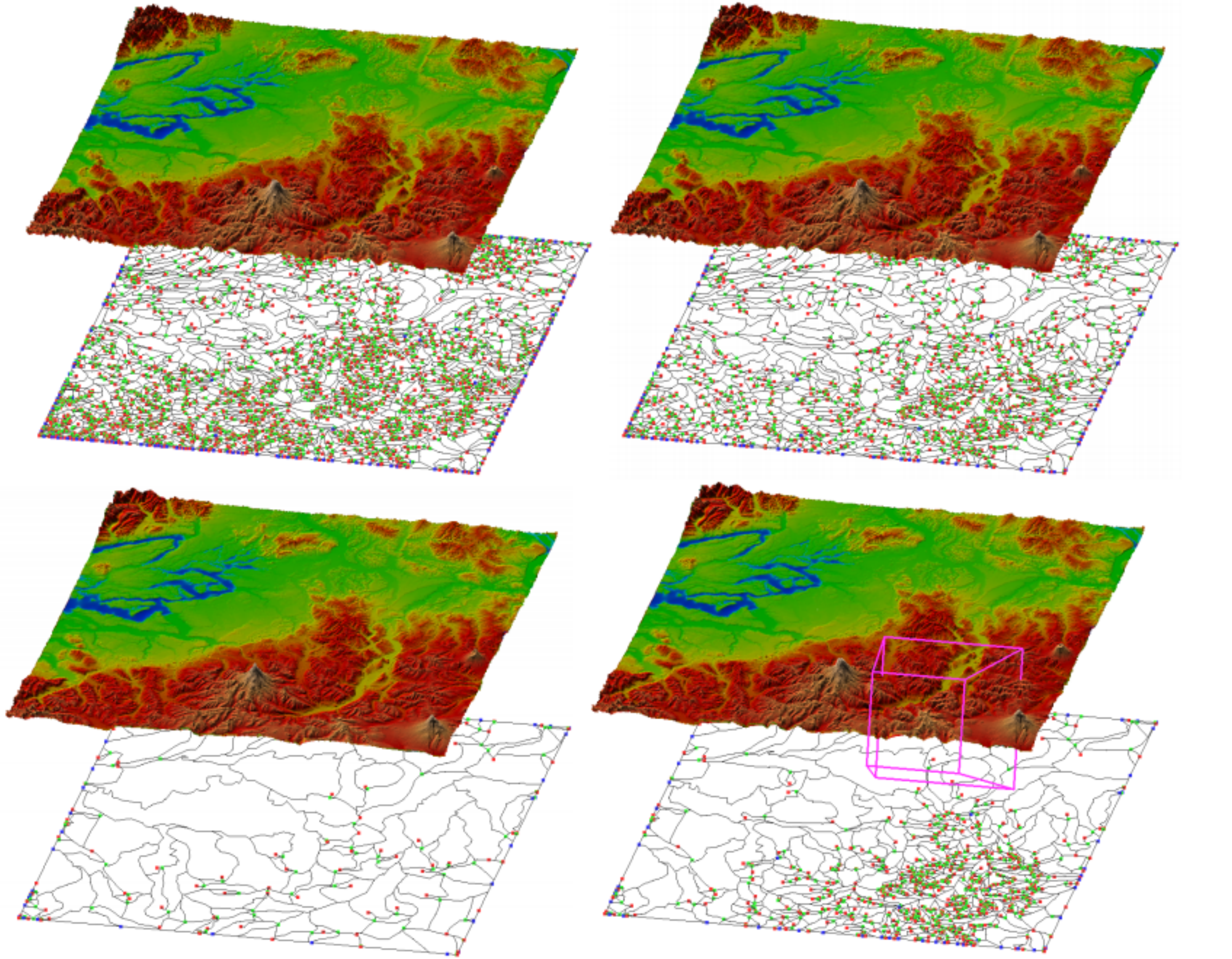
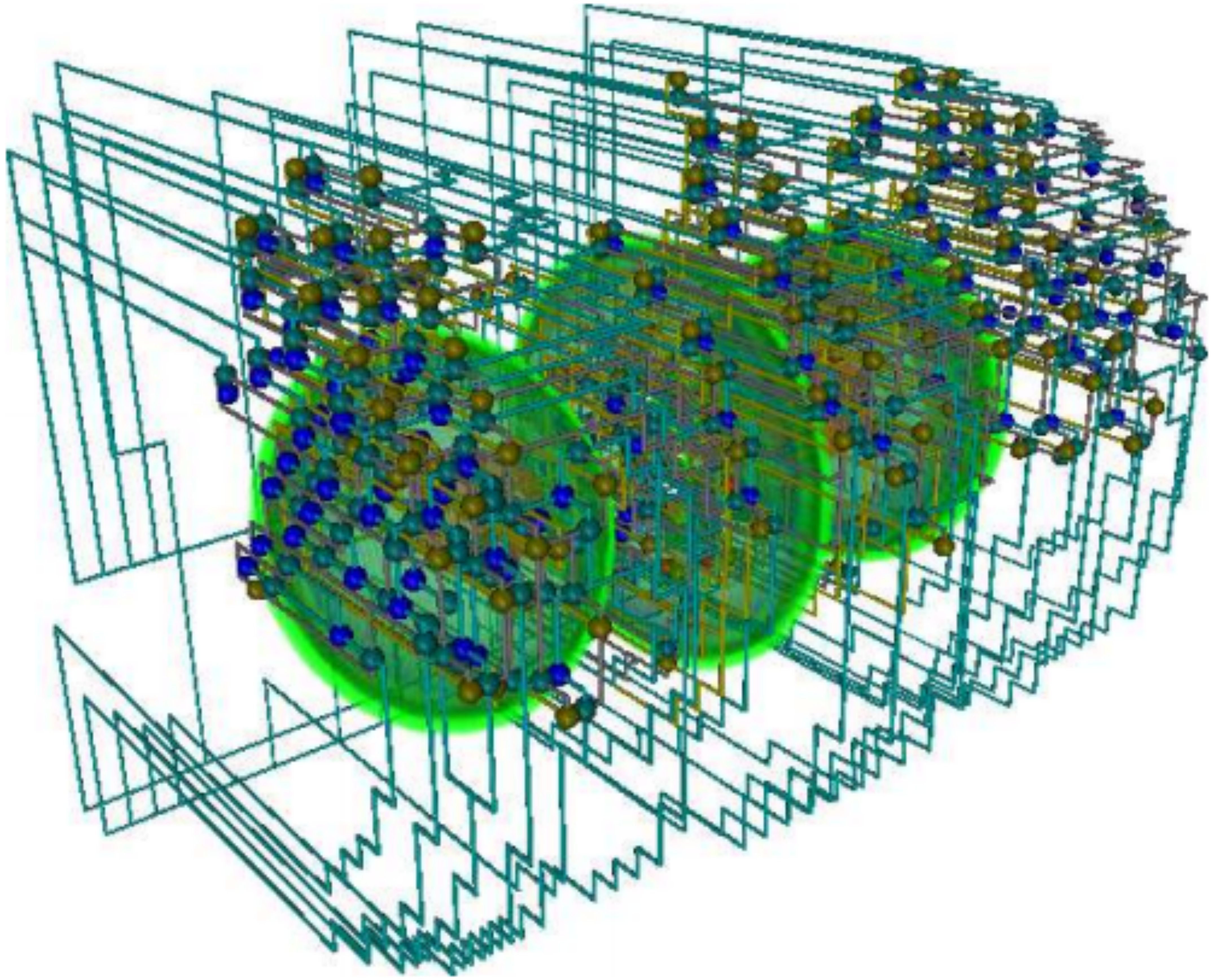
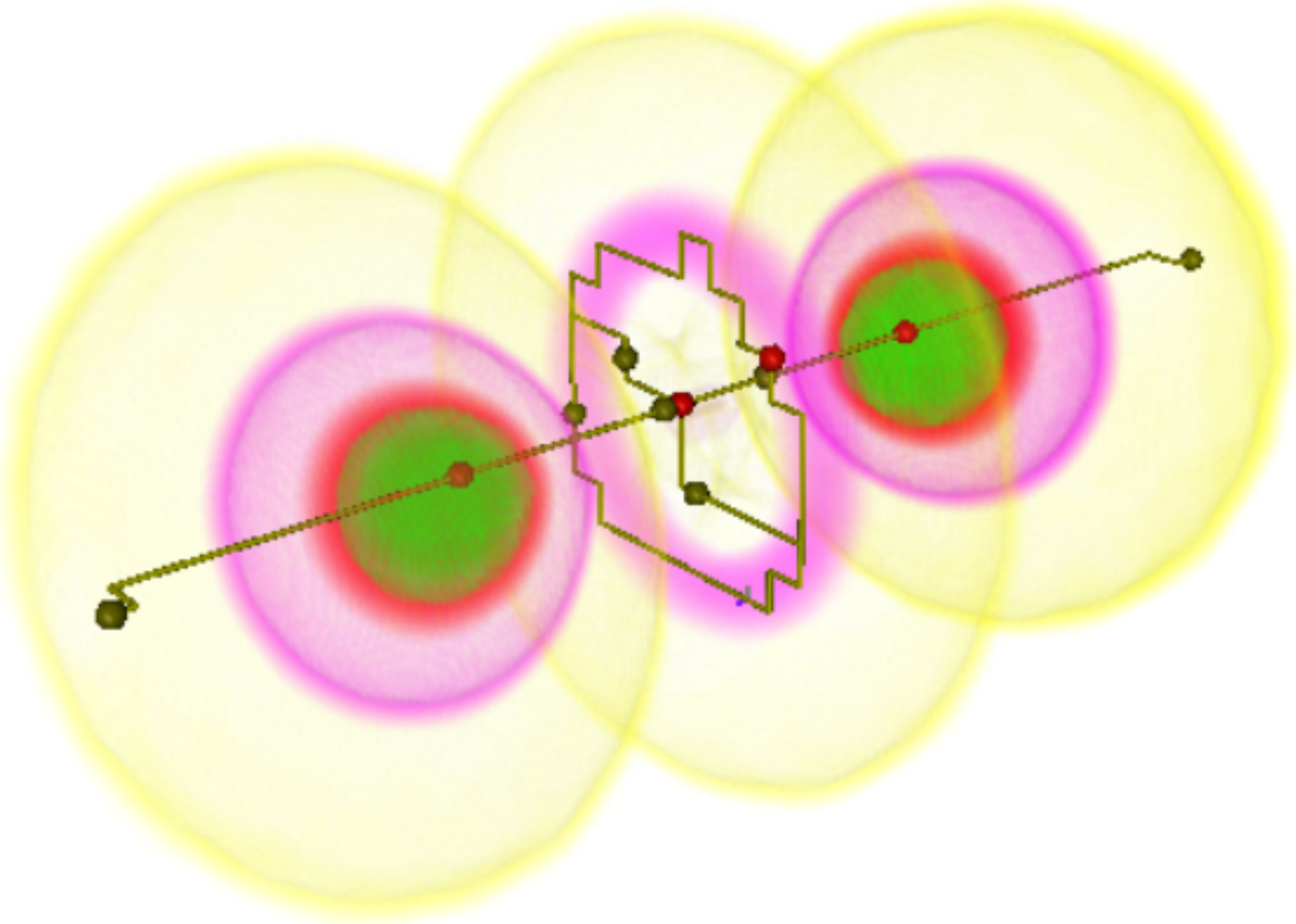


Figure 11: (Upper-left) Puget Sound data after topological noise removal. (Upper-right) Data at persistence of 1.2% of the maximum height. (Lower-left) Data at persistence 20% of the maximum height. (Lower-right) View-dependent refinement (purple: view frustum).

P.-T Bremer, H. Edelsbrunner, B. Hamann and V. Pascucci, 2003

Simplify Electron Density Data

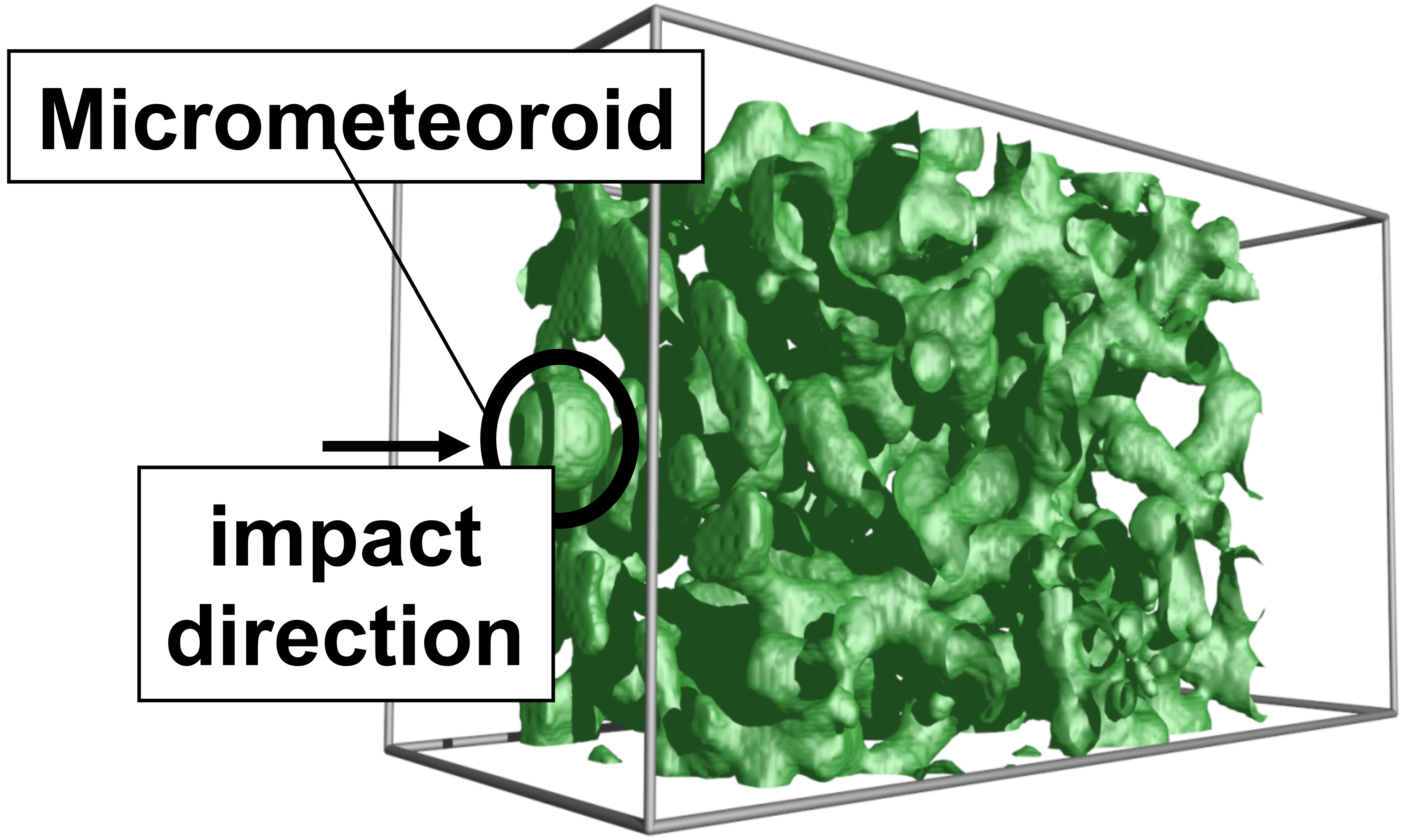


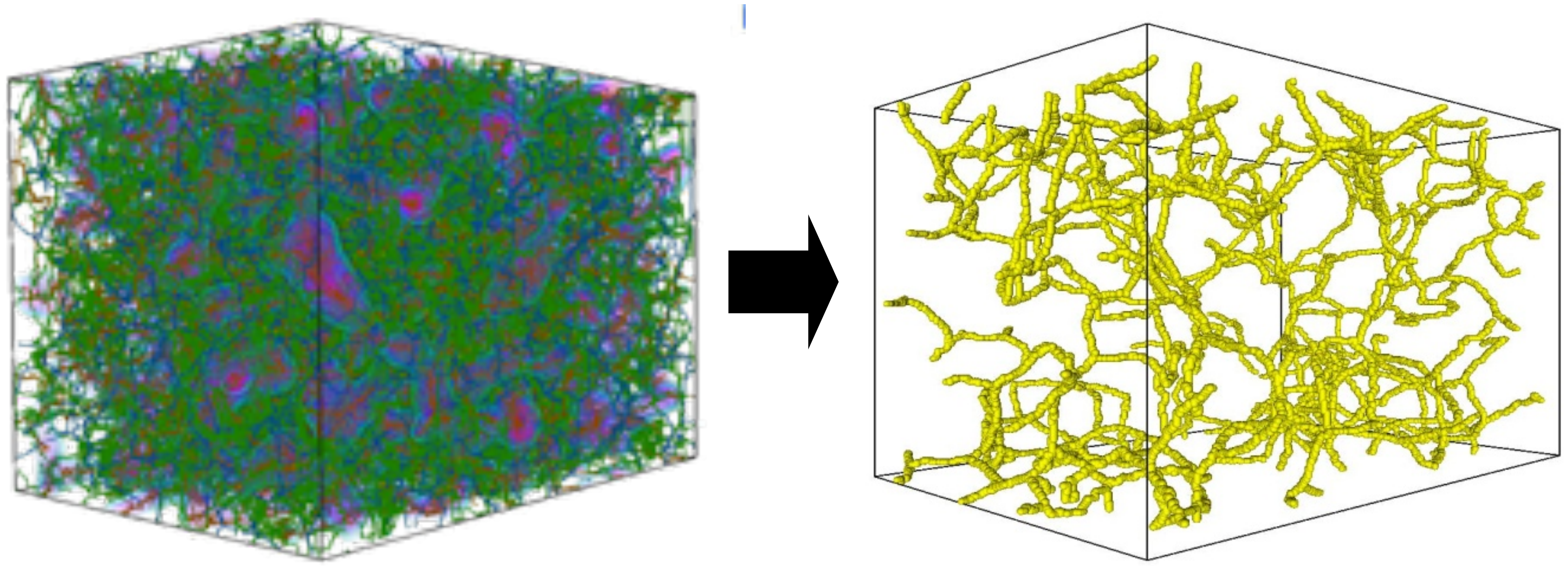


A. Gyulassy, V. Natarajan, V. Pascucci, P.-T. Bremer, B. Hamann,2005

Case Study A: Reconstructing porous material

Quantitative Analysis of the Impact of a Micrometeoroid in a Porous Medium; reconstructing the structure of porous medium

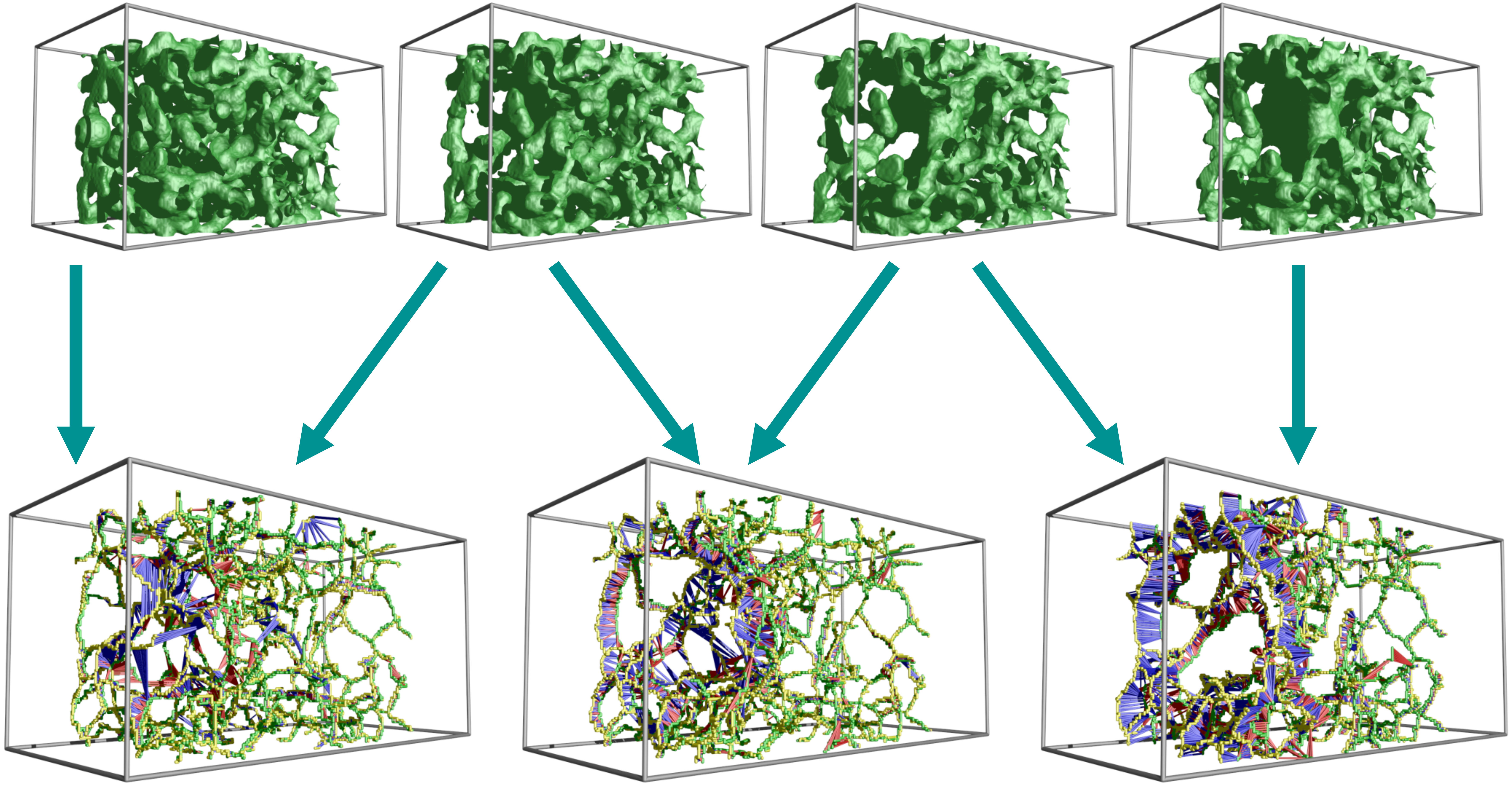




Source: Valerio Pascucci

Case study A: Porous Medium

We Track the Evolution of the Filament Structure of the Material Under Impact

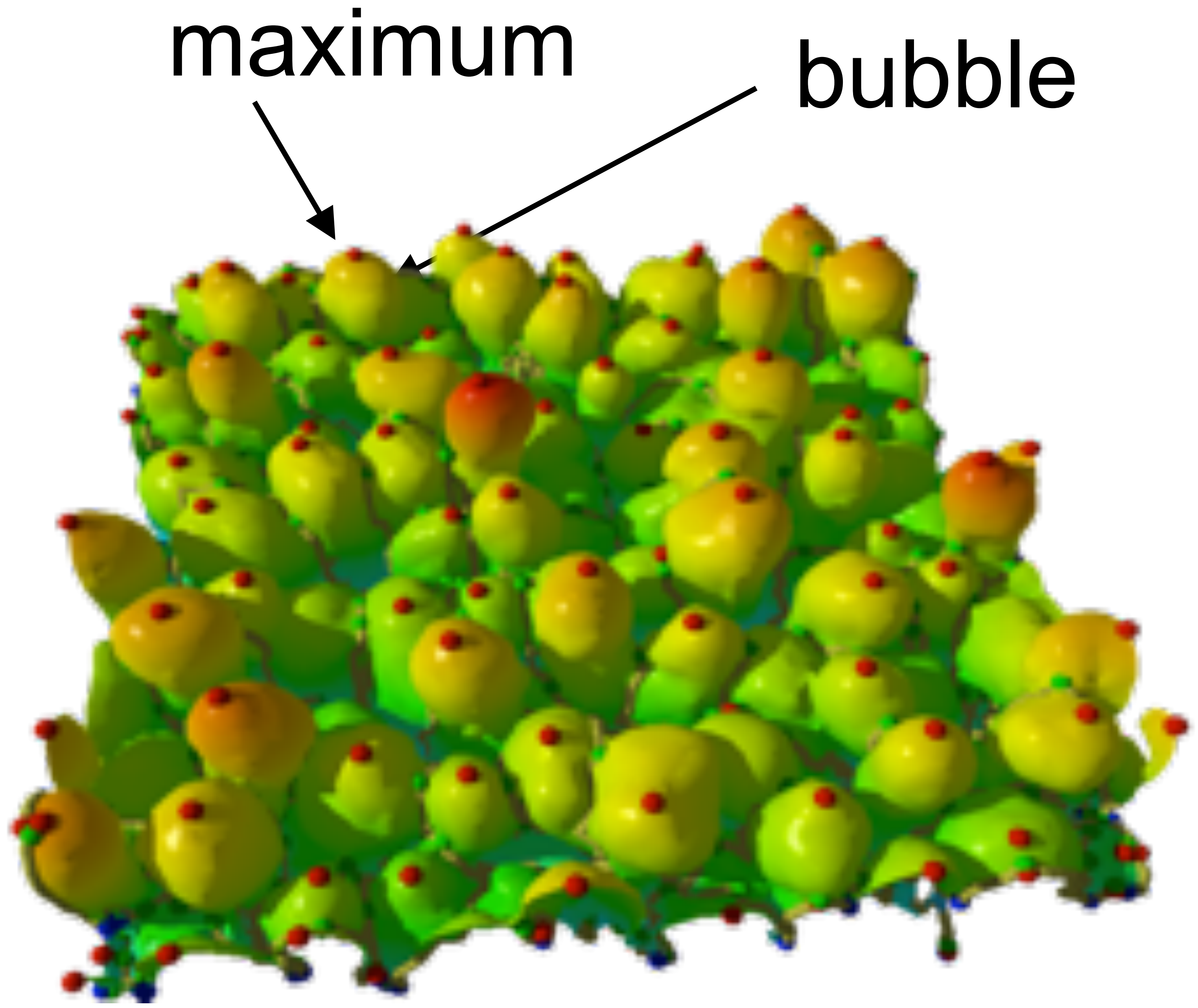


Time comparison of the reconstructions

Source: Valerio Pascucci

Case study B: feature definition - Bubble Tracking

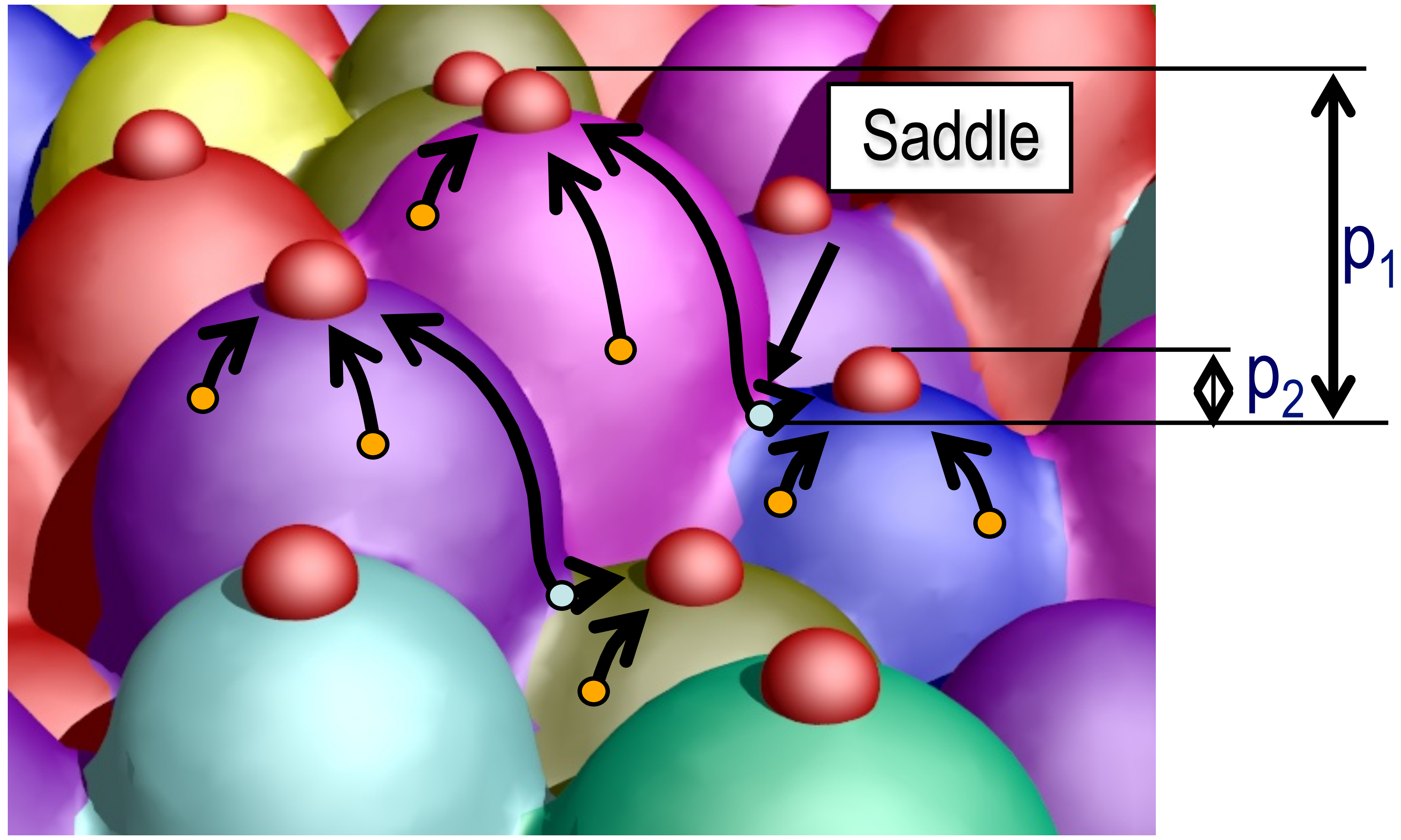
Analyze high-resolution Rayleigh Taylor instability simulations

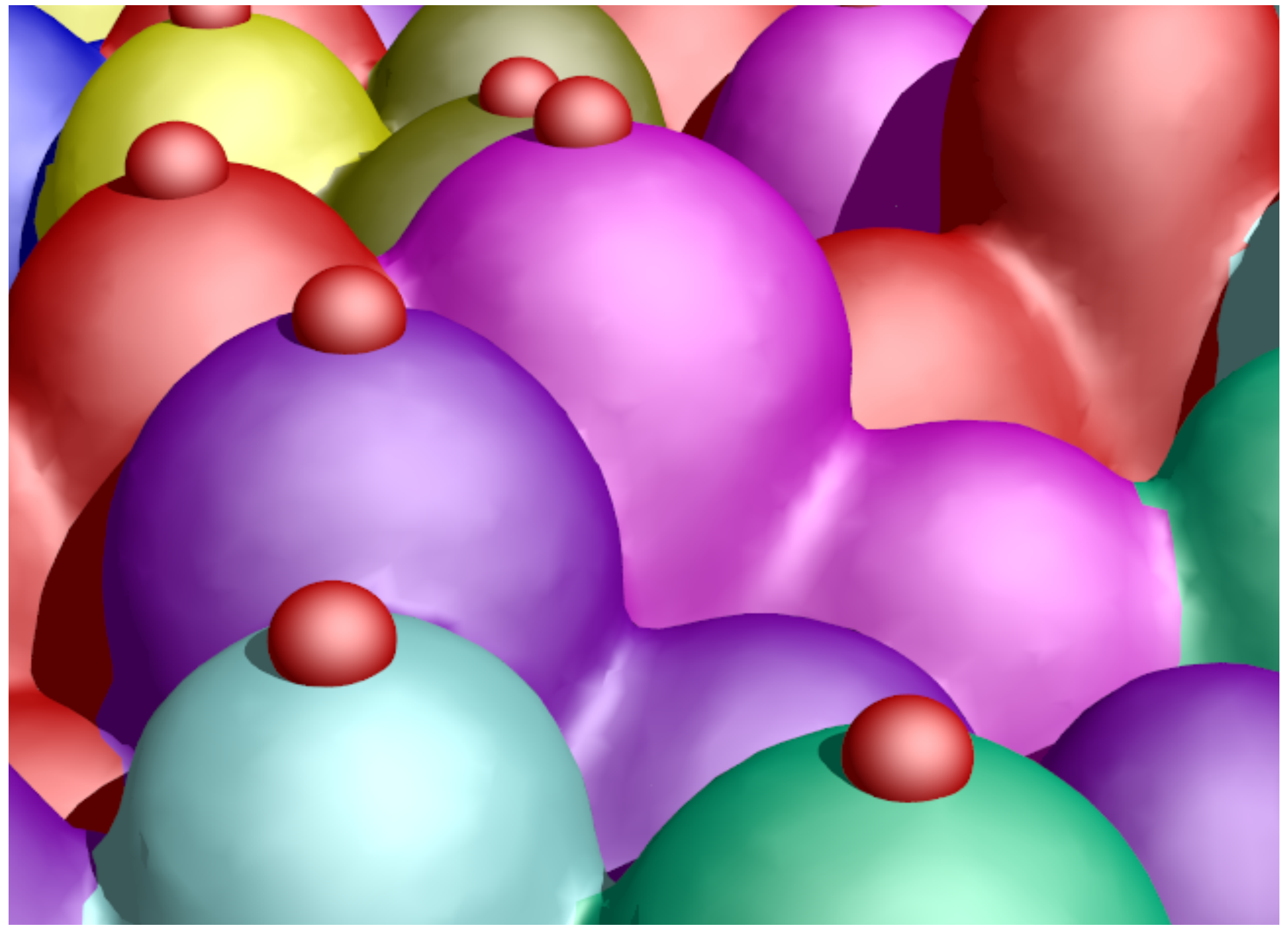


Source: Valerio Pascucci

Case study B: persistence simplification

Analyze high-resolution Rayleigh Taylor instability simulations

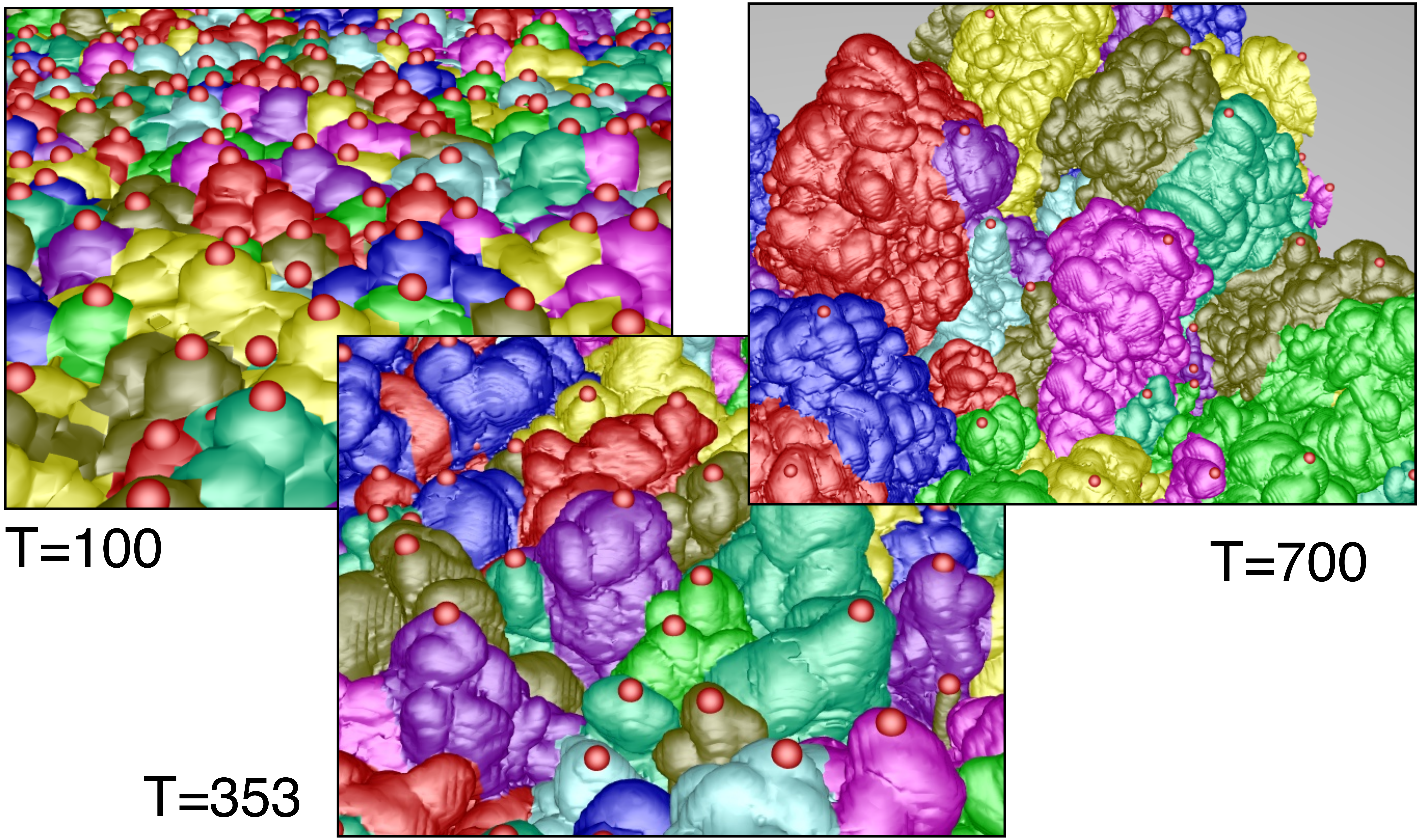




Source: Valerio Pascucci

Case study B: robust segmentation

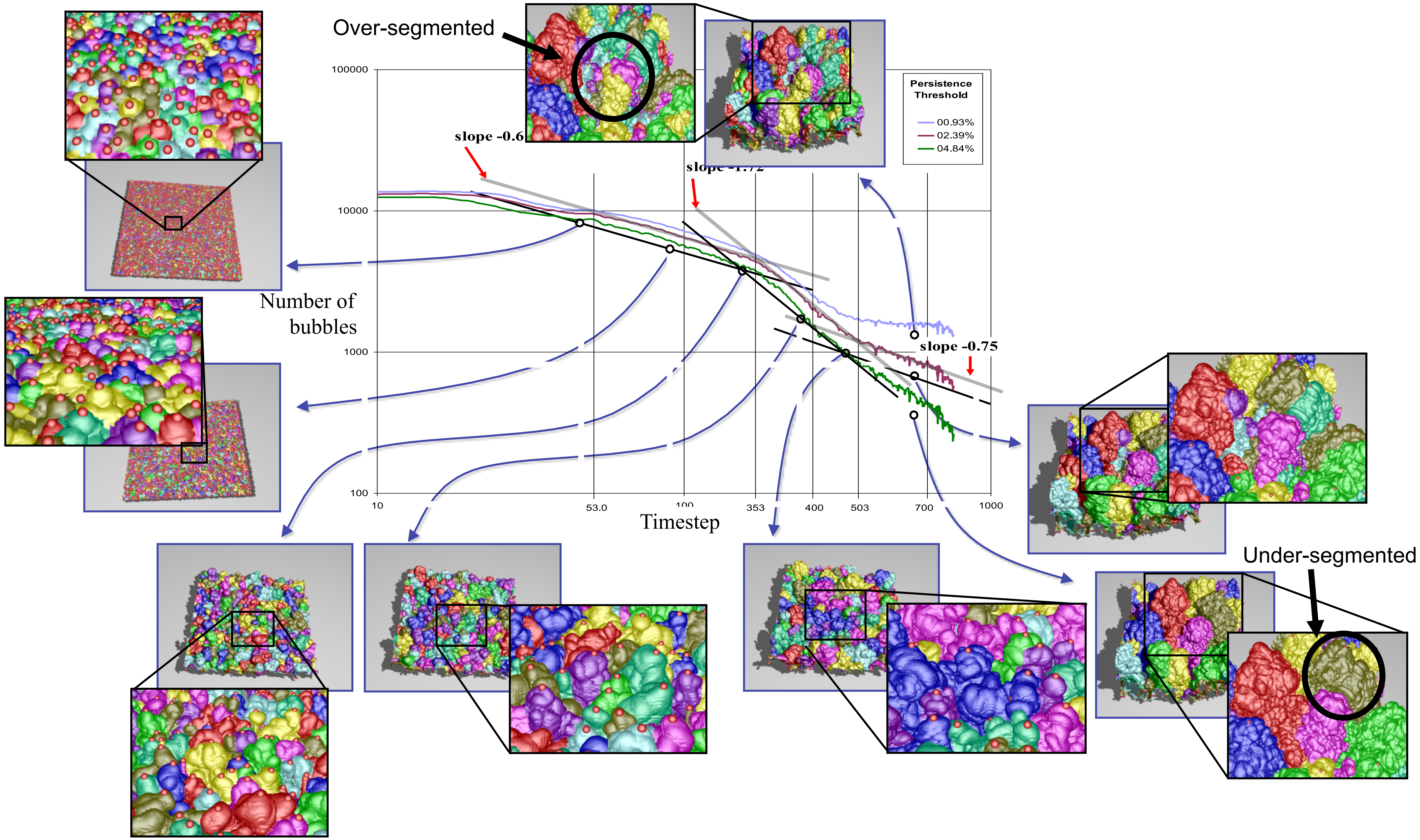
The segmentation method is robust from early mixing to late turbulence



Source:
Valerio
Pascucci

Case study B: multiple scales

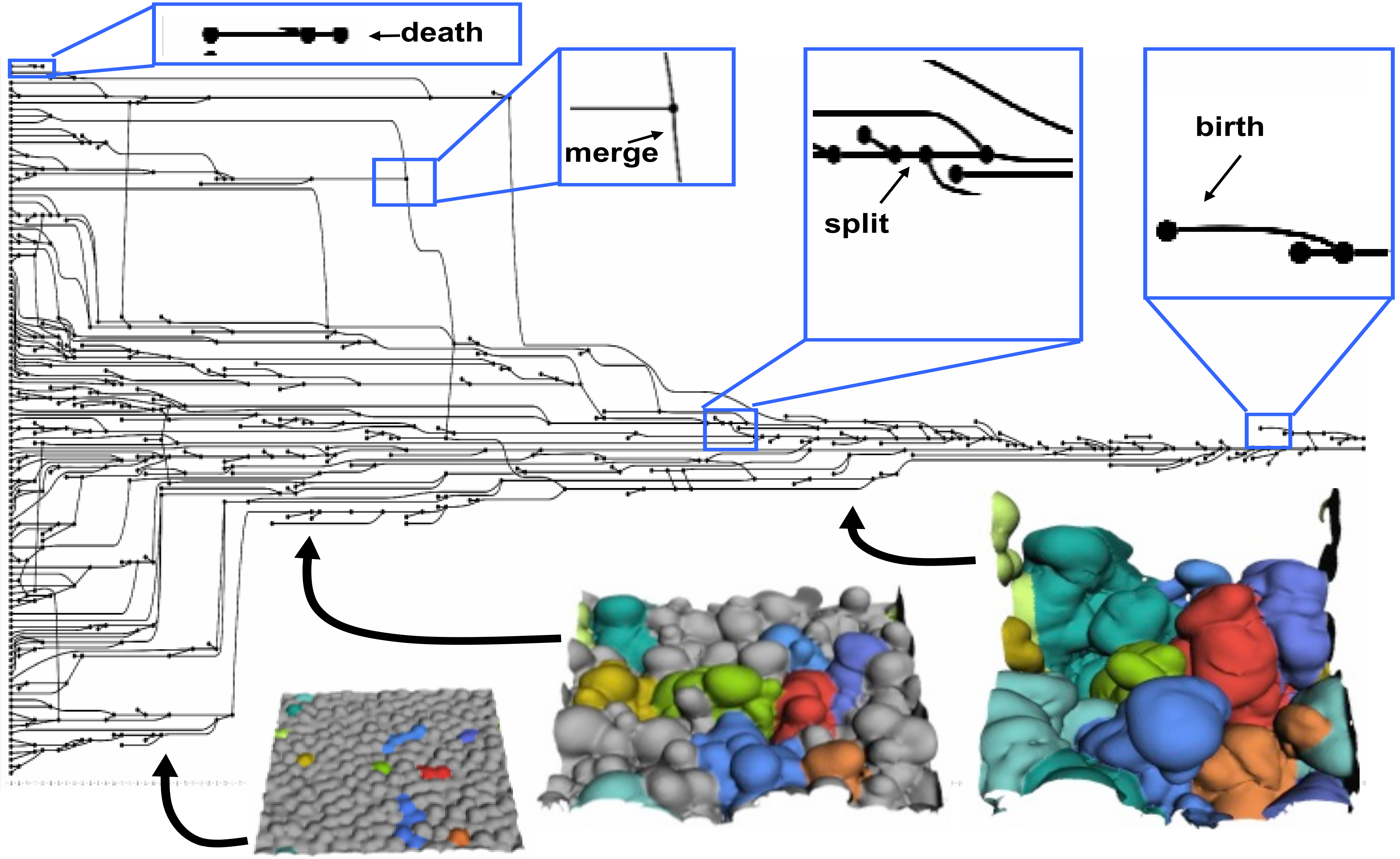
We Evaluated Our Quantitative Analysis at Multiple Scales



Source: Valerio Pascucci

Case study B: event characterization

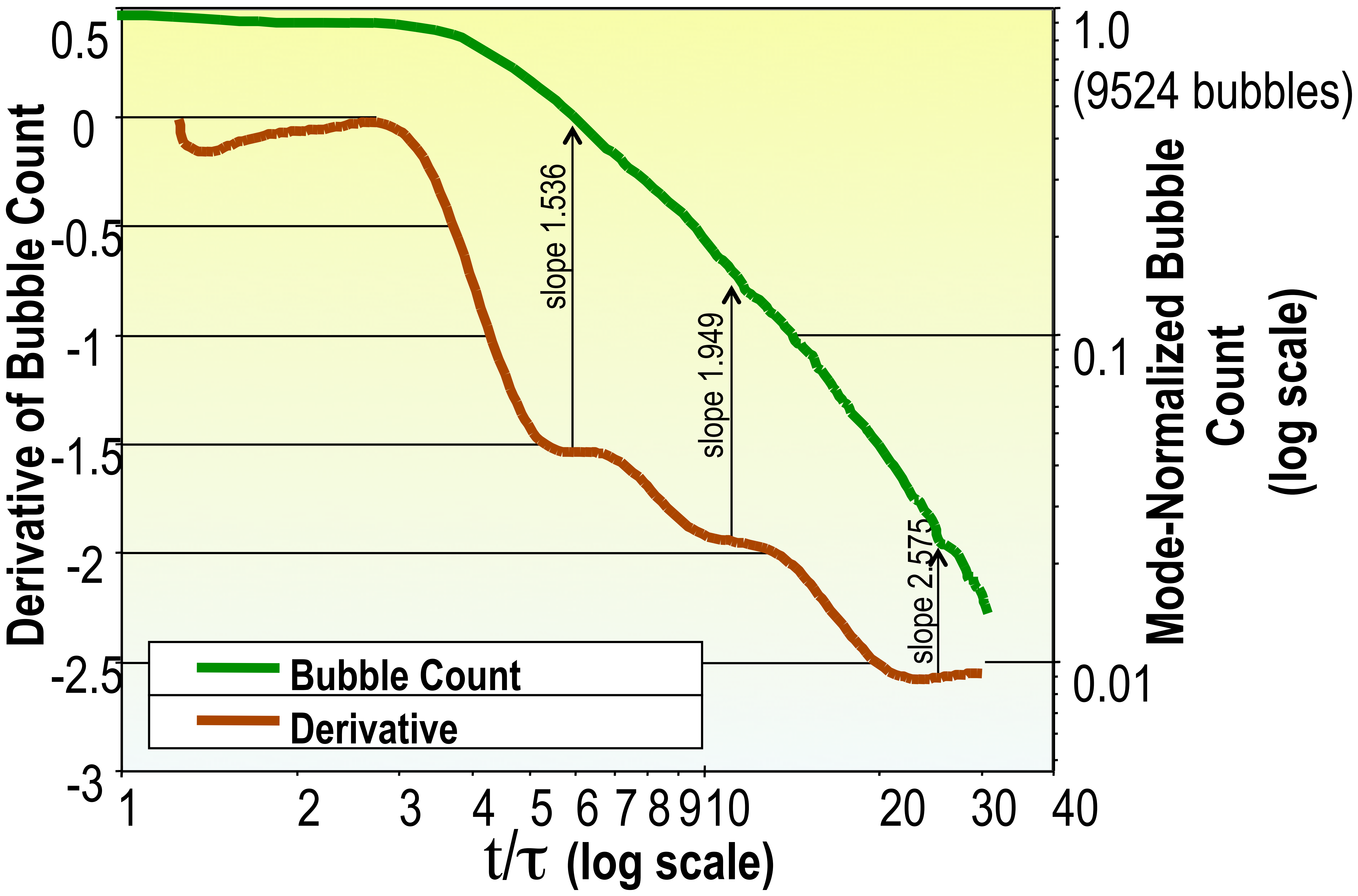
We characterize events that occur in the mixing process



Source: Valerio Pascucci

Case study B: Exciting Result

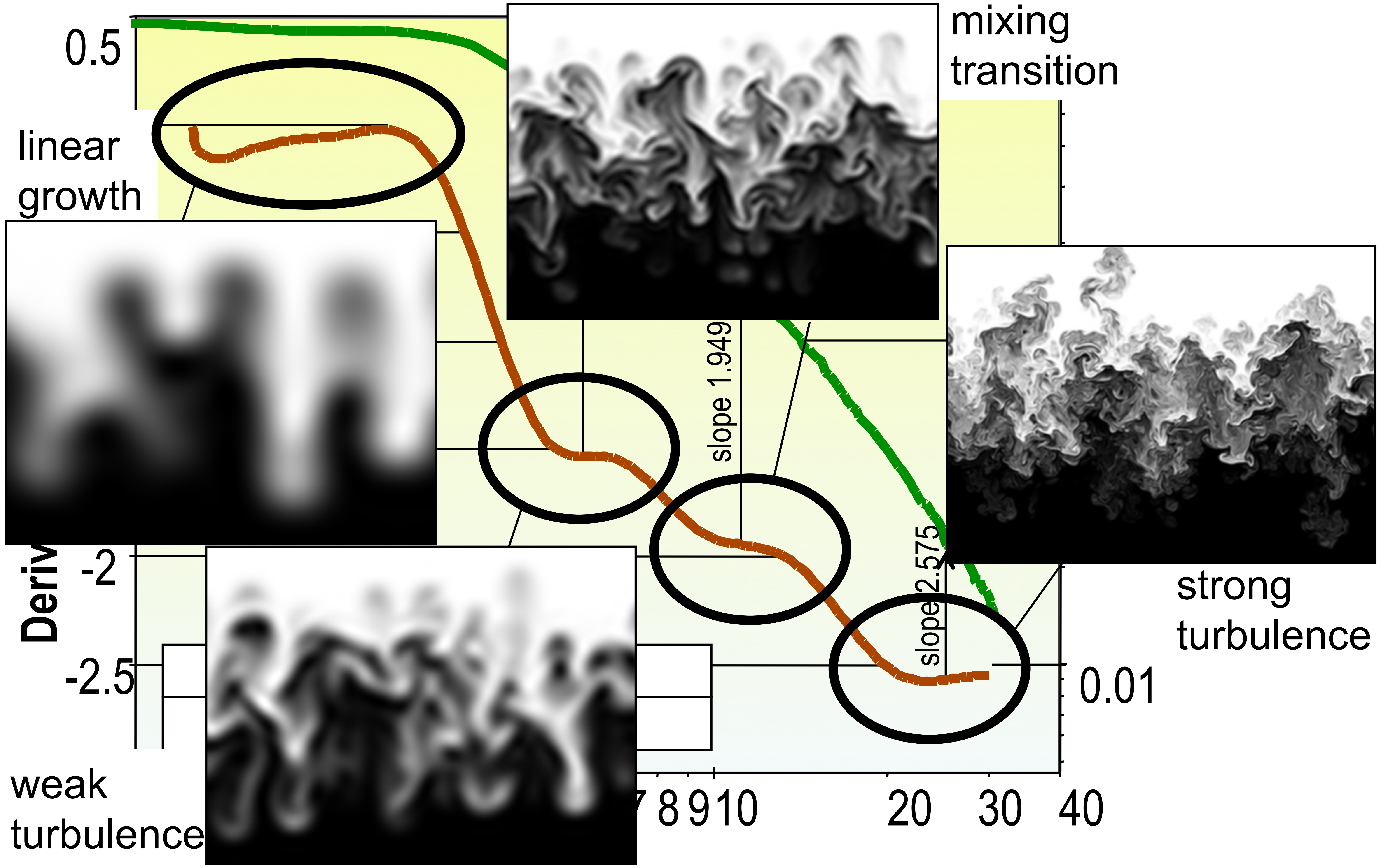
First Time Scientists Can Quantify Robustly Mixing Rates by Bubble Count



Source: Valerio Pascucci

Case study B: Exciting Result

We Provide the First Quantification of Known Stages of the Mixing Process



Source:
Valerio
Pascucci

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

🟦 🟦 🟨 🟩 ⬛