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!

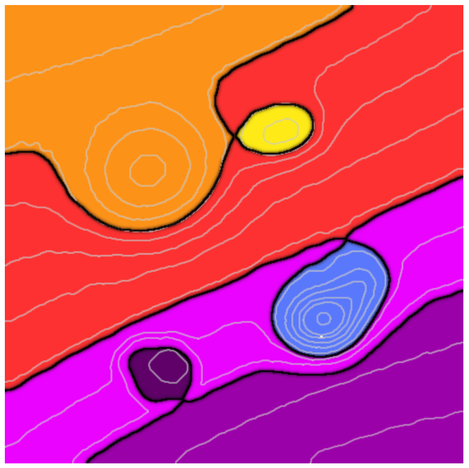
Topological structures and TTK

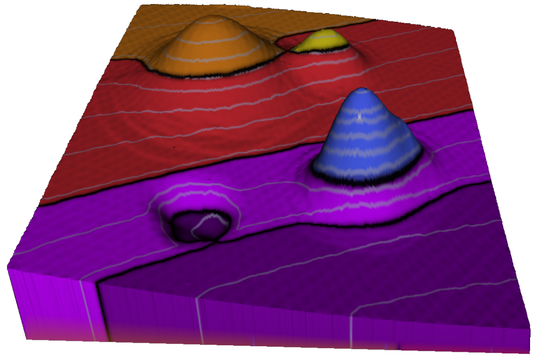
TOPO

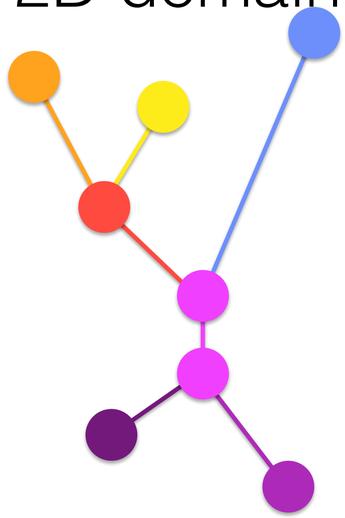
Review: Contour Trees and Morse-Smale Complex

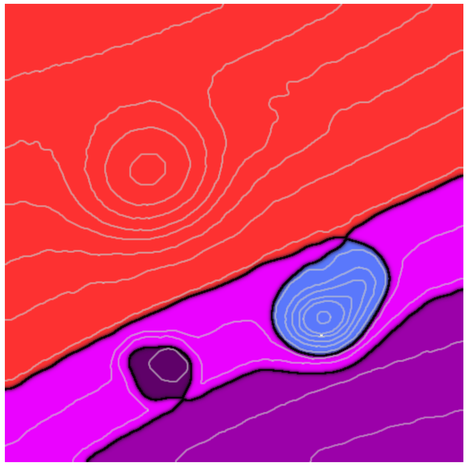
Data has shape

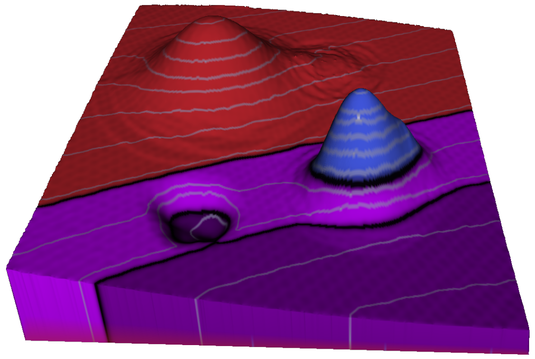
Elevation on a terrain: function on a 2D domain

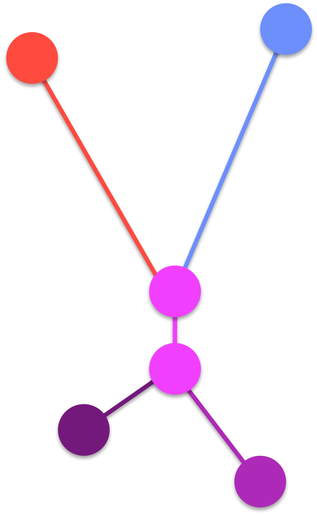










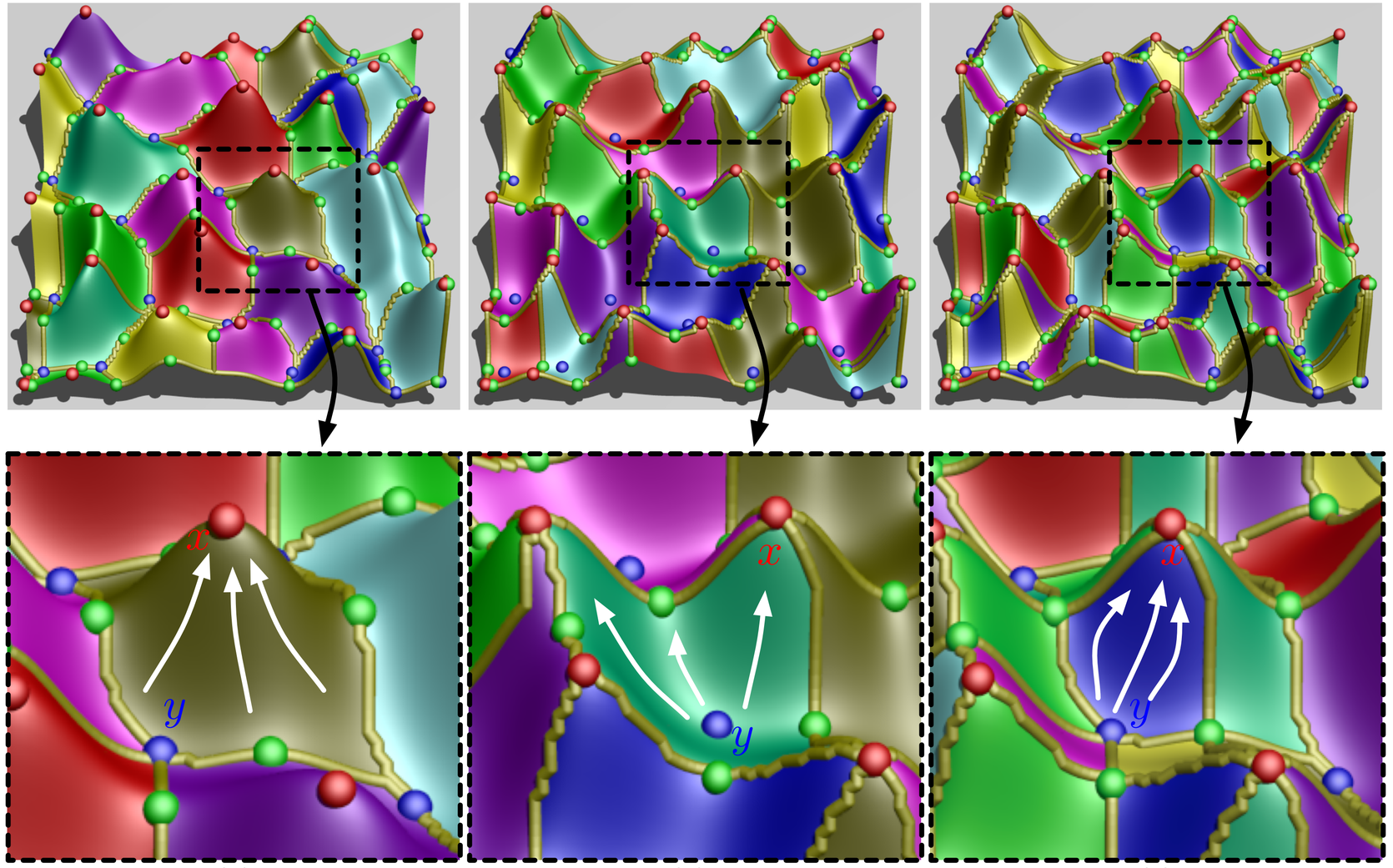


Shape of data?

Contour Tree

Data has shape

Elevation on a terrain: function on a 2D domain



Shape of data?

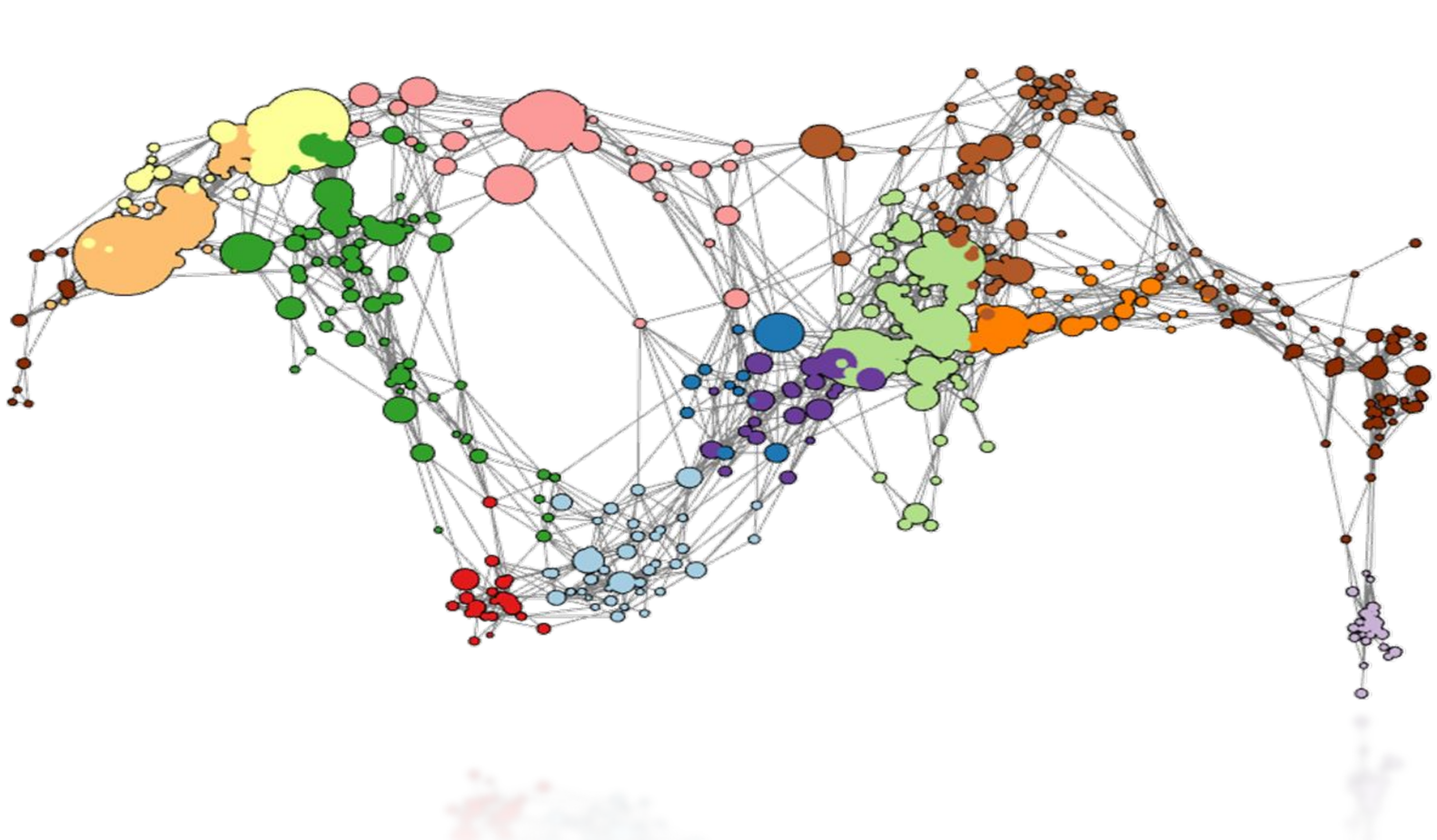
Morse-Smale Complex

A Map of Science Example

Map of Science?

554 subdiscipline belong to 13 scientific discipline

Map of Science



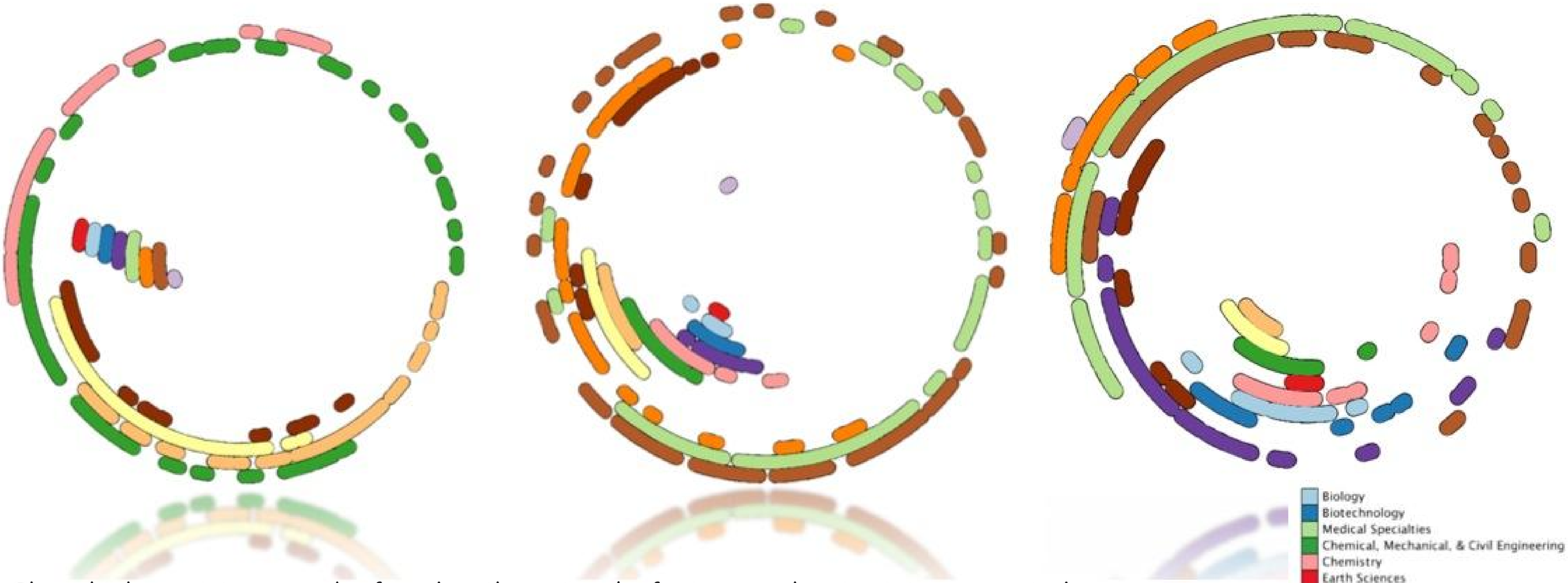
- Biology
- Biotechnology
- Medical Specialties
- Chemical, Mechanical, & Civil Eng
- Chemistry
- Earth Sciences
- Electrical Engineering & Compute
- Brain Research
- Humanities
- Infectious Diseases
- Math & Physics
- Health Professionals
- Social Sciences

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”.

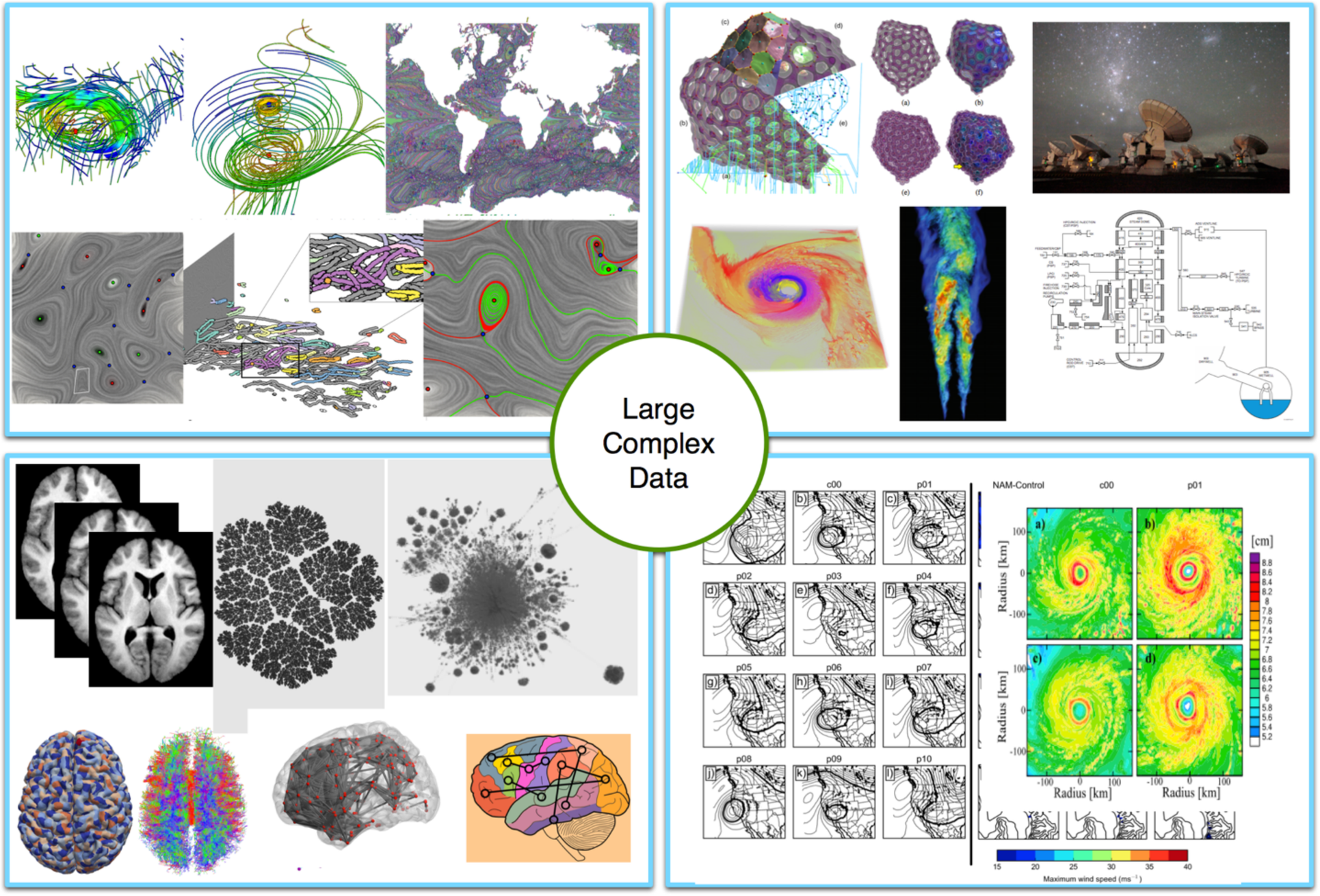
With TDA: what is the shape of the map of science?

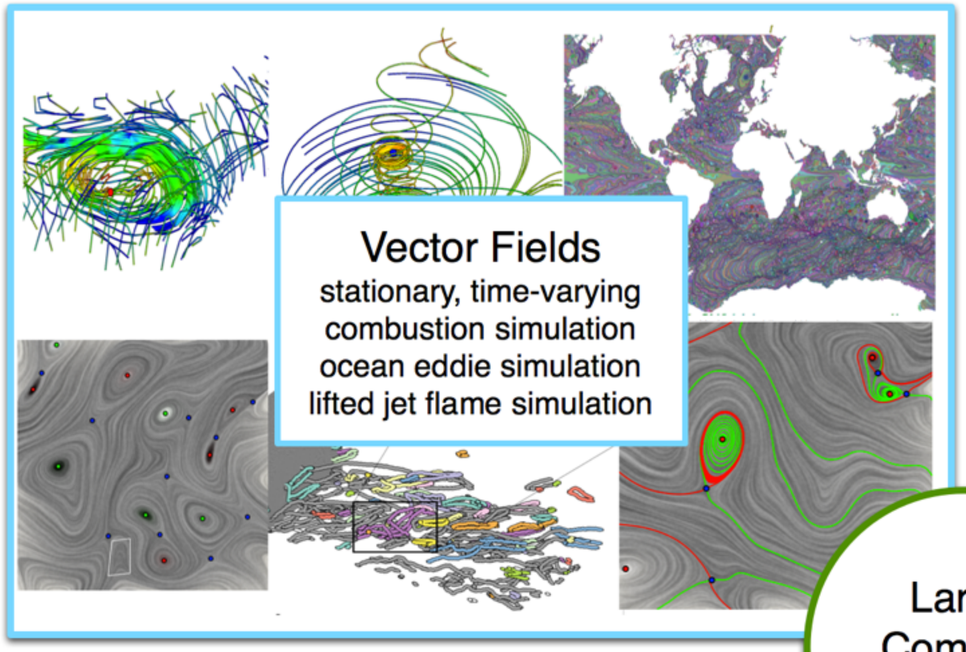


Three high persistence cycles found on the network of science – showing interconnectedness of specialties – left: core science and engineering, middle: healthcare, right: science (non-engineering)

Bei Wang, Carlos Scheidegger, Paul Rosen, unpublished

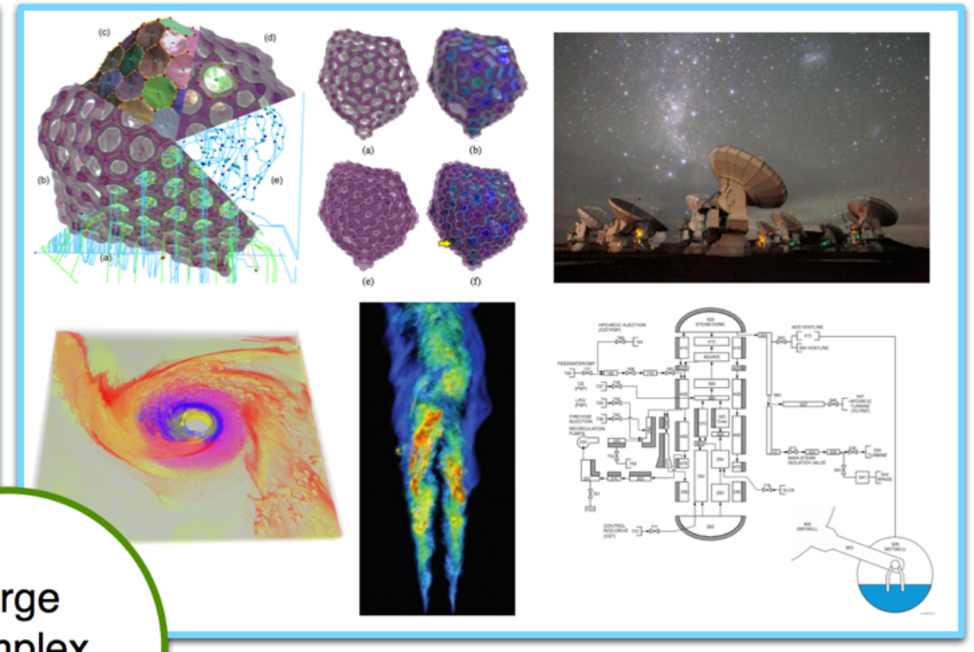
What is data?


Large
Complex
Data
NAM-Control
c00
p01
p02
p03
p04
p05
p06
p07
p08
p09
p10
Radius [km]
[cm]
Maximum wind speed (ms⁻¹)

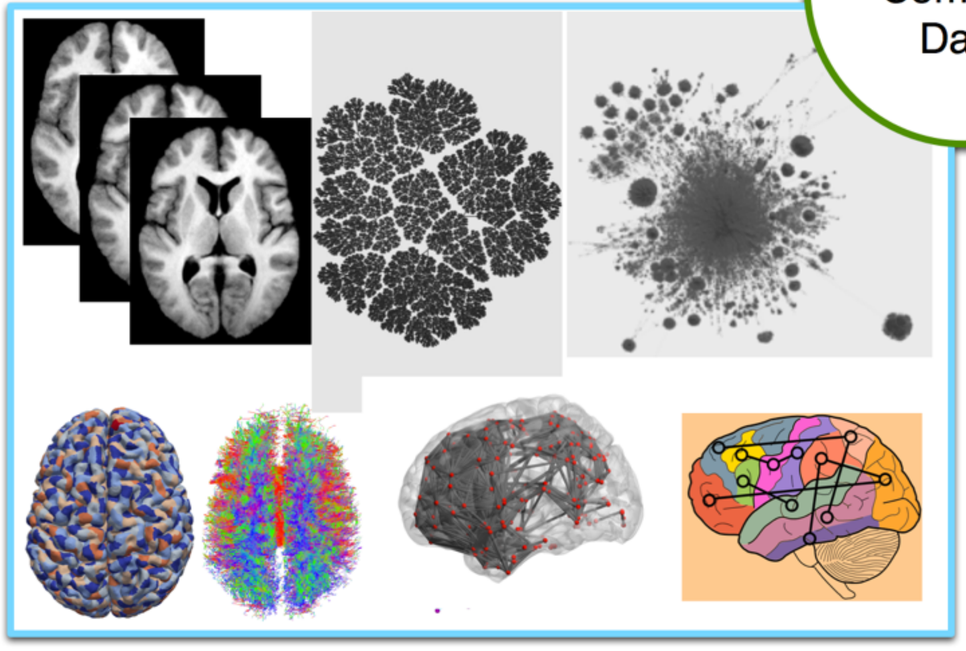


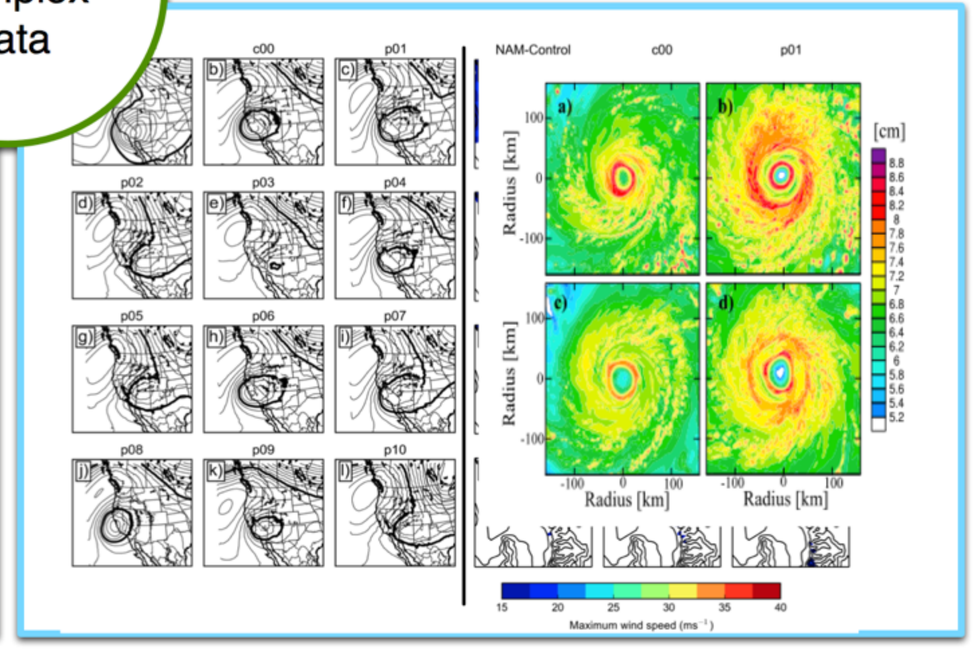
Vector Fields

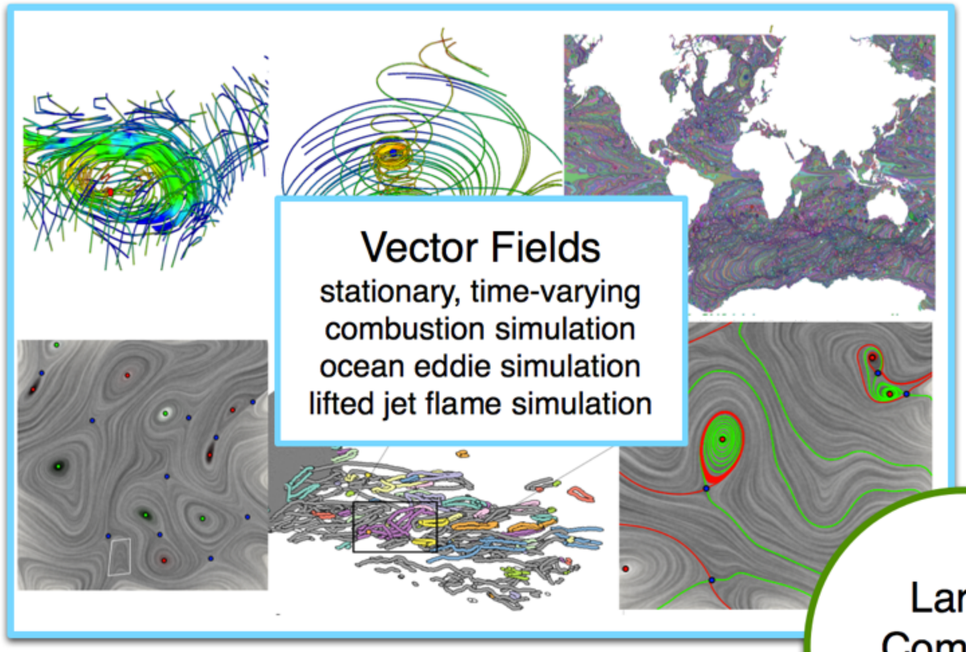
stationary, time-varying
combustion simulation
ocean eddie simulation
lifted jet flame simulation



Large Complex Data

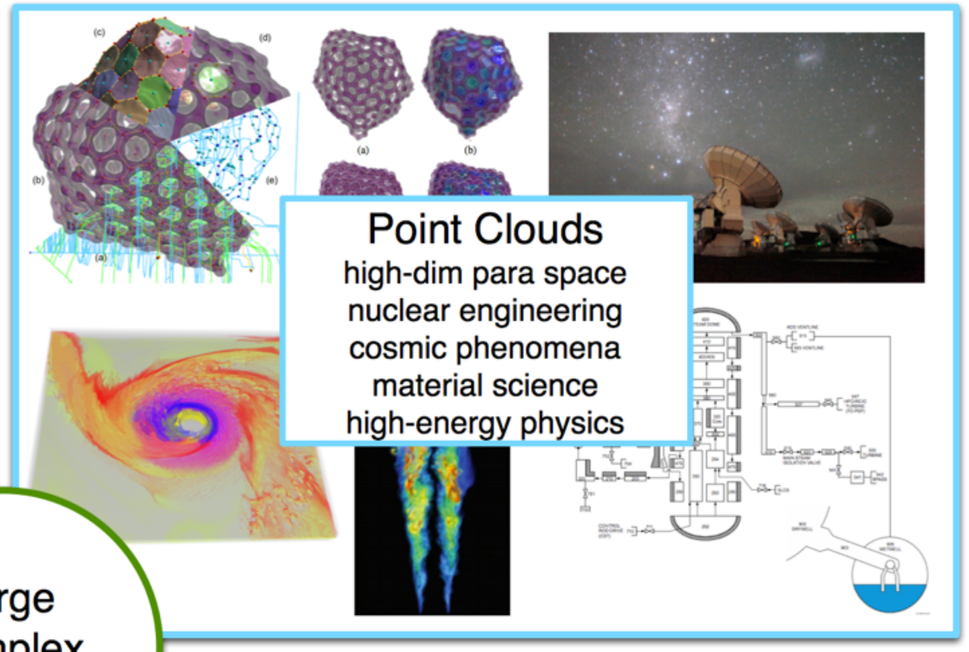






Vector Fields

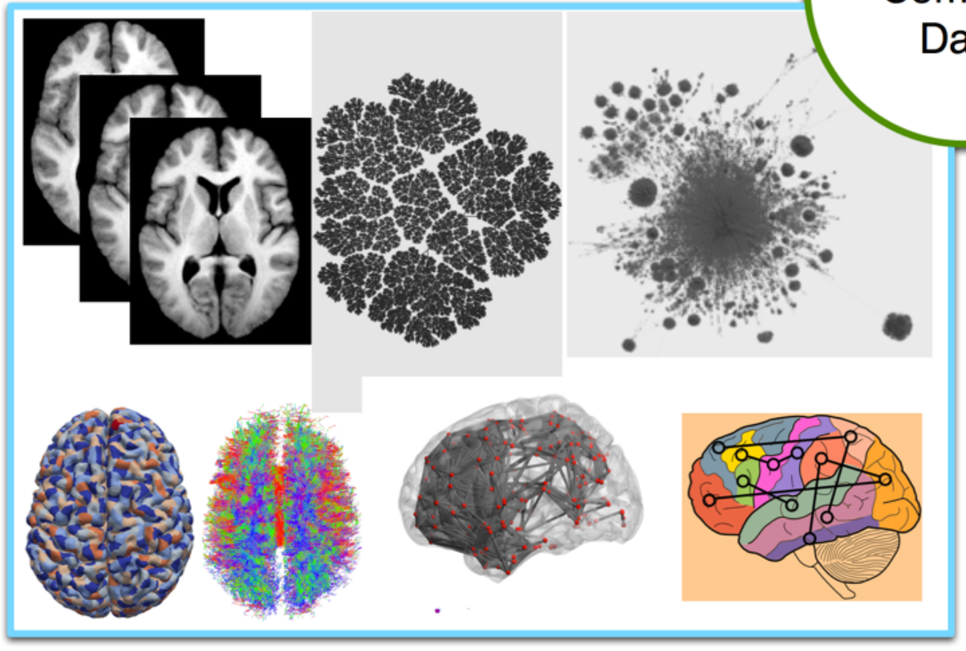
stationary, time-varying
combustion simulation
ocean eddie simulation
lifted jet flame simulation

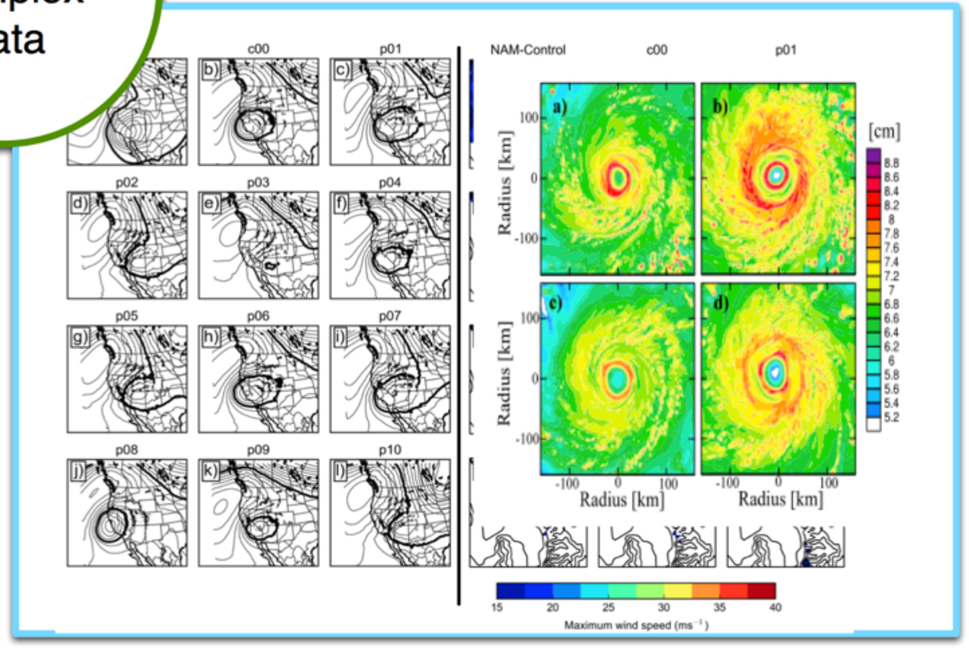


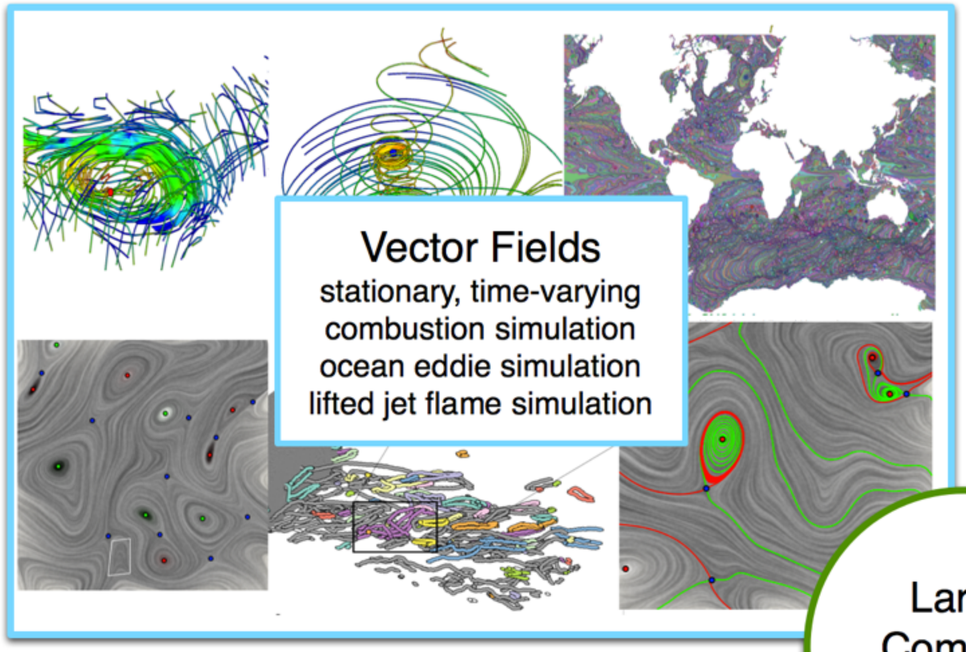
Point Clouds

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

Large Complex Data

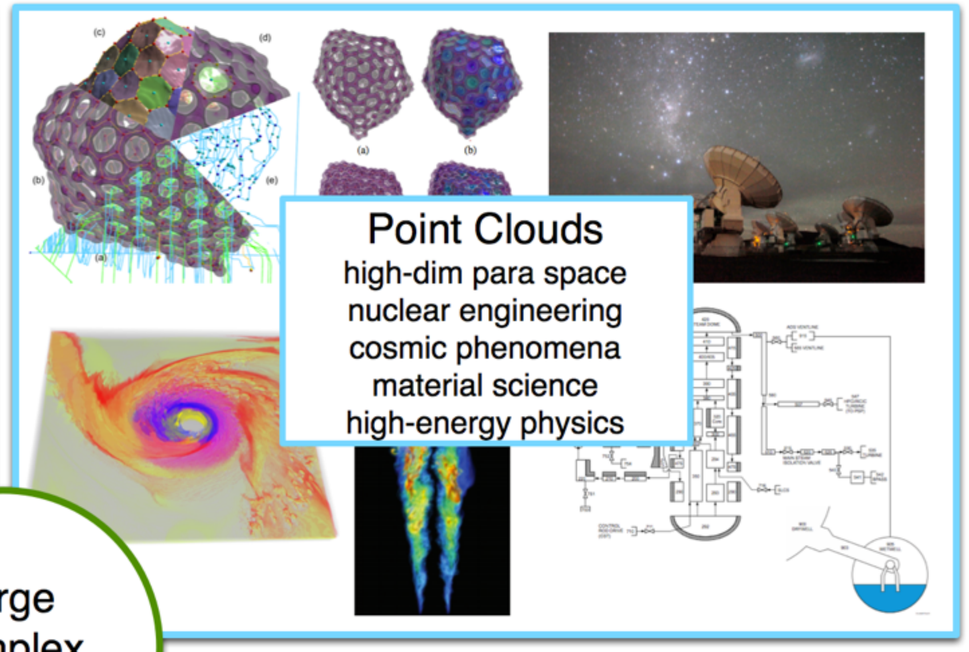






Vector Fields

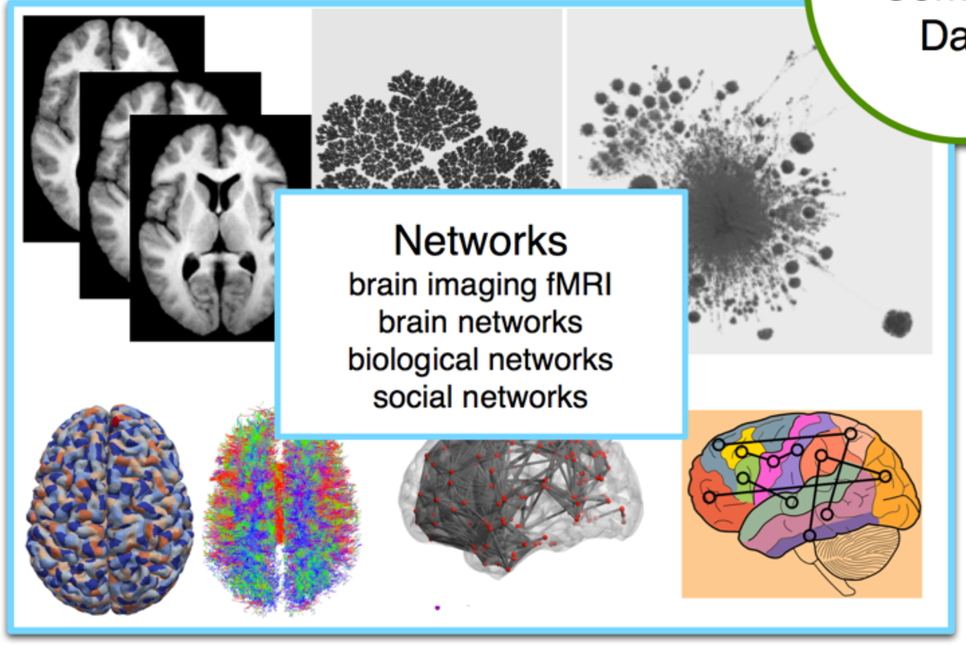
stationary, time-varying
combustion simulation
ocean eddie simulation
lifted jet flame simulation



Point Clouds

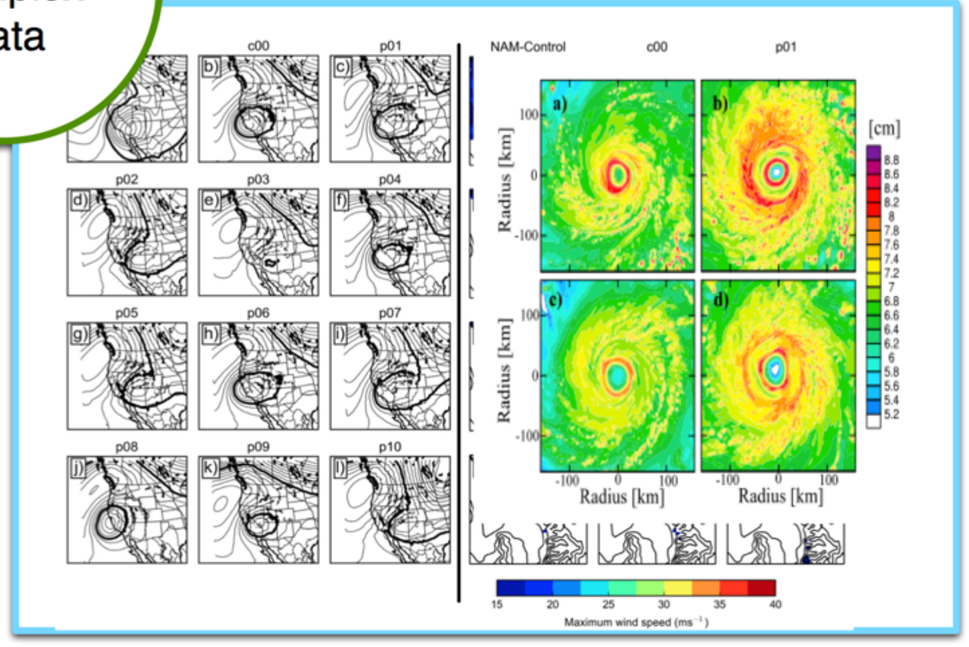
high-dim para space
nuclear engineering
cosmic phenomena
material science
high-energy physics

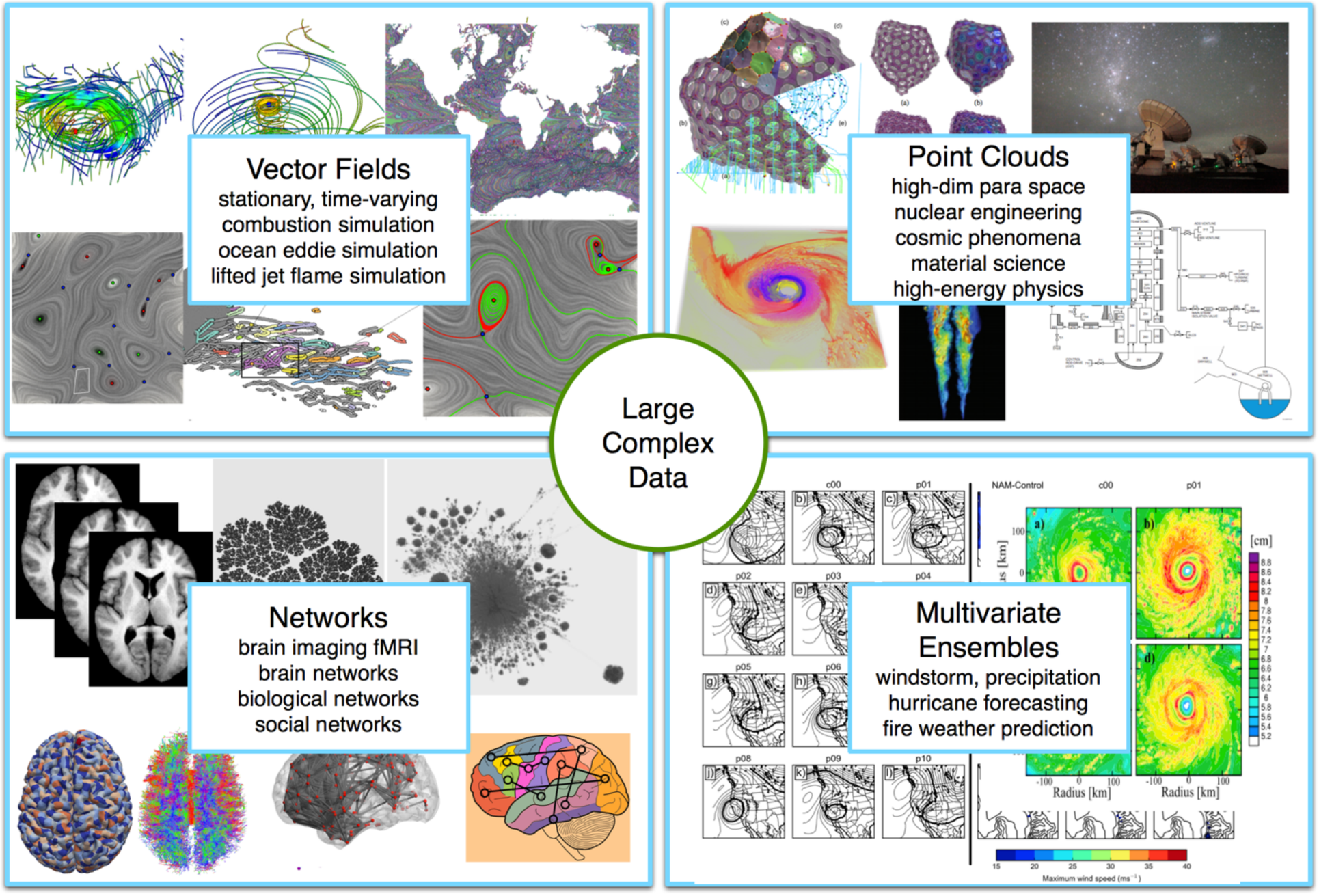
Large Complex Data



Networks

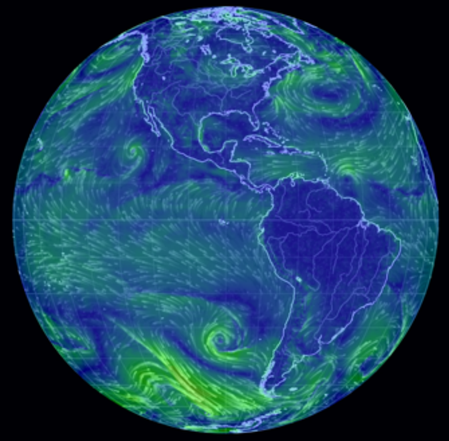
brain imaging fMRI
brain networks
biological networks
social networks

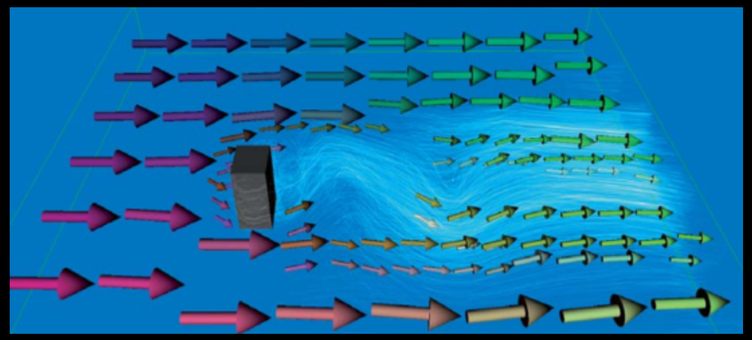


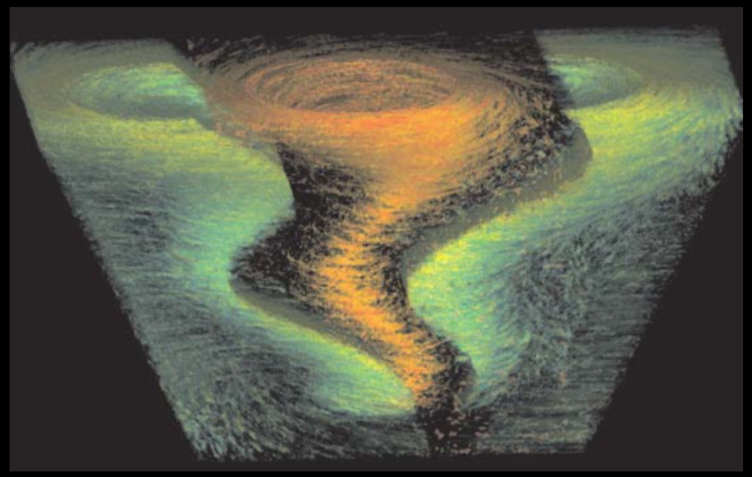

Vector Fields
stationary, time-varying
combustion simulation
ocean eddie simulation
lifted jet flame simulation
Point Clouds
high-dim para space
nuclear engineering
cosmic phenomena
material science
high-energy physics
Large
Complex
Data
Networks
brain imaging fMRI
brain networks
biological networks
social networks
Multivariate
Ensembles
windstorm, precipitation
hurricane forecasting
fire weather prediction
NAM-Control
c00
p01
p02
p03
p04
p05
p06
p08
p09
p10
Radius [km]
[cm]
Maximum wind speed (ms⁻¹)

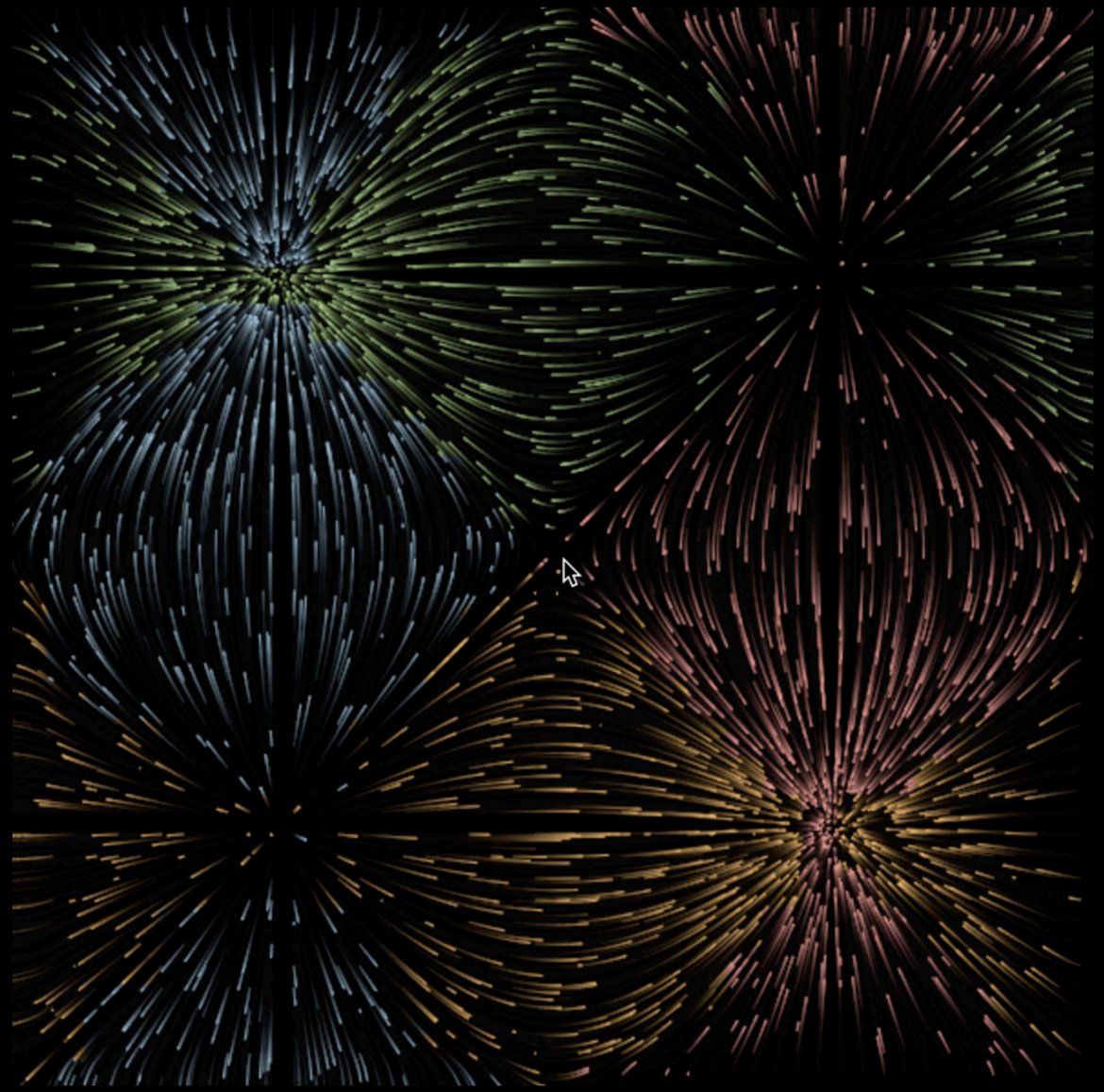
Vector Fields
Combustion and Ocean

Make the flow patterns visible & Interpretable



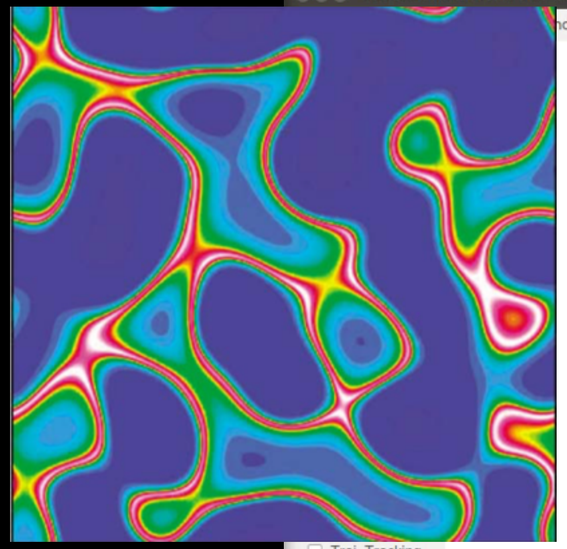


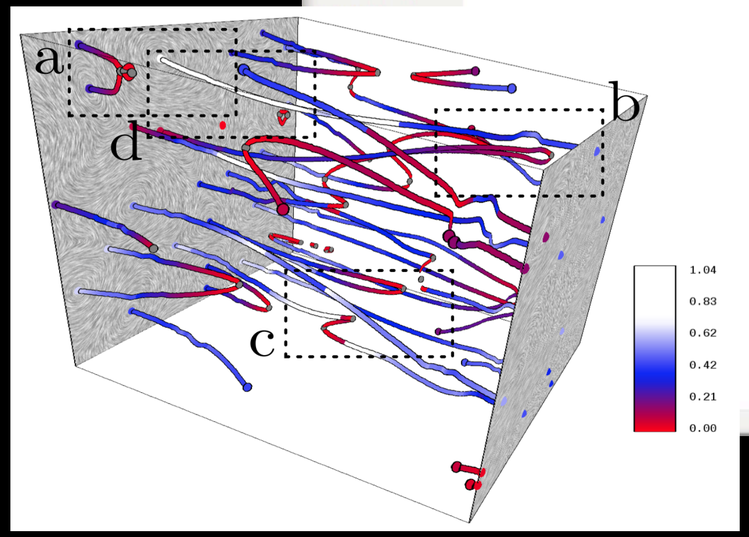




Sources: Dan Maljovec, Cameron Beccario, [Correa, Silver, Chen 2007]
[Burger, Kondratieva, Kruger, Westermann 2008]

Quantify feature stability

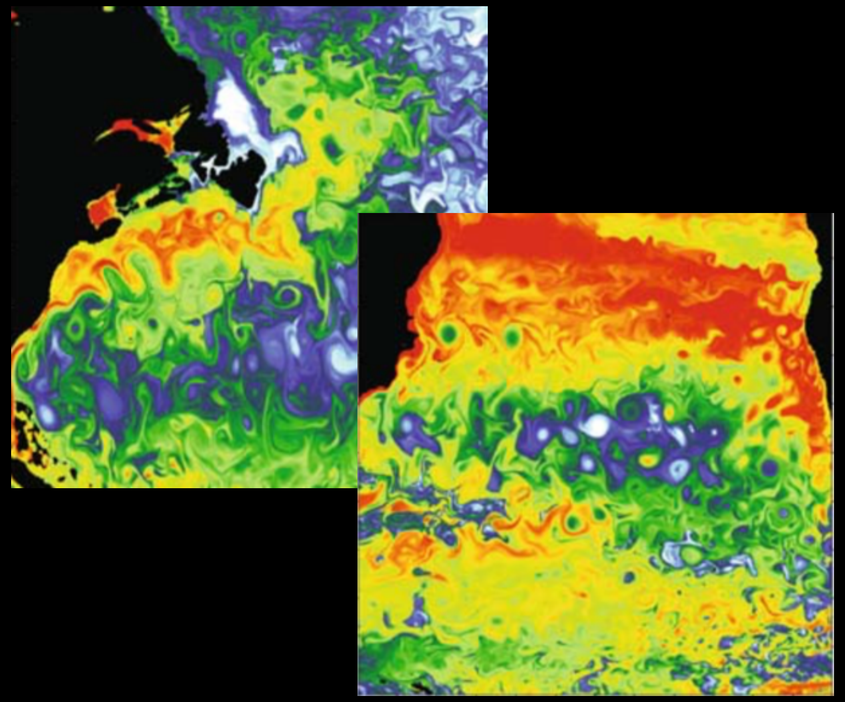


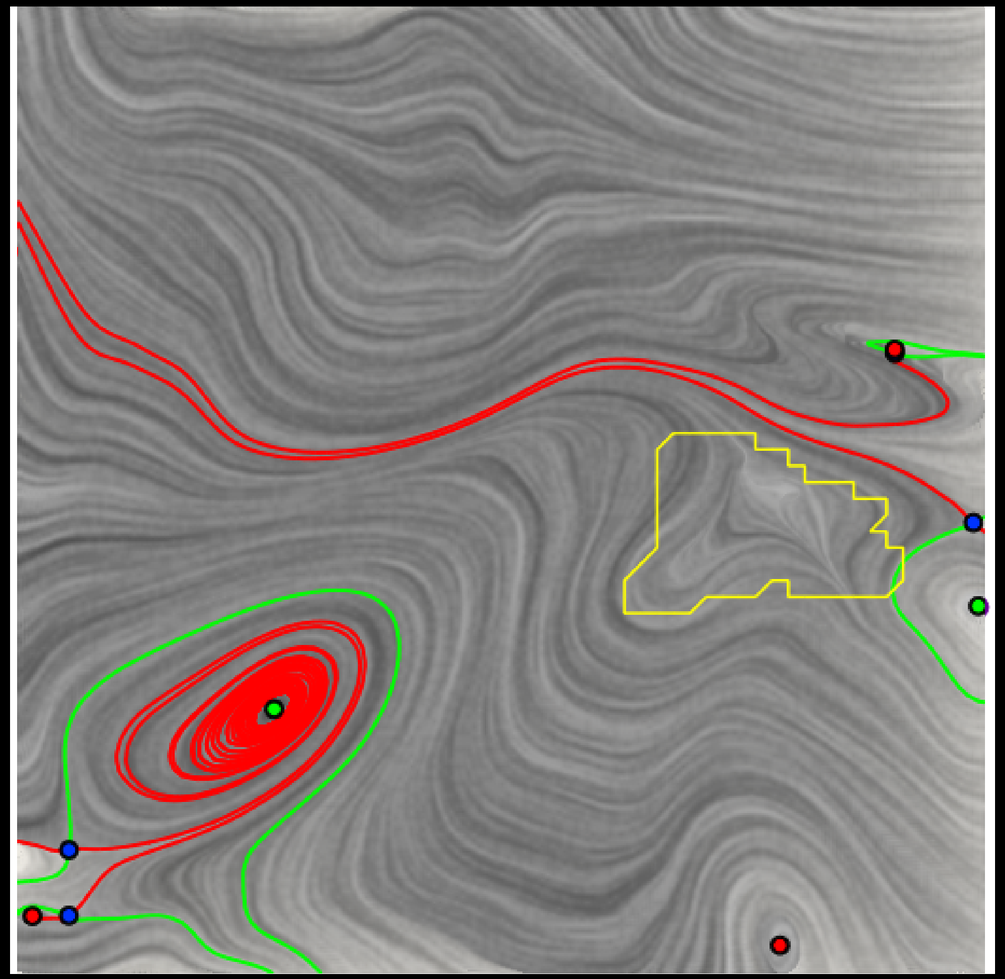


Combustion Simulation

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

Separate features from noise at multi-scale

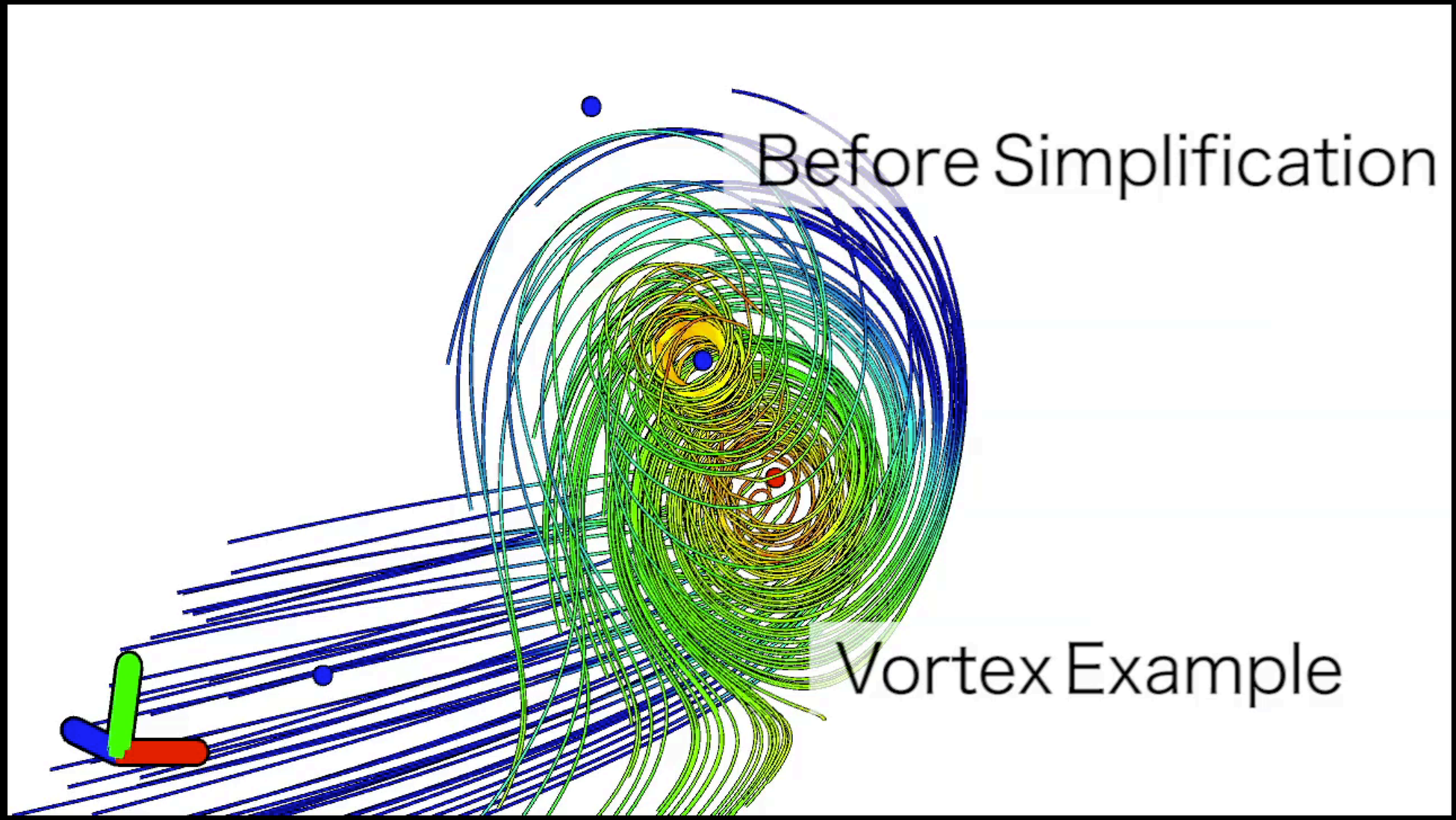




Ocean Eddy Simulation

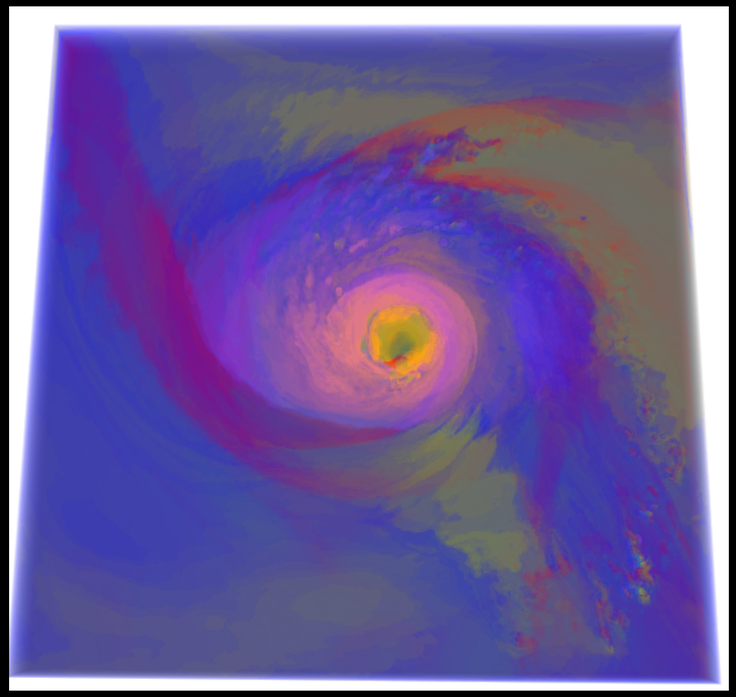
Map: Courtesy of SlidesCarnival & Unsplash
Simulation: [Maltrud, Bryan, Peacock 2010]

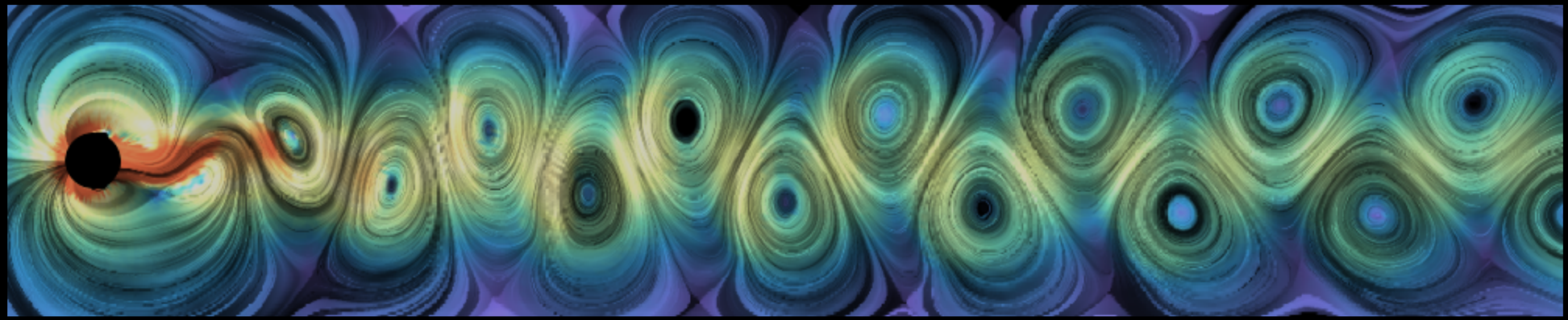
Visualize flow in 3D



Understand turbulent flow

Source: NASA

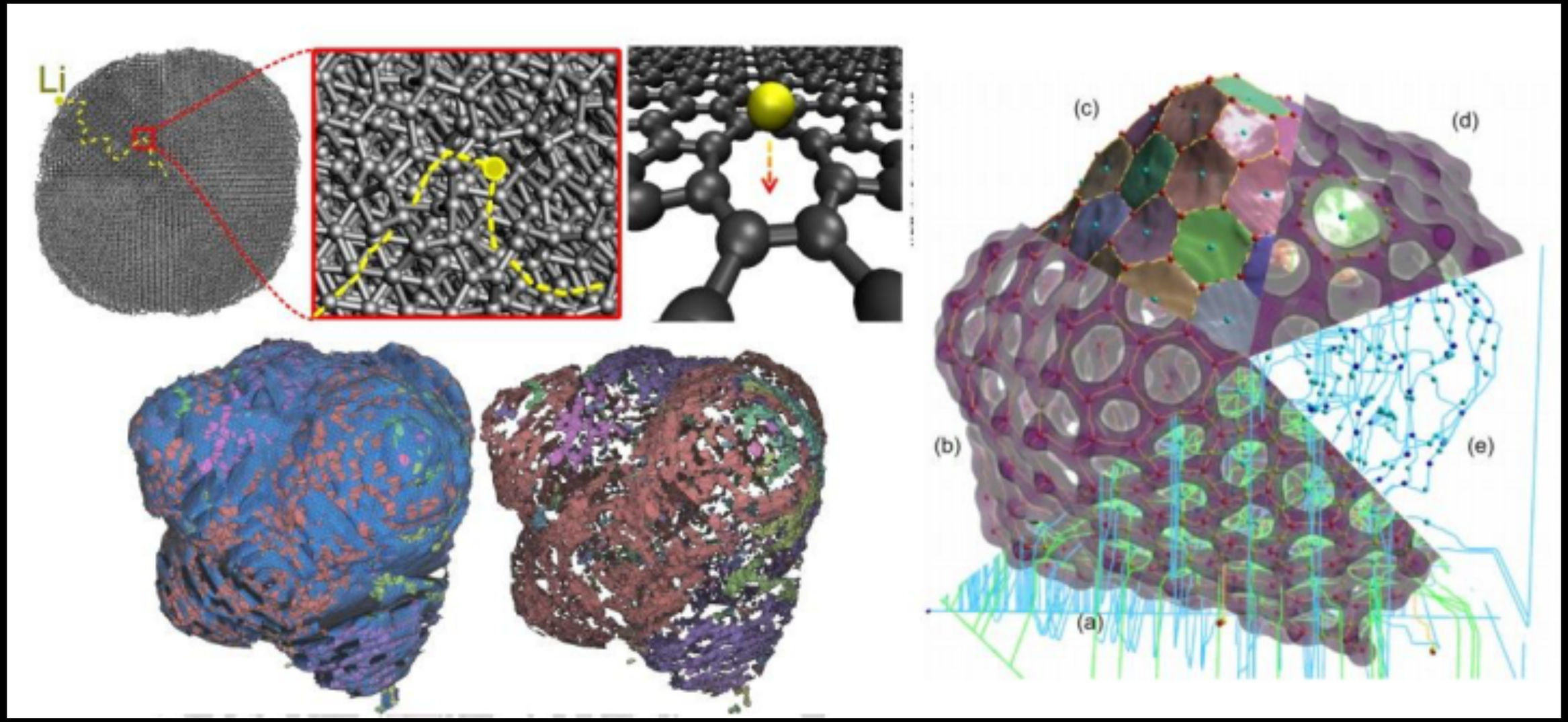




Material Science
Your iPhone Battery

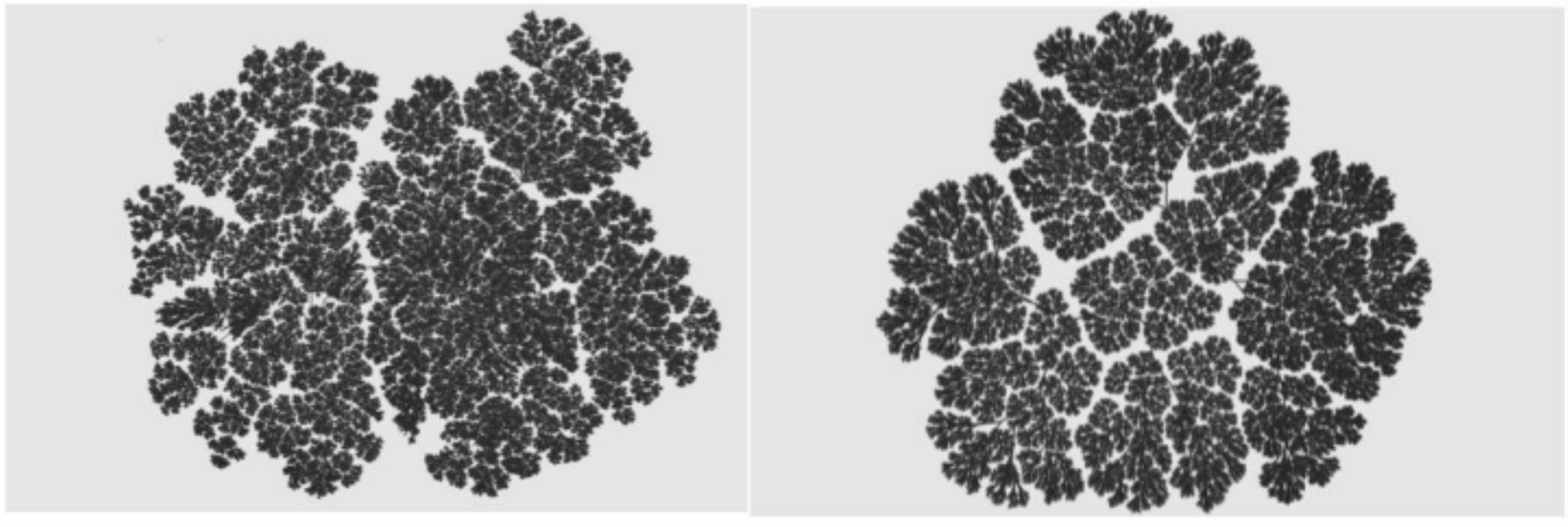
How long can
your battery last?

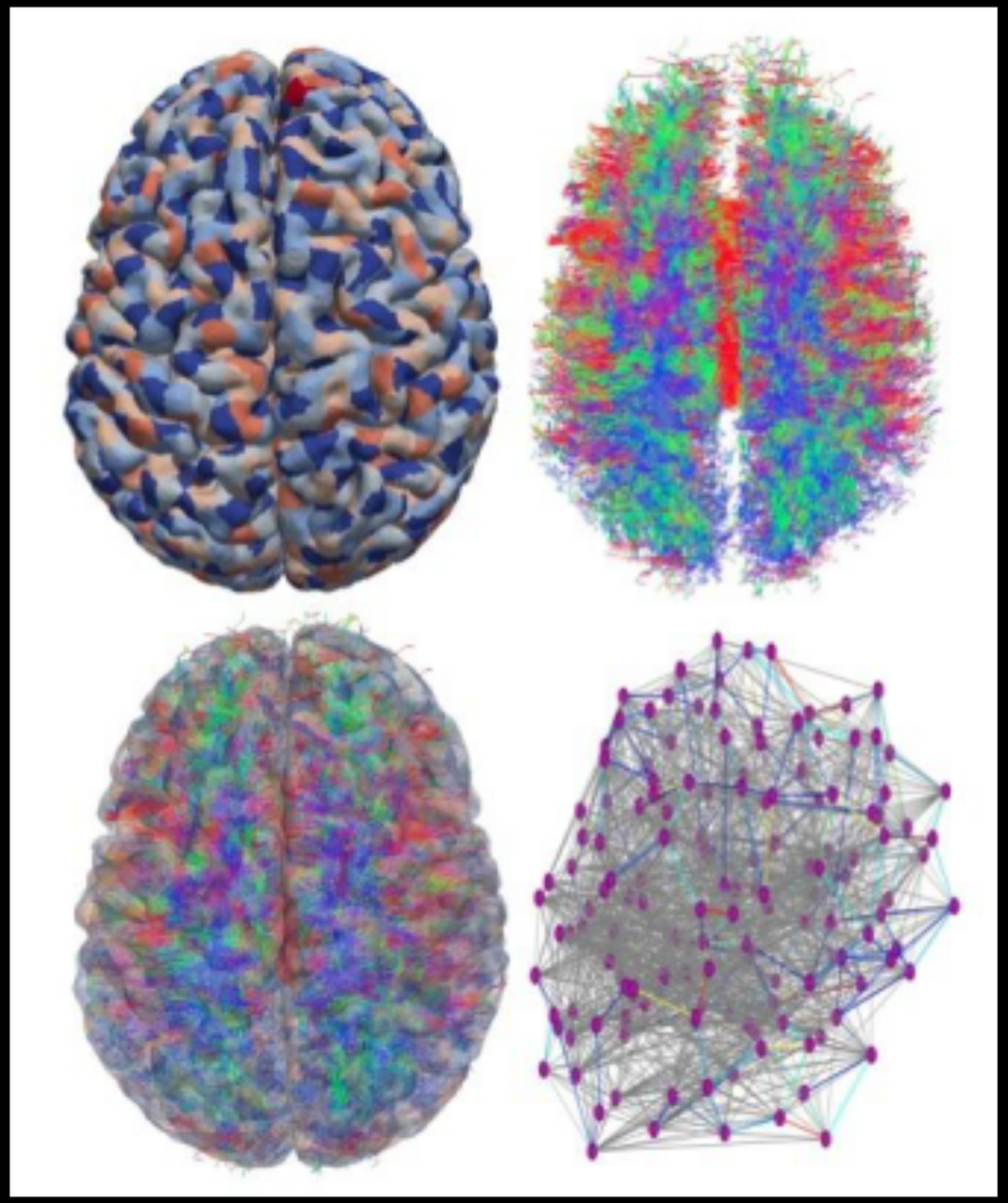
Ion diffusion geometry extraction in battery



Networks
Brain networks

Inadequate Network Visualization

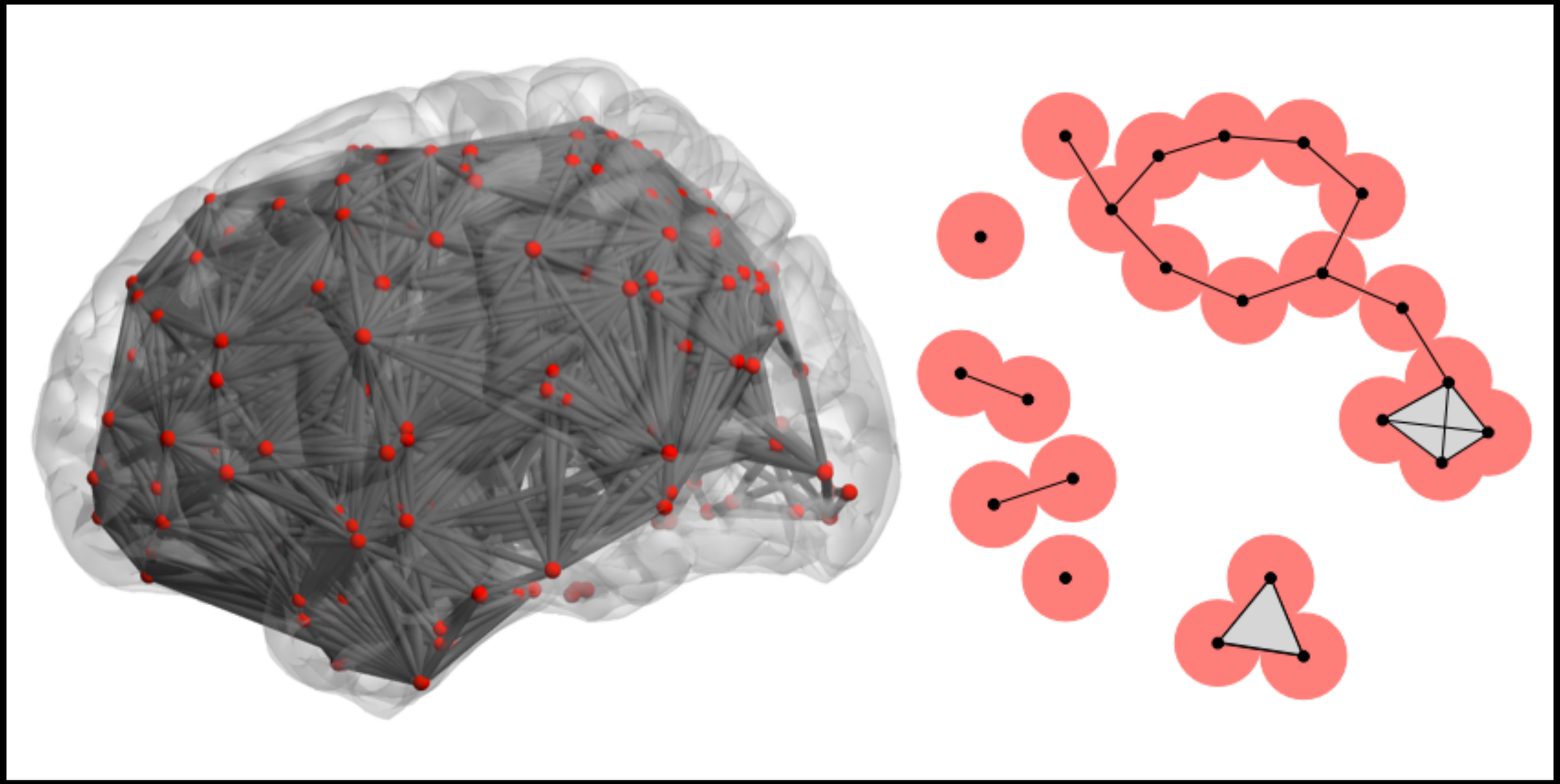




Brain Network Visualization

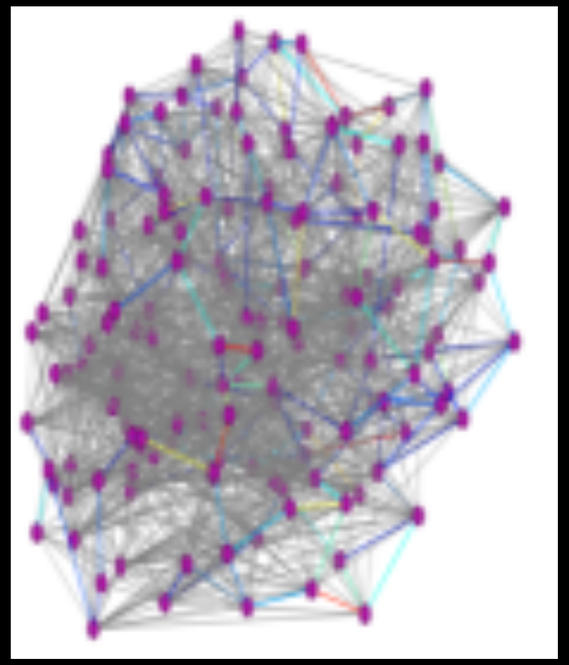
Avoid network hairballs
while preserving structure?

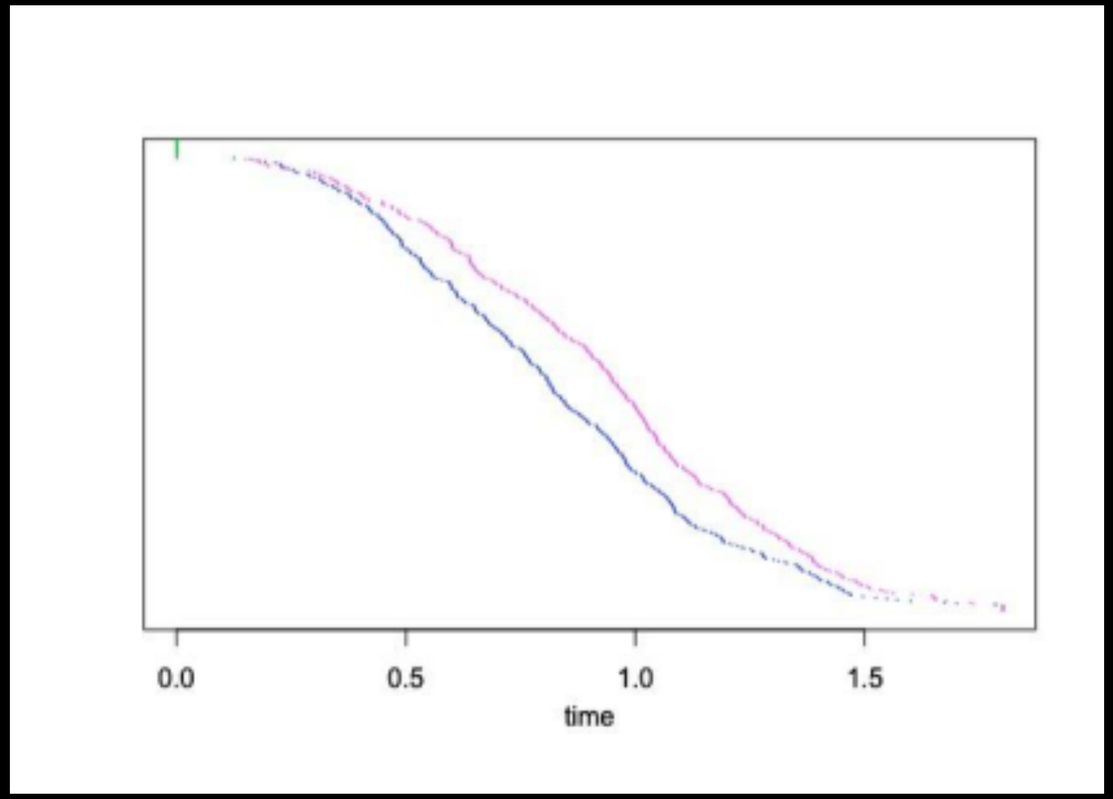
Topology and brain networks

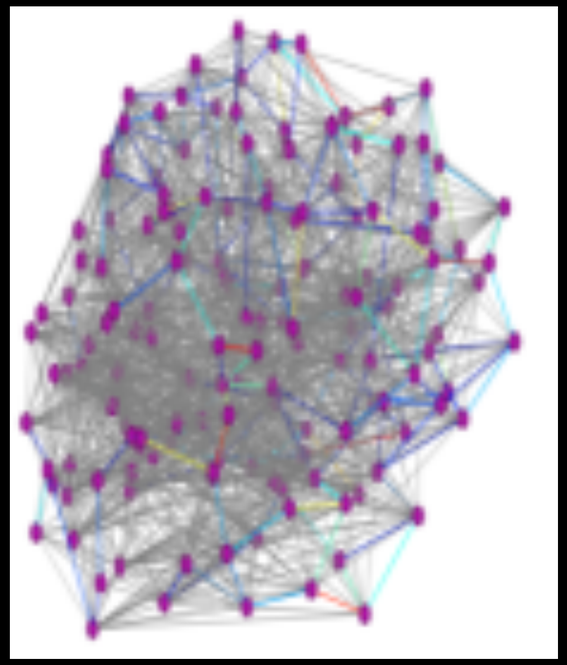


Autism Brain Networks

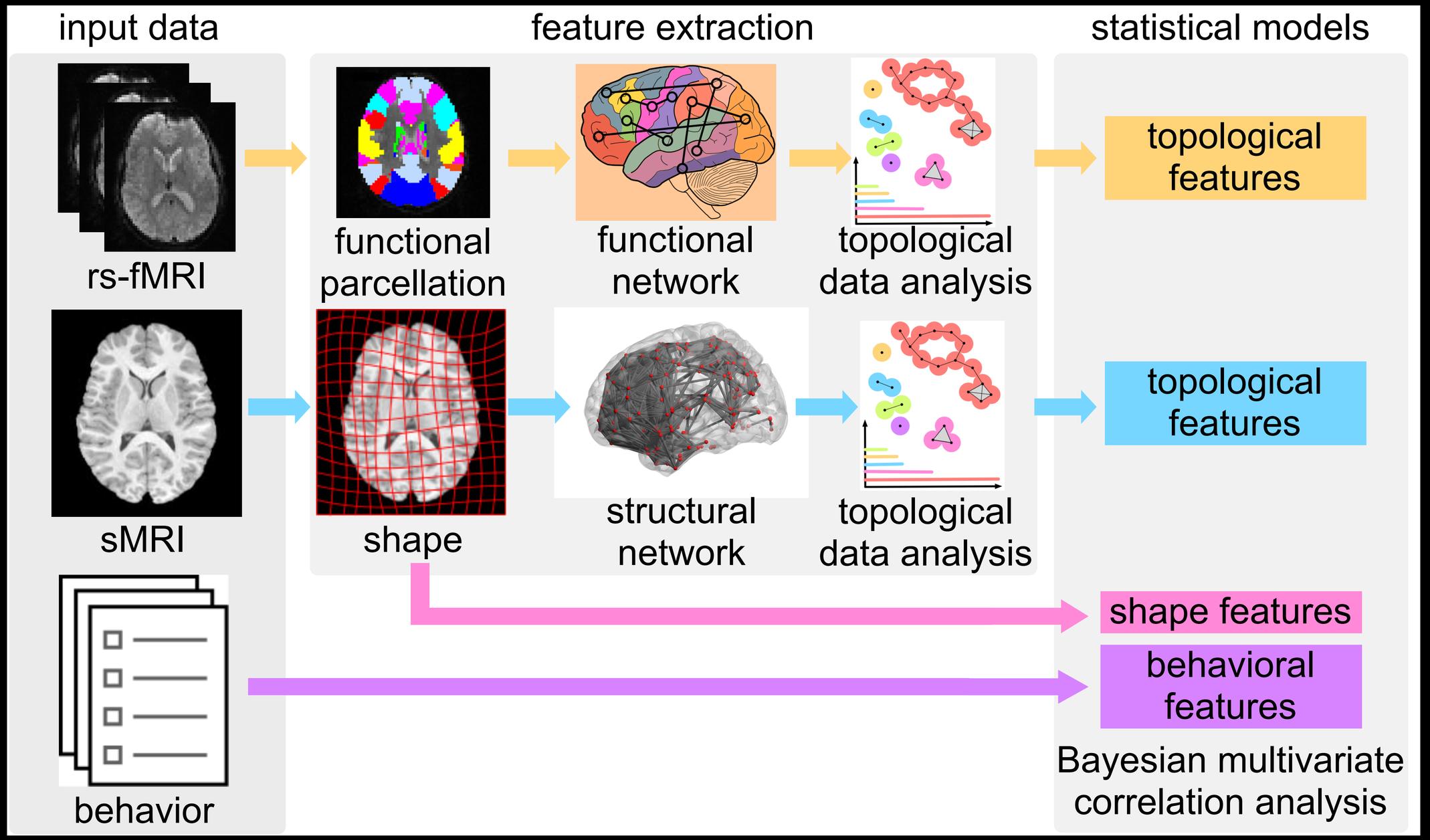
Can we tell autism subject from control?







Autism Brain Networks

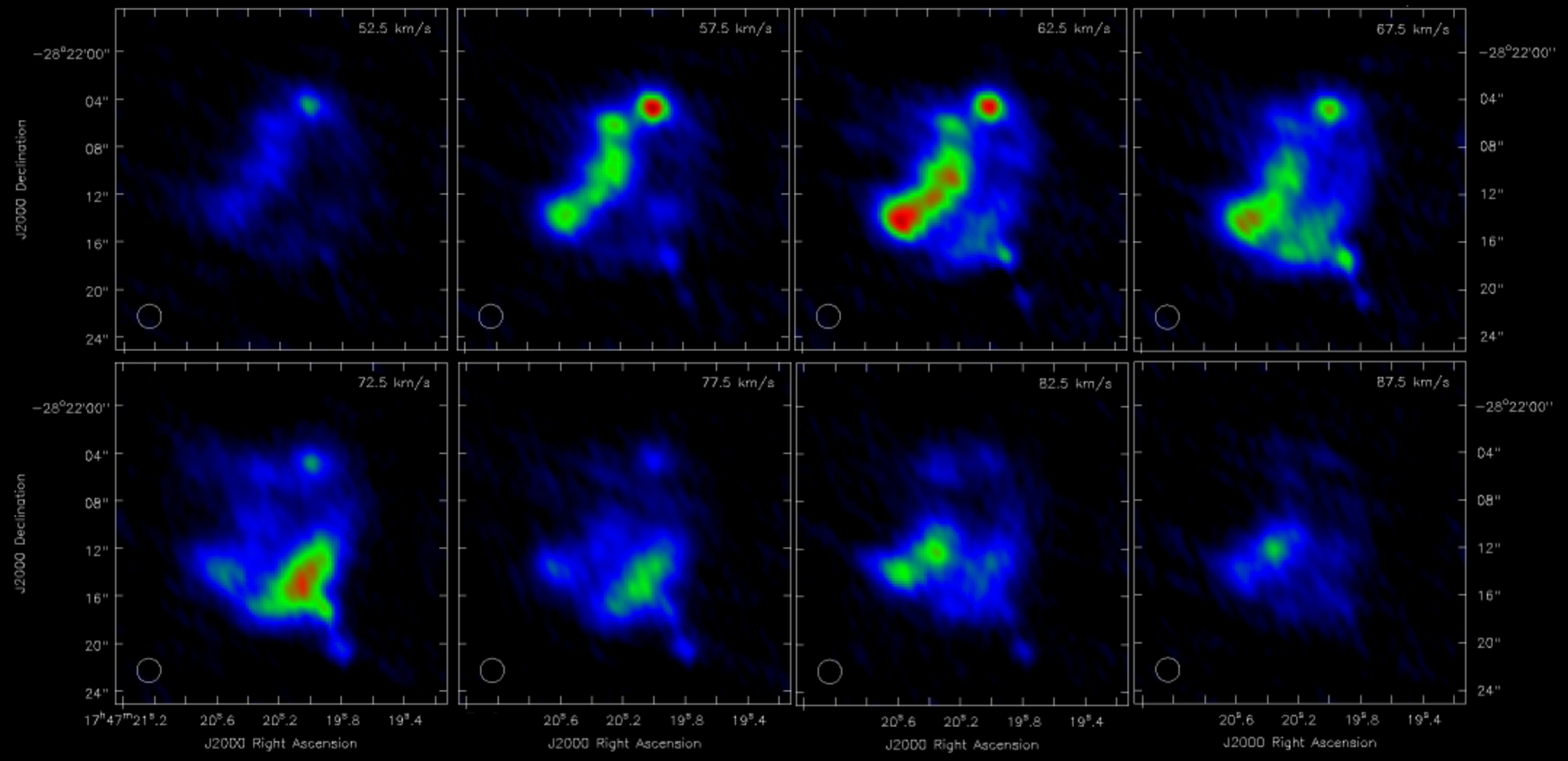


Astronomy
Telescopes and Black Holes

Largest radio telescopes in the world

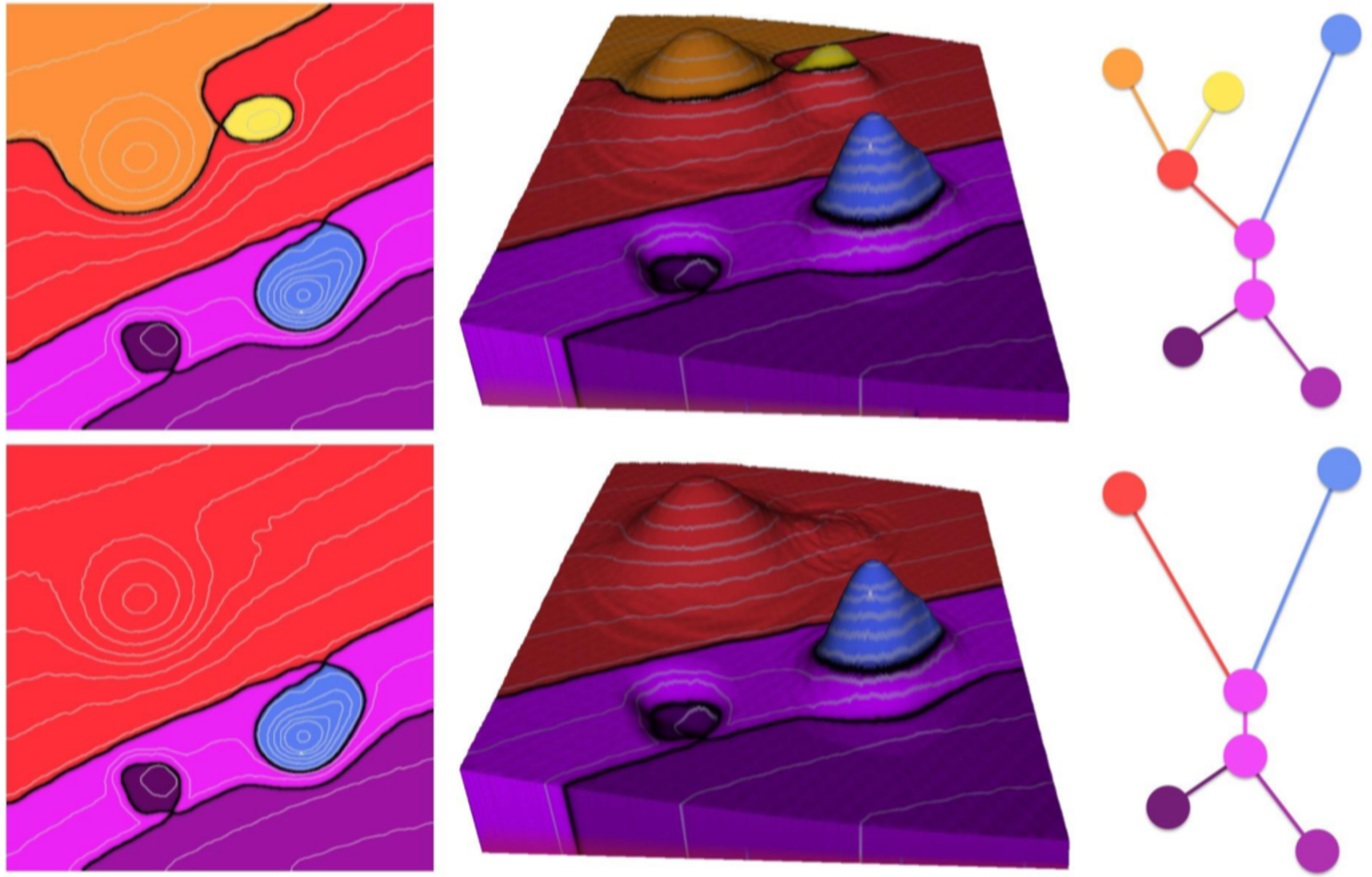
Credit: ALMA

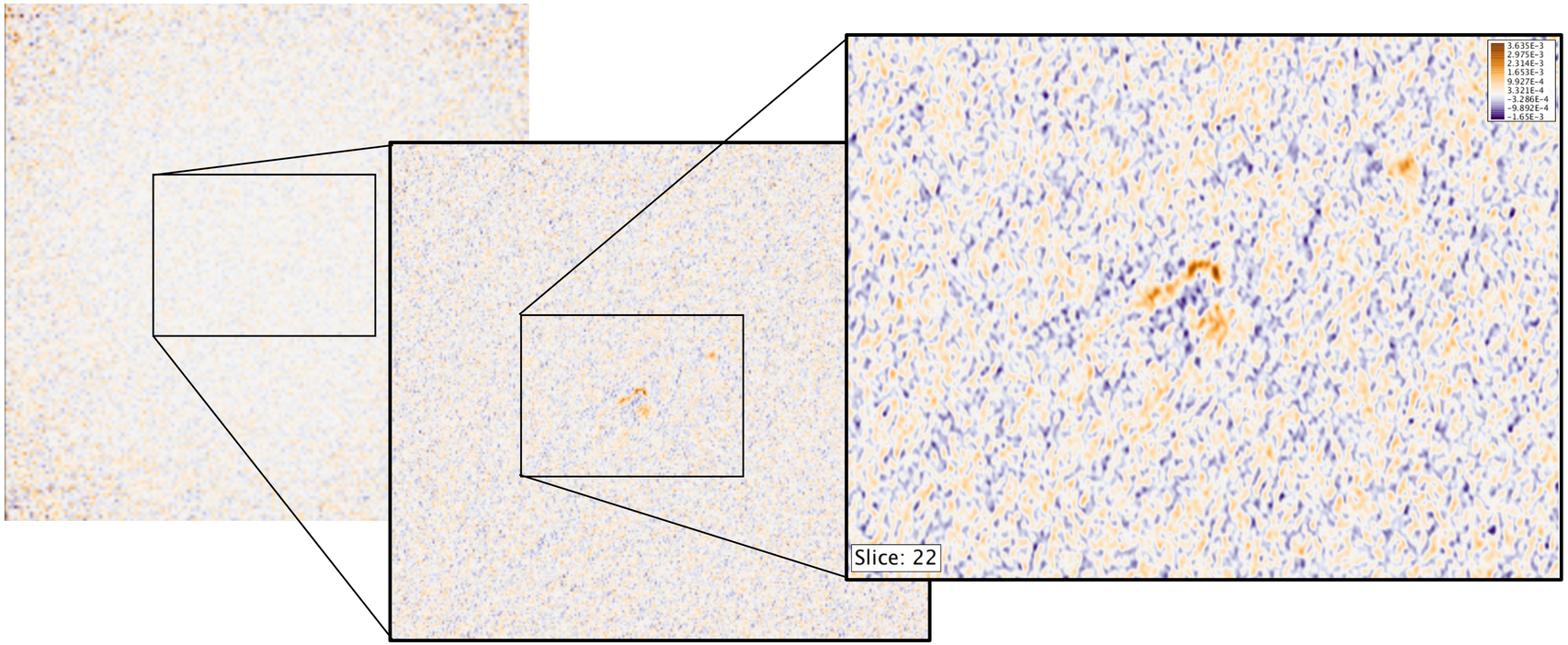
Radio telescope Data



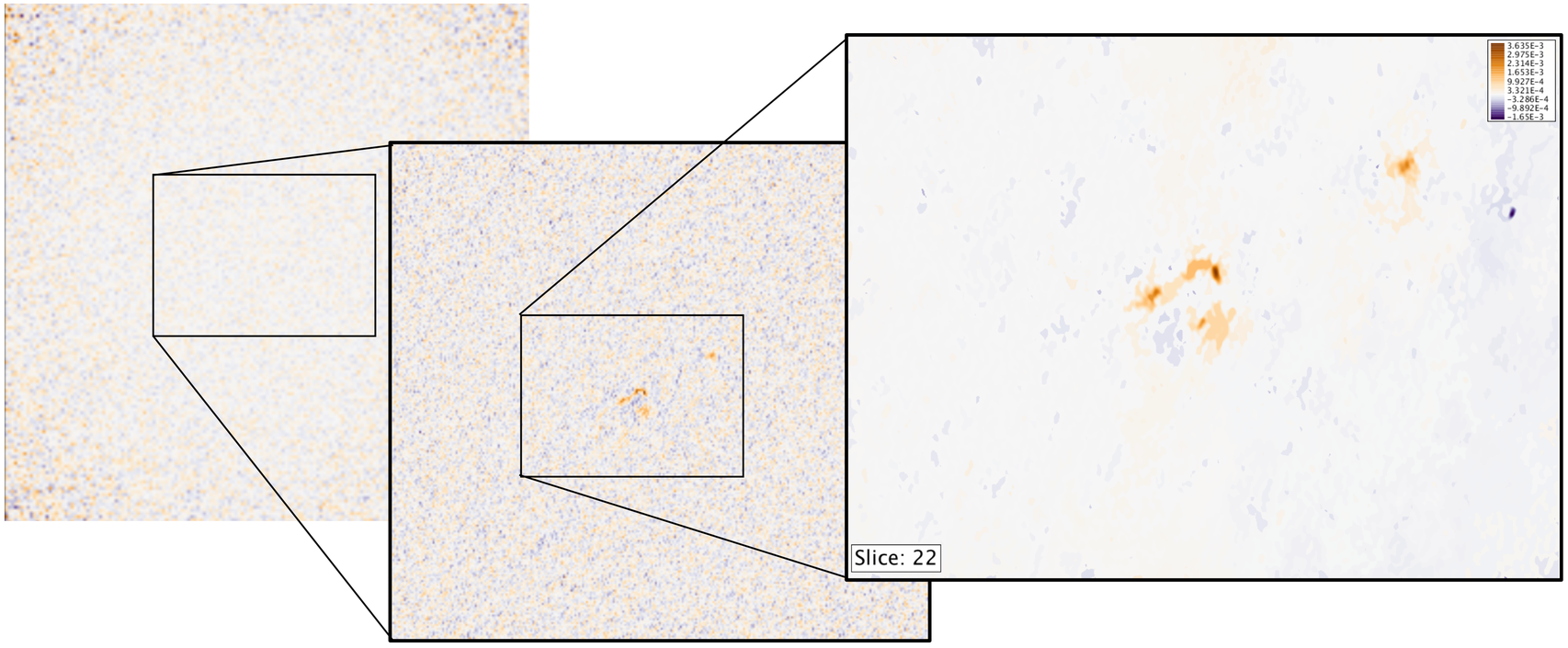
NGC 404: Mirach's Ghost Galaxy

Feature Denoting and Source Finding

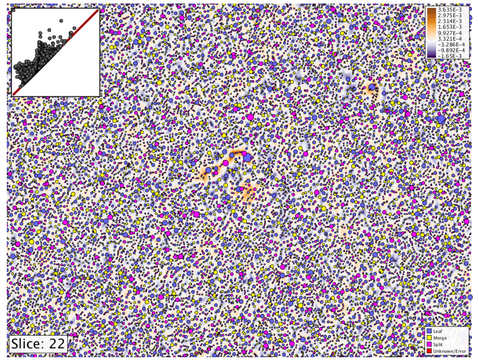


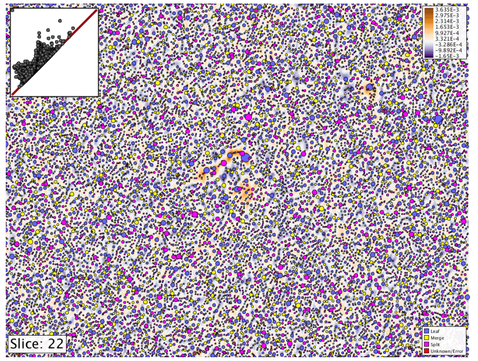


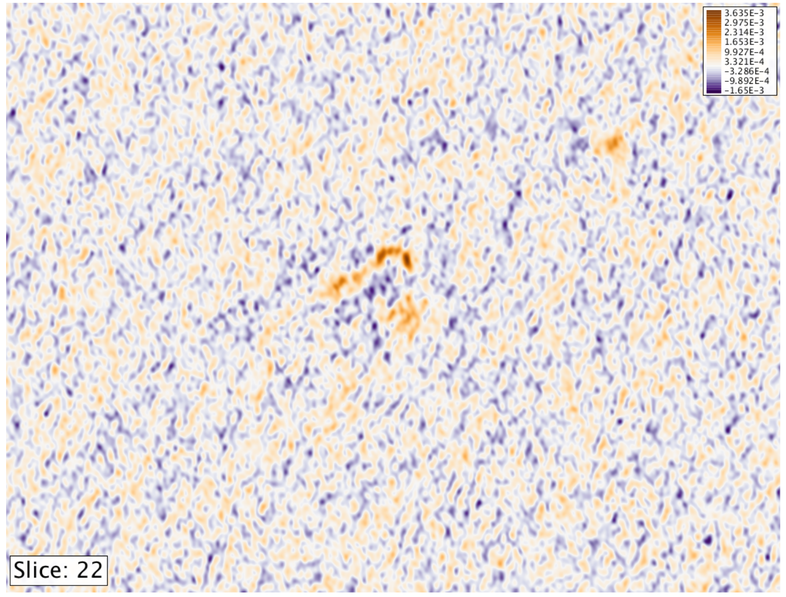
Paul Rosen, Bei Wang, Anil Seth, Betsy Mills, Adam Ginsburg, Julia Kamenetzky, Jeff Kern, Chris R. Johnson. Manuscript, 2017.

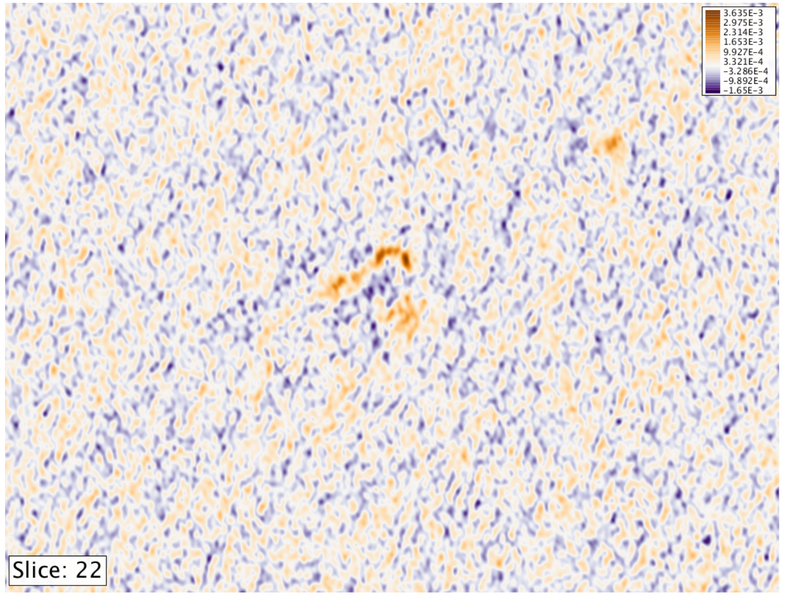

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22

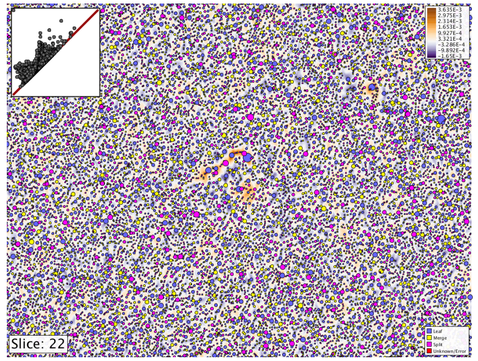
Denoising at Multi-scale and Source Finding

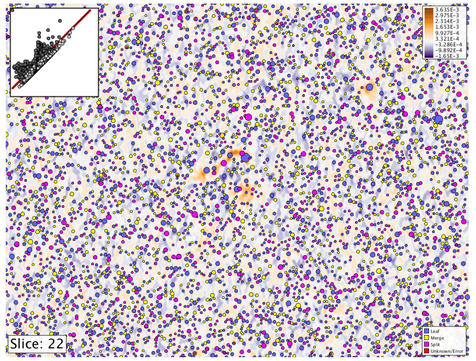

Slice: 22

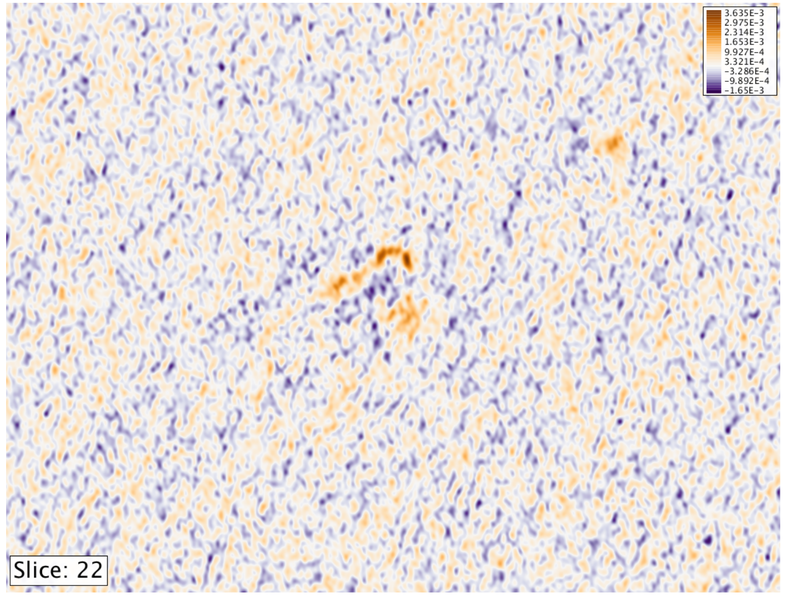

Slice: 22

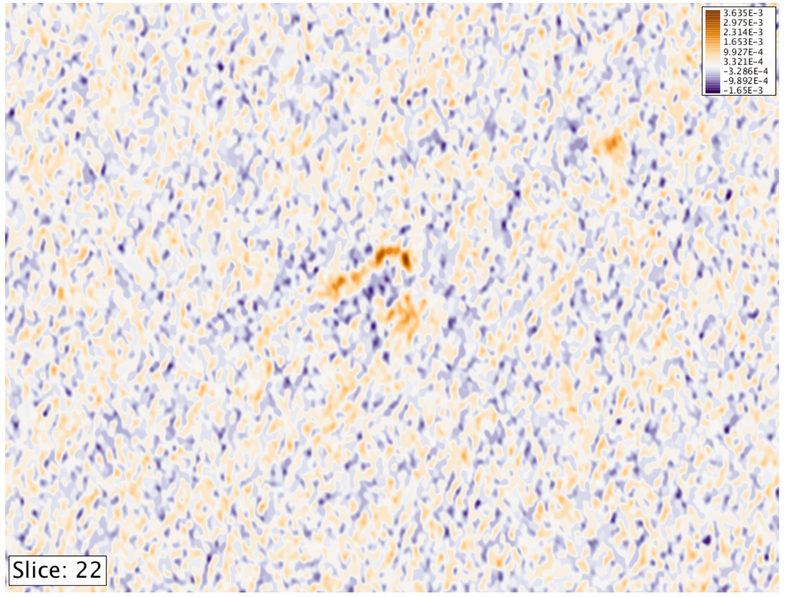

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22

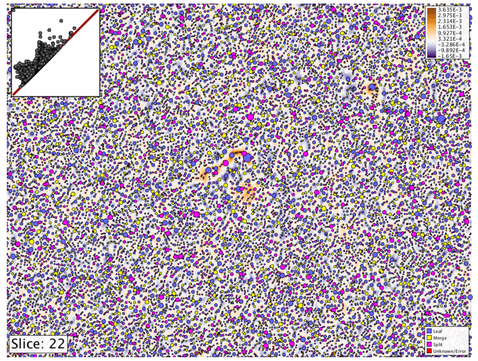

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22

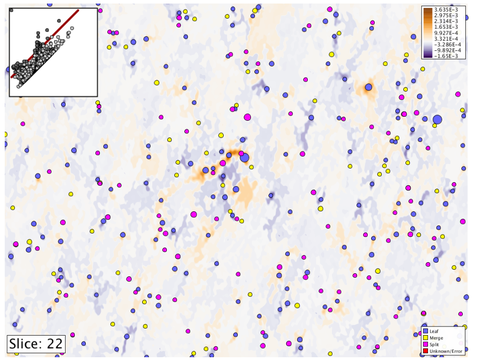

Slice: 22

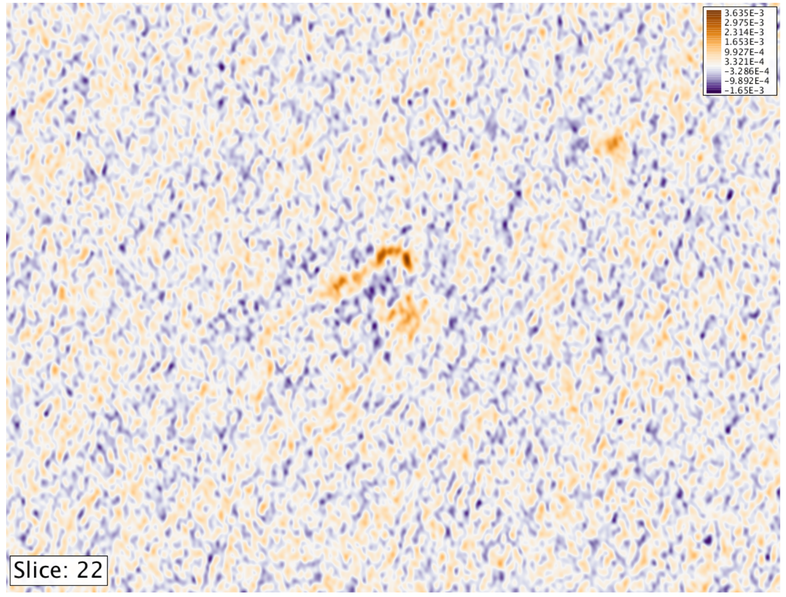

Slice: 22

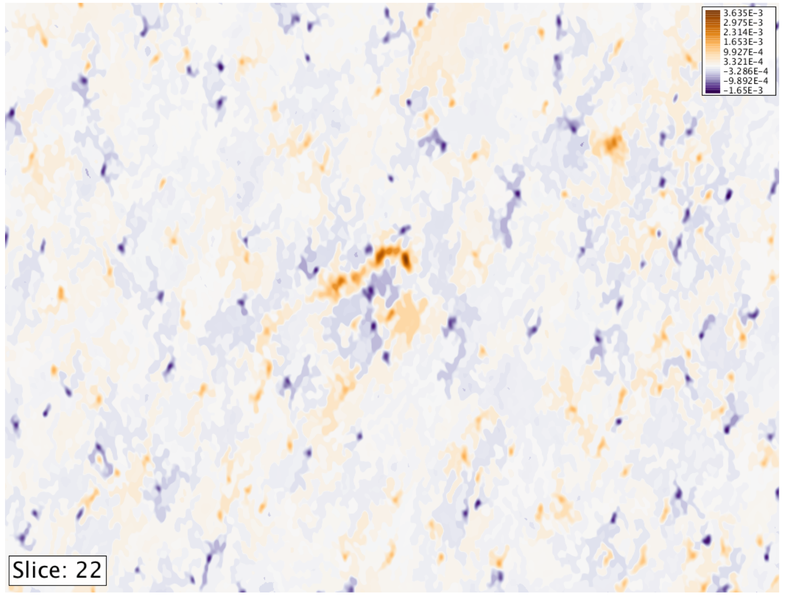

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22

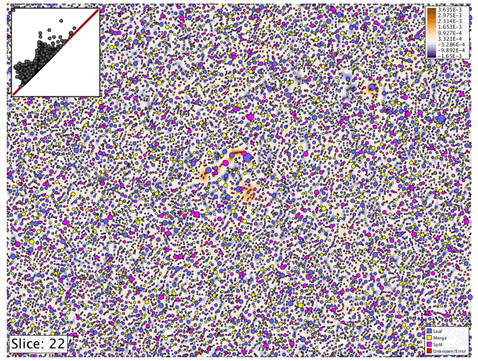

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22

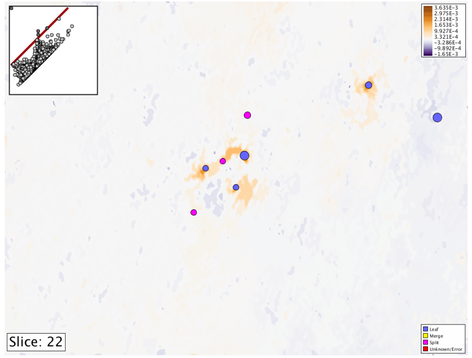

Slice: 22

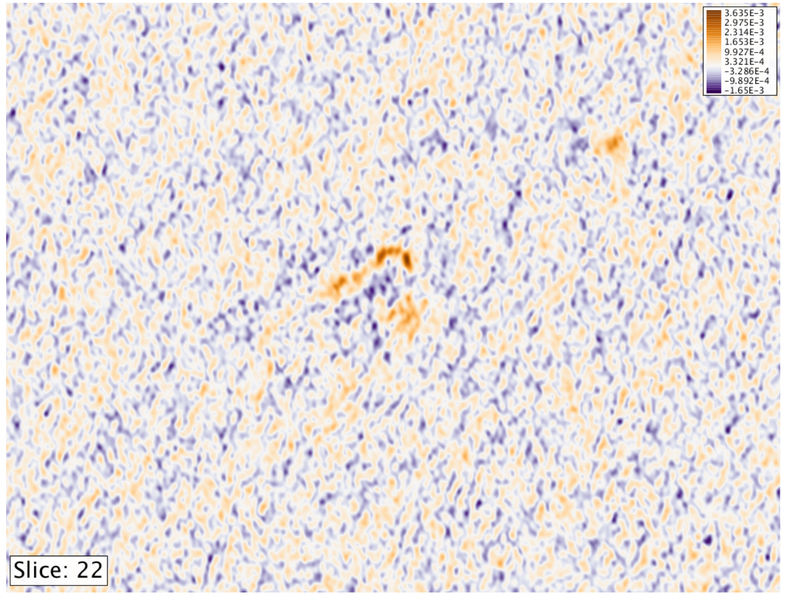

Slice: 22

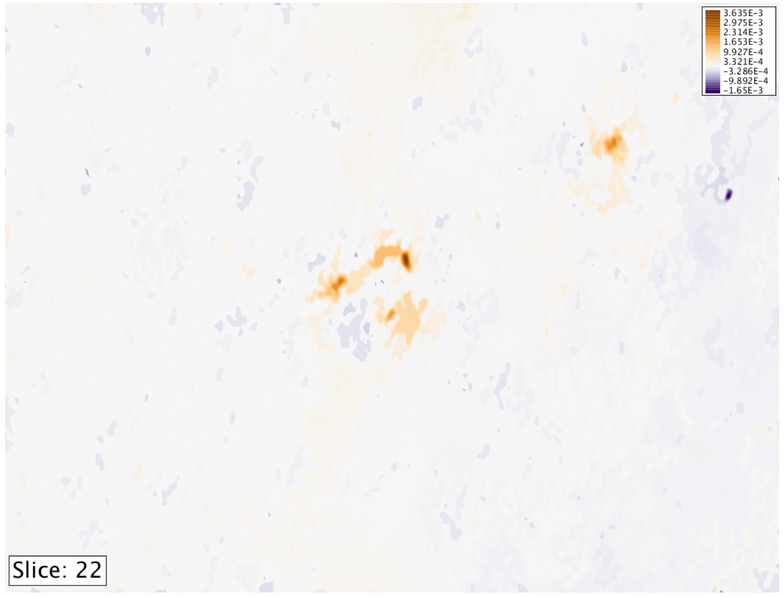

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22

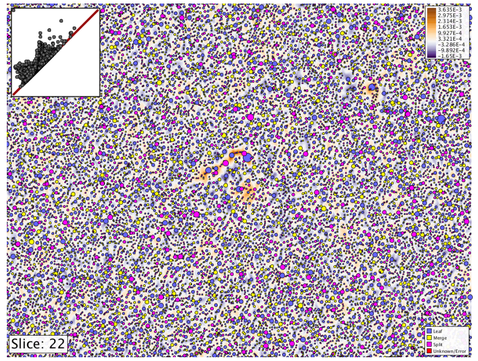

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22

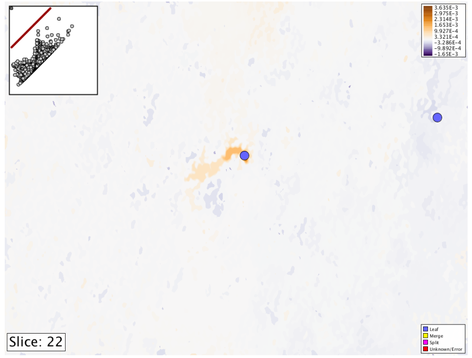

Slice: 22

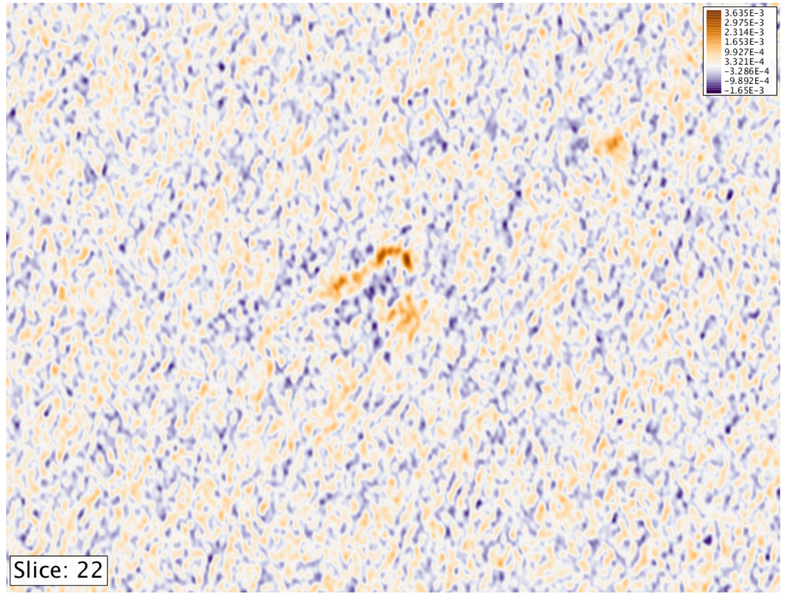

Slice: 22


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


Slice: 22

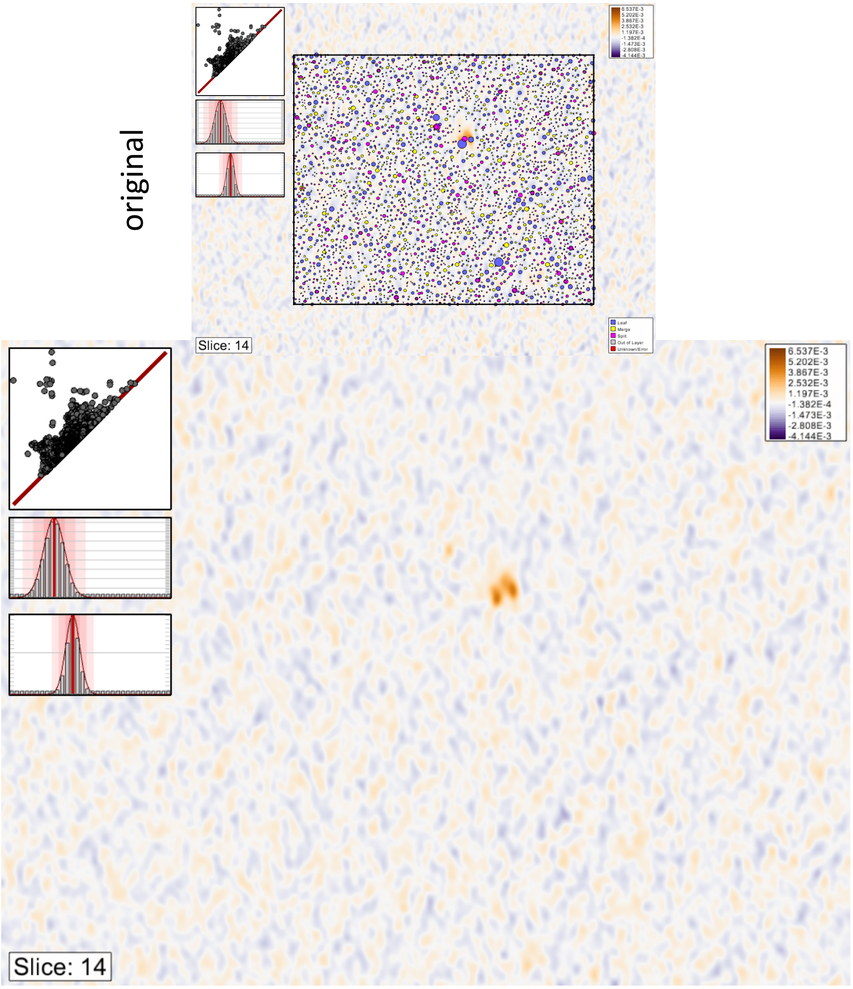

Slice: 22


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

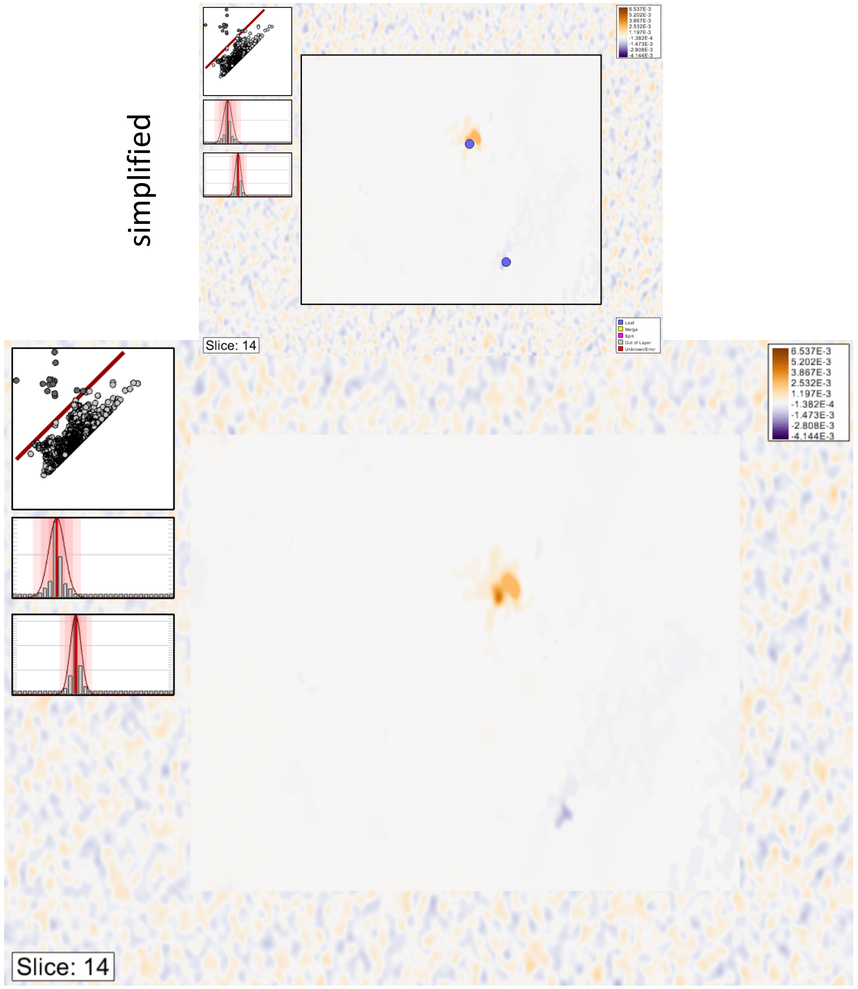

3.635E-3
2.975E-3
2.314E-3
1.653E-3
9.927E-4
3.321E-4
-3.286E-4
-9.892E-4
-1.65E-3
Slice: 22

Stepping Through Slices

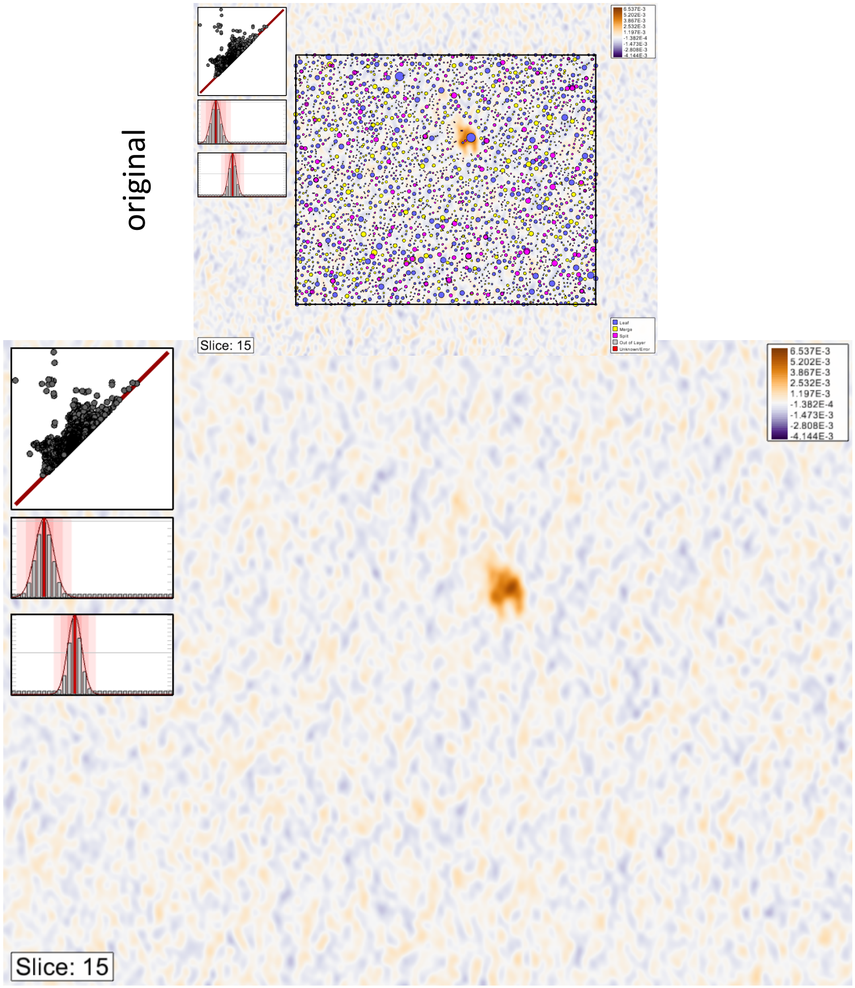
original



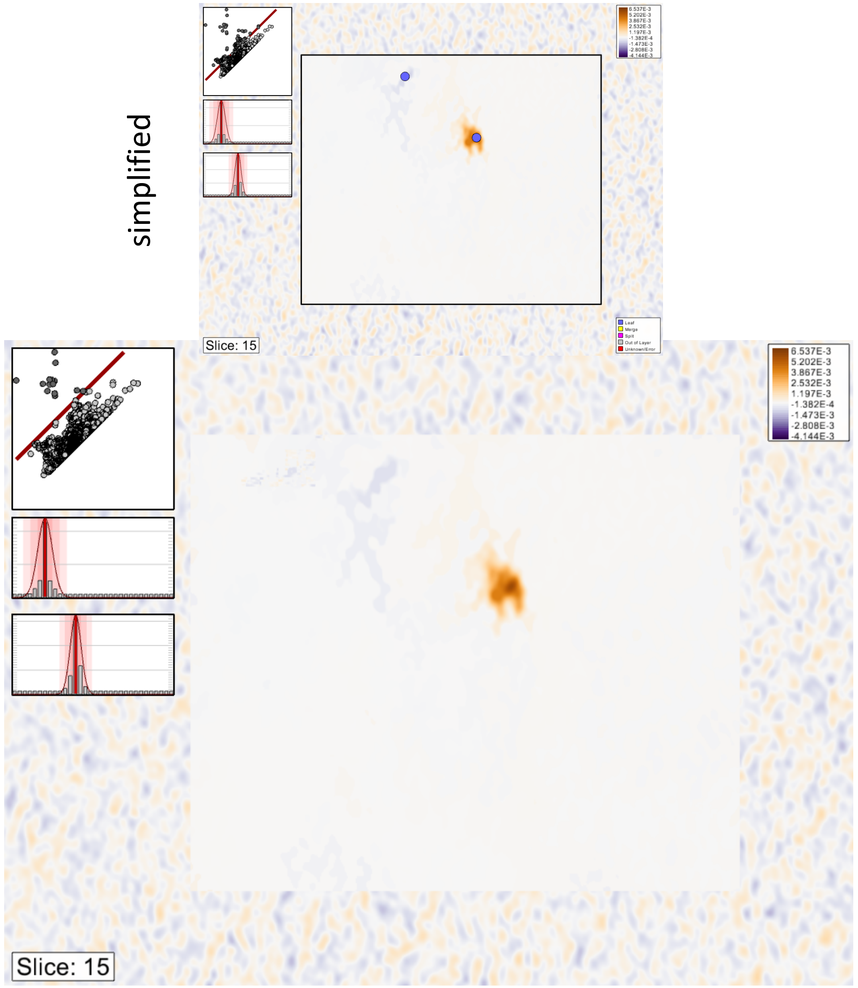
simplified



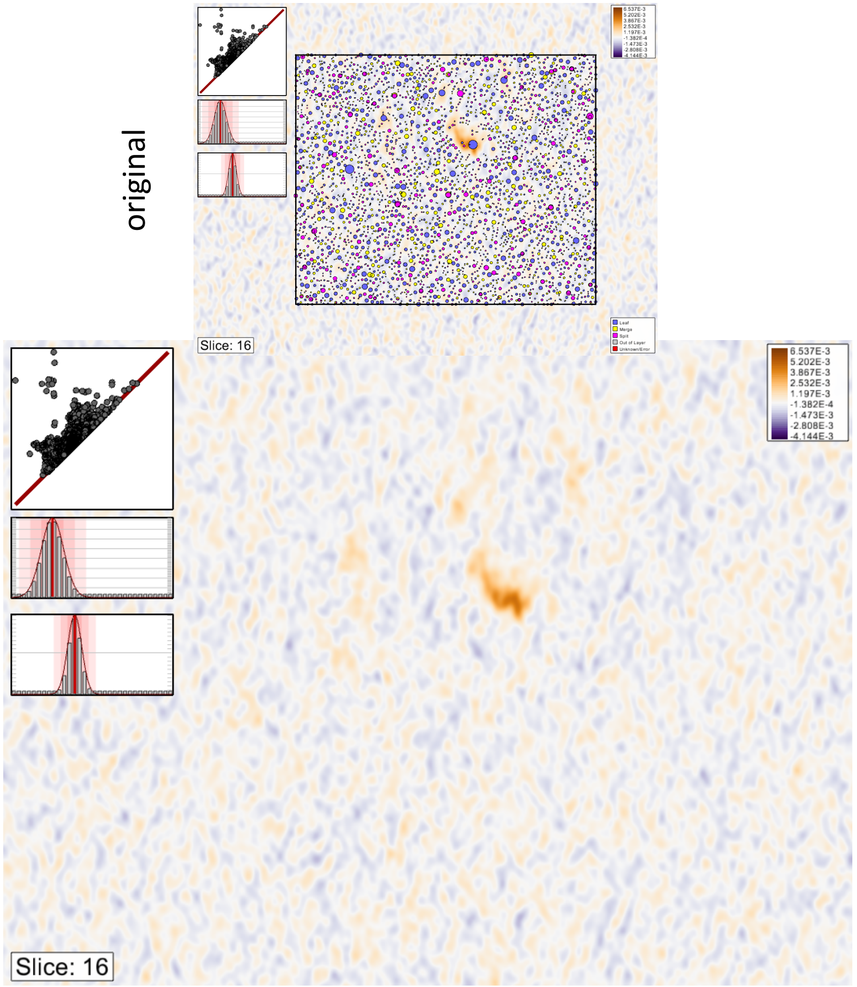
original



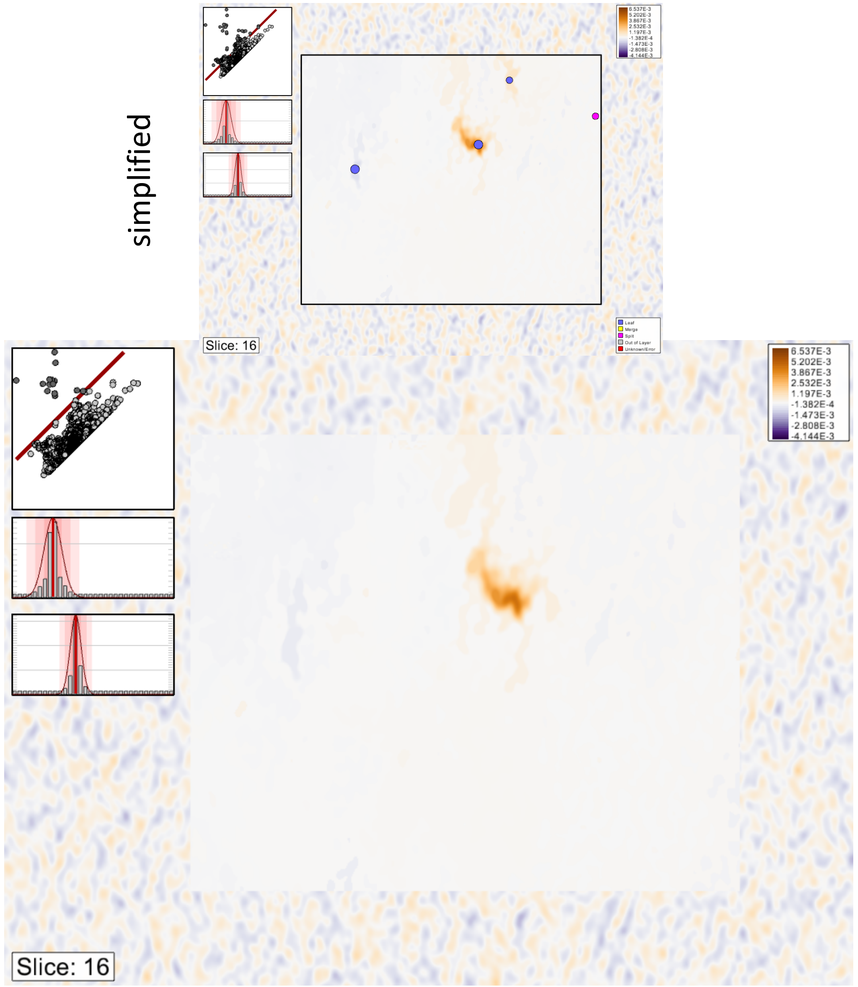
simplified



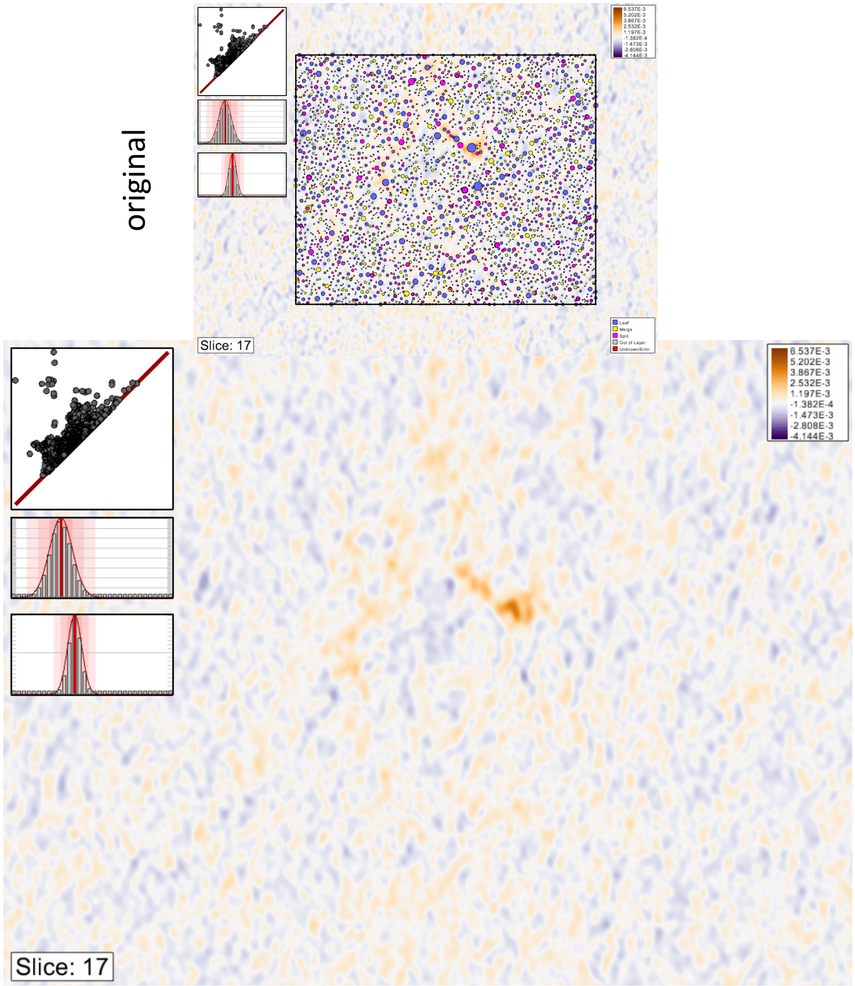
original



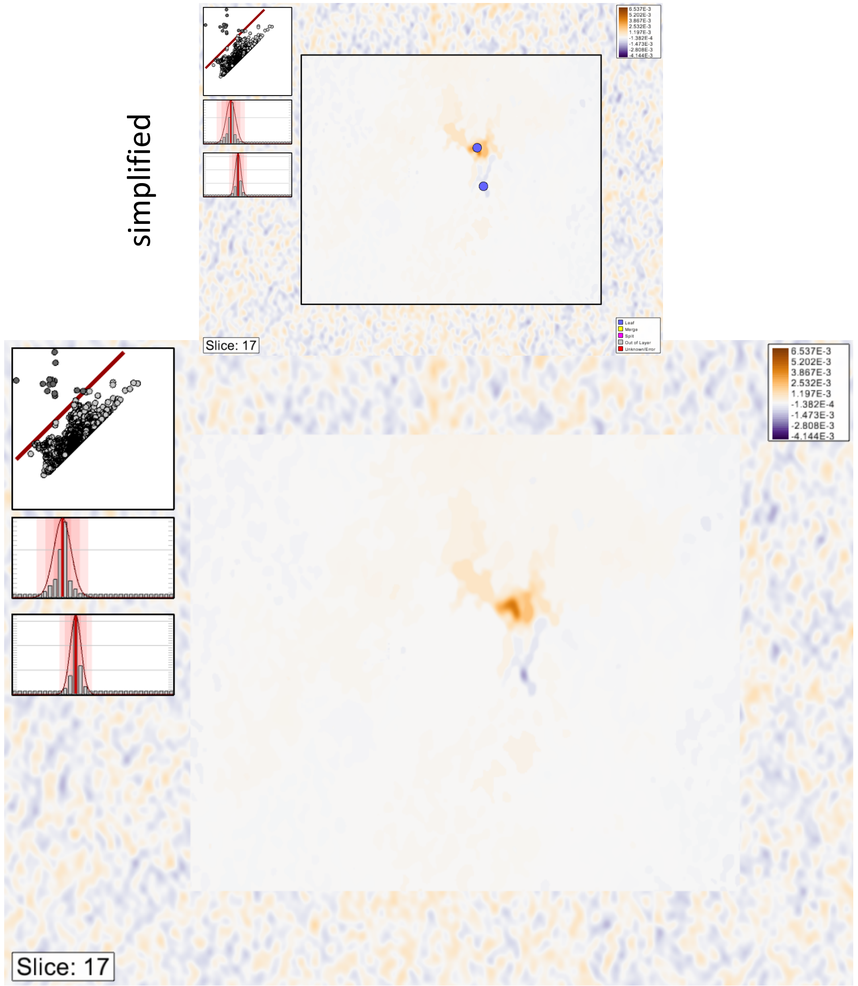
simplified



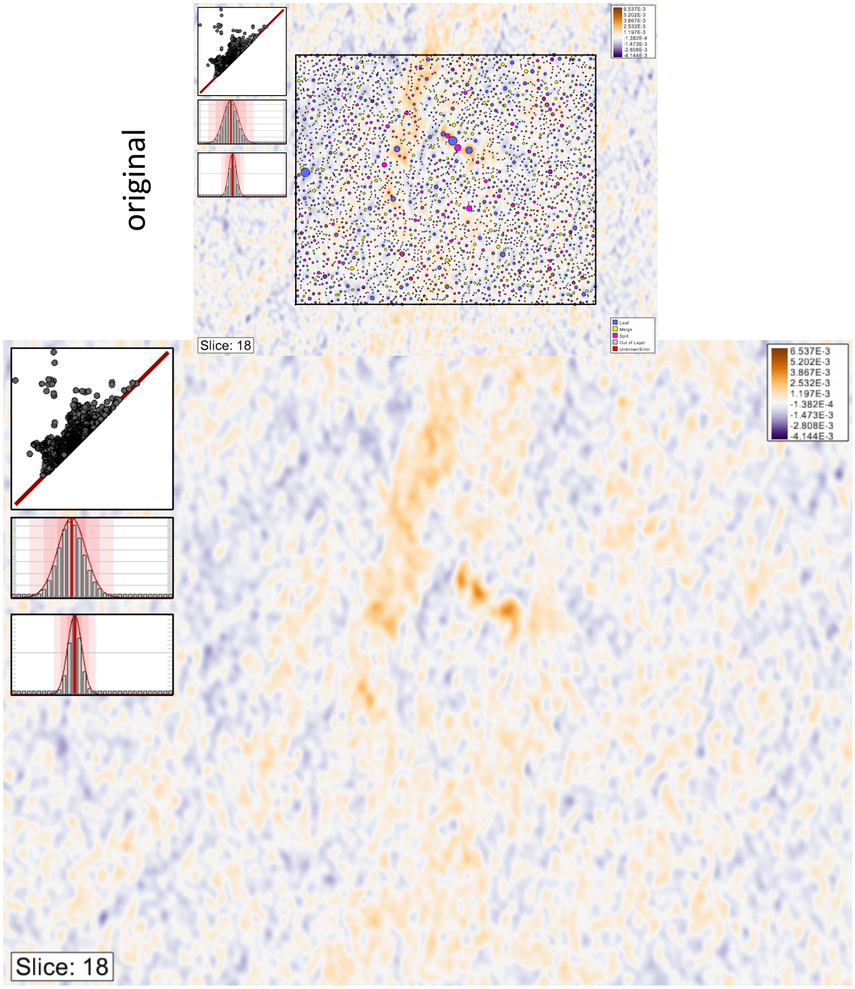
original



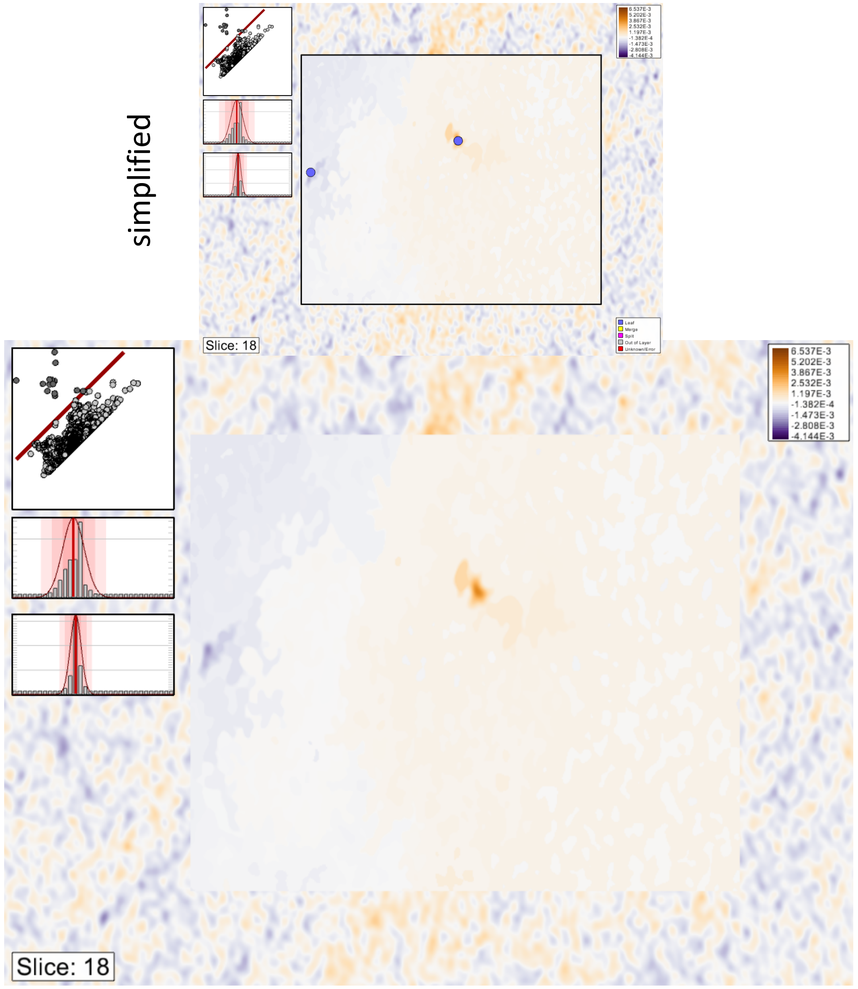
simplified



original

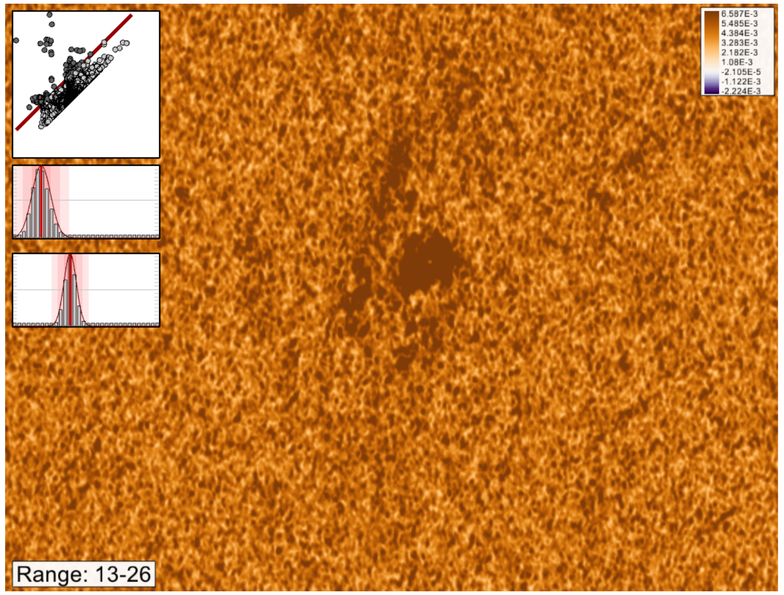


simplified

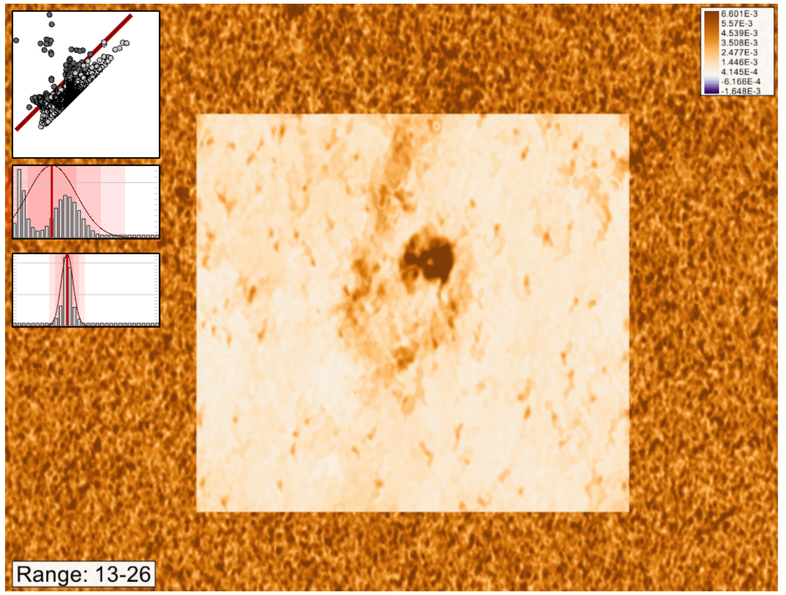


MOMENT 0 ANALYSIS

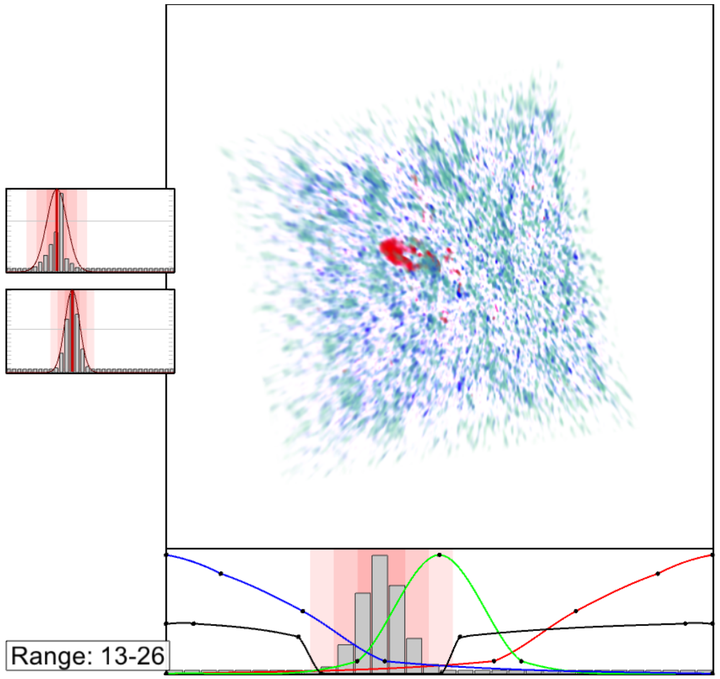
original

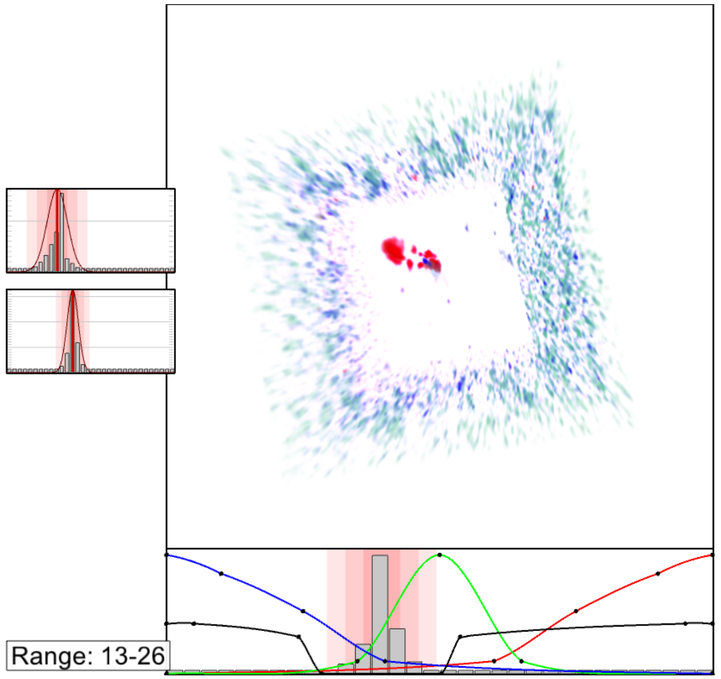


simplified

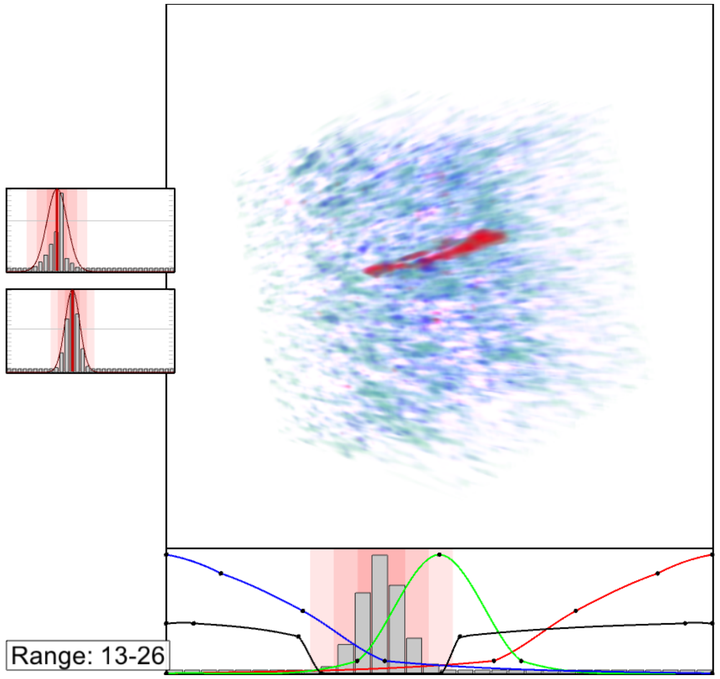


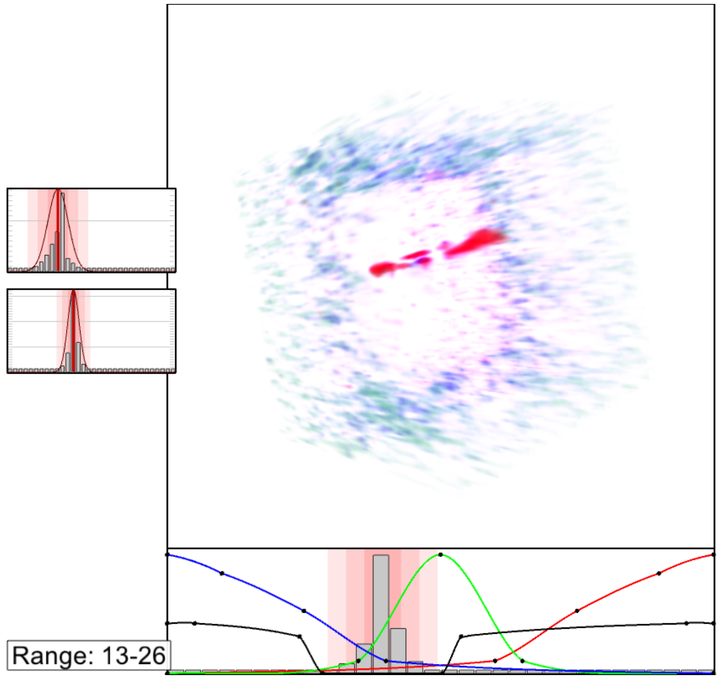
Observing the red shift



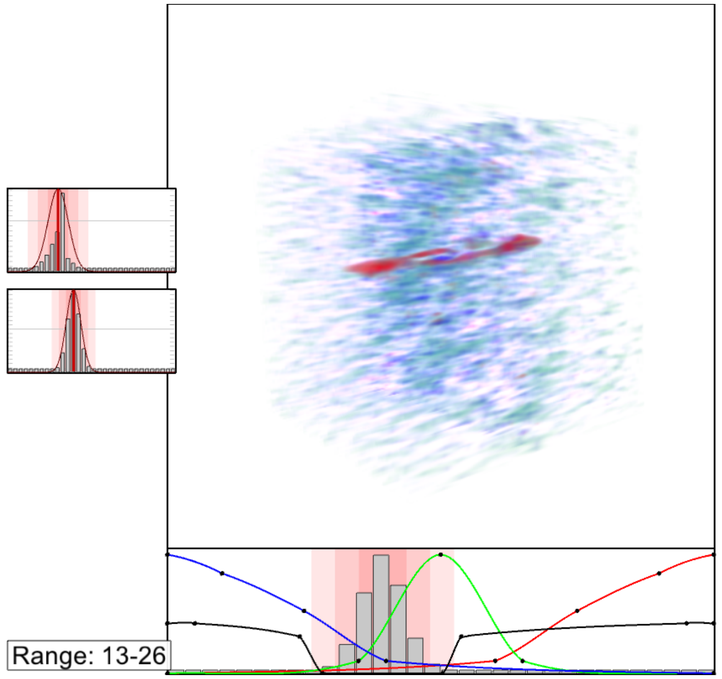


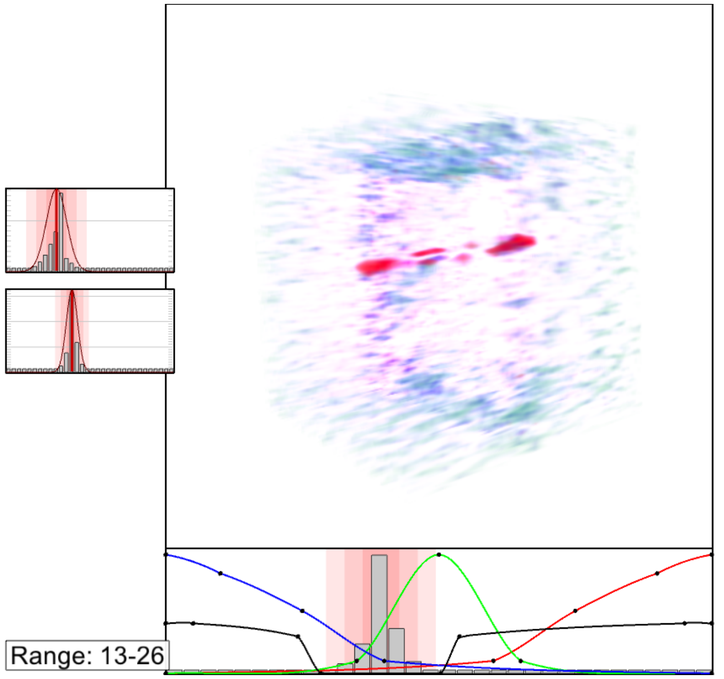
Observing the red shift





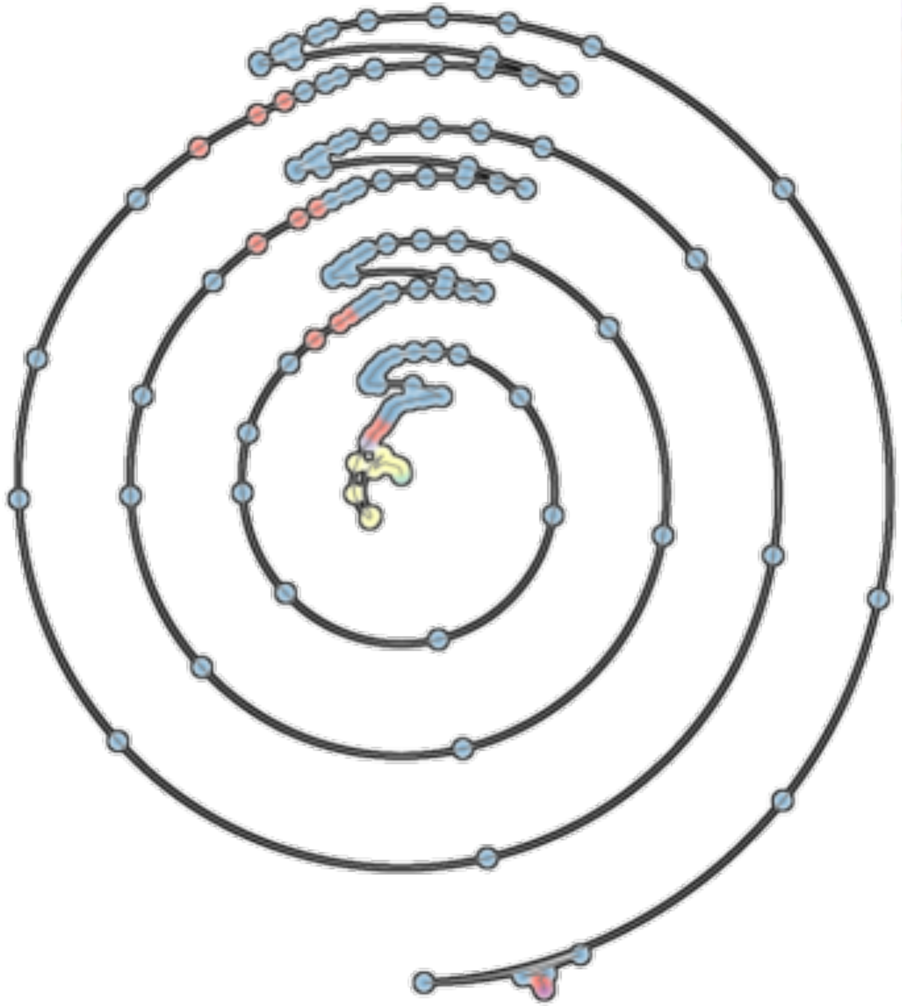
Observing the red shift





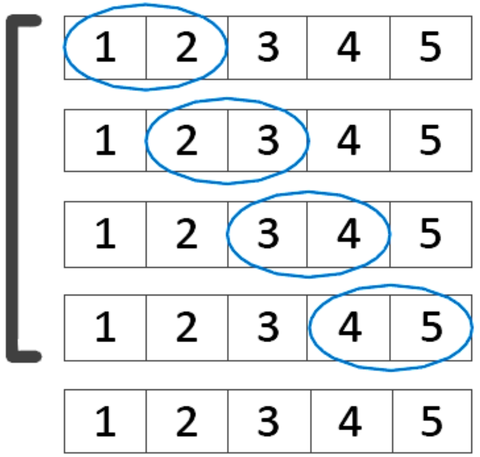
Software Visualization
Circular patterns in a program

An example



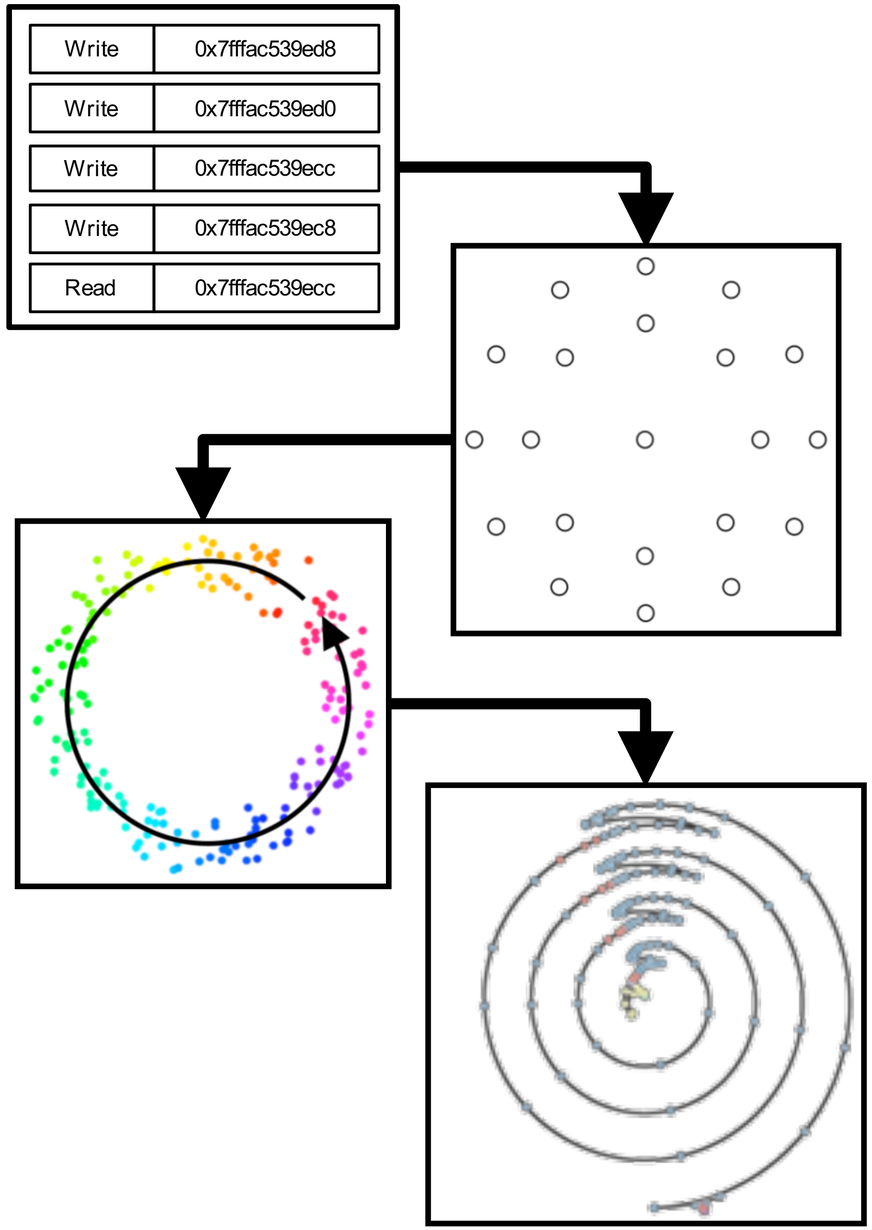
File: sort.cpp

```
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:}
```



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

Convert memory reference traces to a point cloud

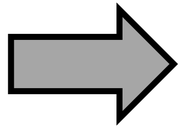


- Execute an application to capture memory reference trace
- Convert to high-dimensional point cloud
- Topological analysis identify cycles
- Visualize result

Capturing a memory reference trace

File: sort.cpp

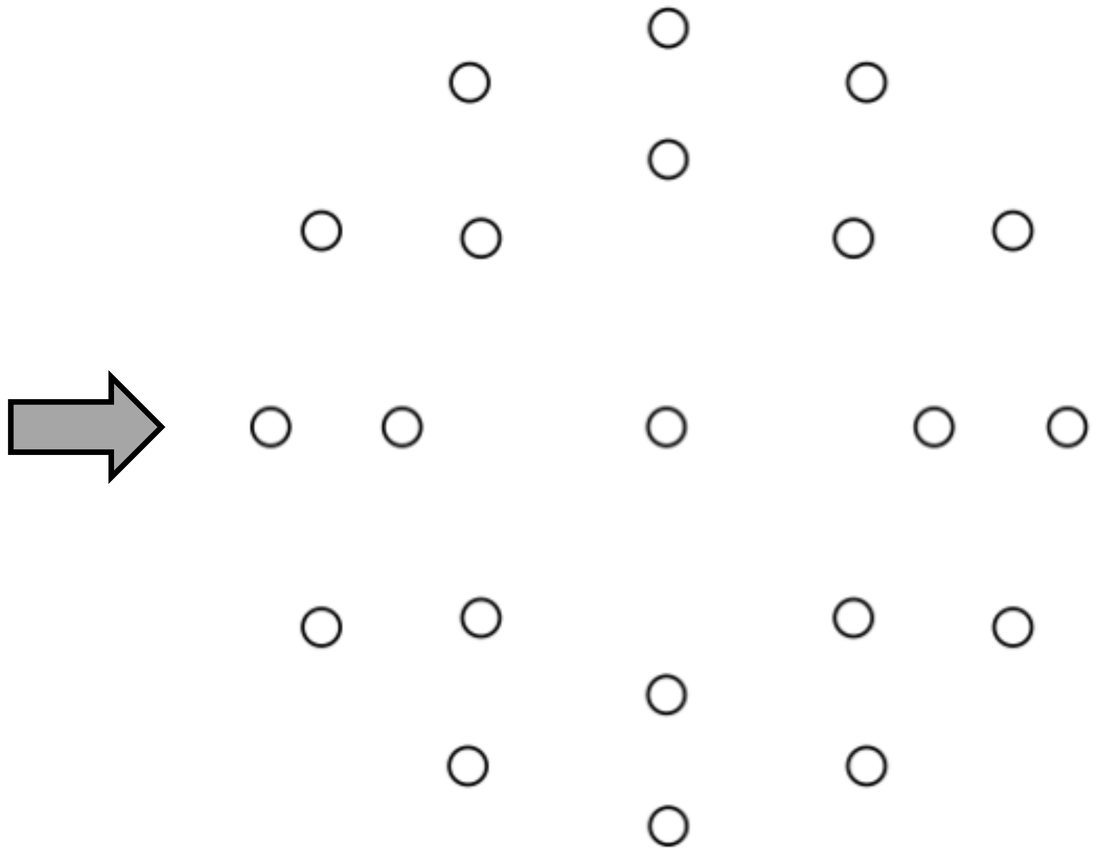
```
 1:void bubblesort(std::vector<double>& v){
 2:  for(unsigned end=v.size()-1; end >= 0; end--){
 3:    bool swapped = false;
 4:    for(unsigned i=0; i<end; i++){
 5:      if(v[i] > v[i+1]){
 6:        std::swap(v[i], v[i+1]);
 7:        swapped = true;
 8:      }
 9:    }
10:    if(!swapped) break;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



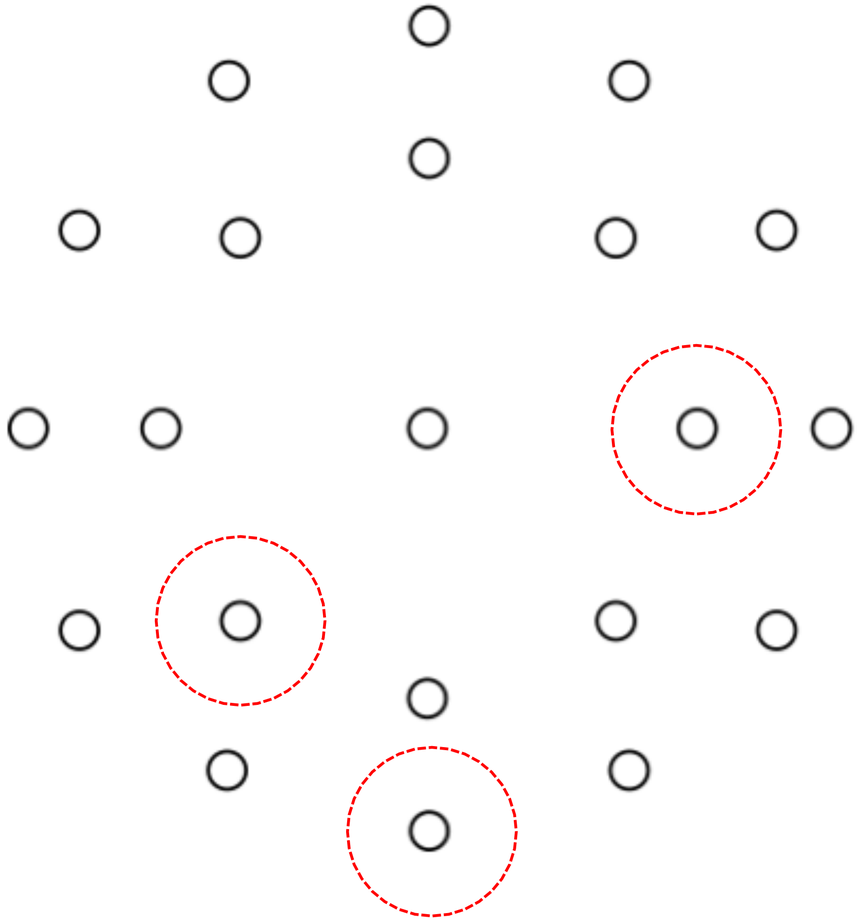
Write	0x7fffac539ed8
Write	0x7fffac539ed0
Write	0x7fffac539ecc
Write	0x7fffac539ec8
Read	0x7fffac539ecc
Read	0x7fffac539ec8
Write	0x7fffac539eb8
Write	0x7fffac539eb0

Memory reference trace to point cloud

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



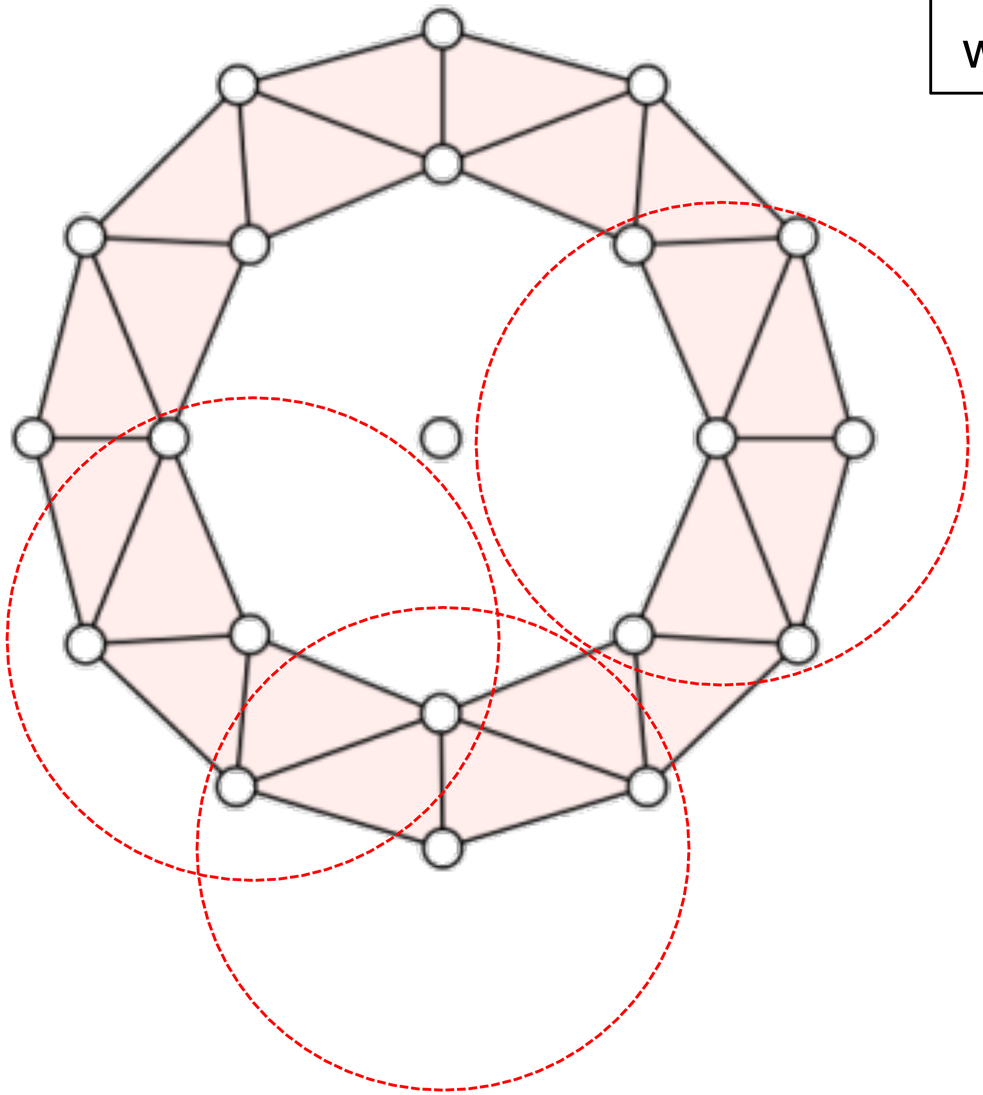
Topological data analysis to identify cycles



No connectivity
No cycles

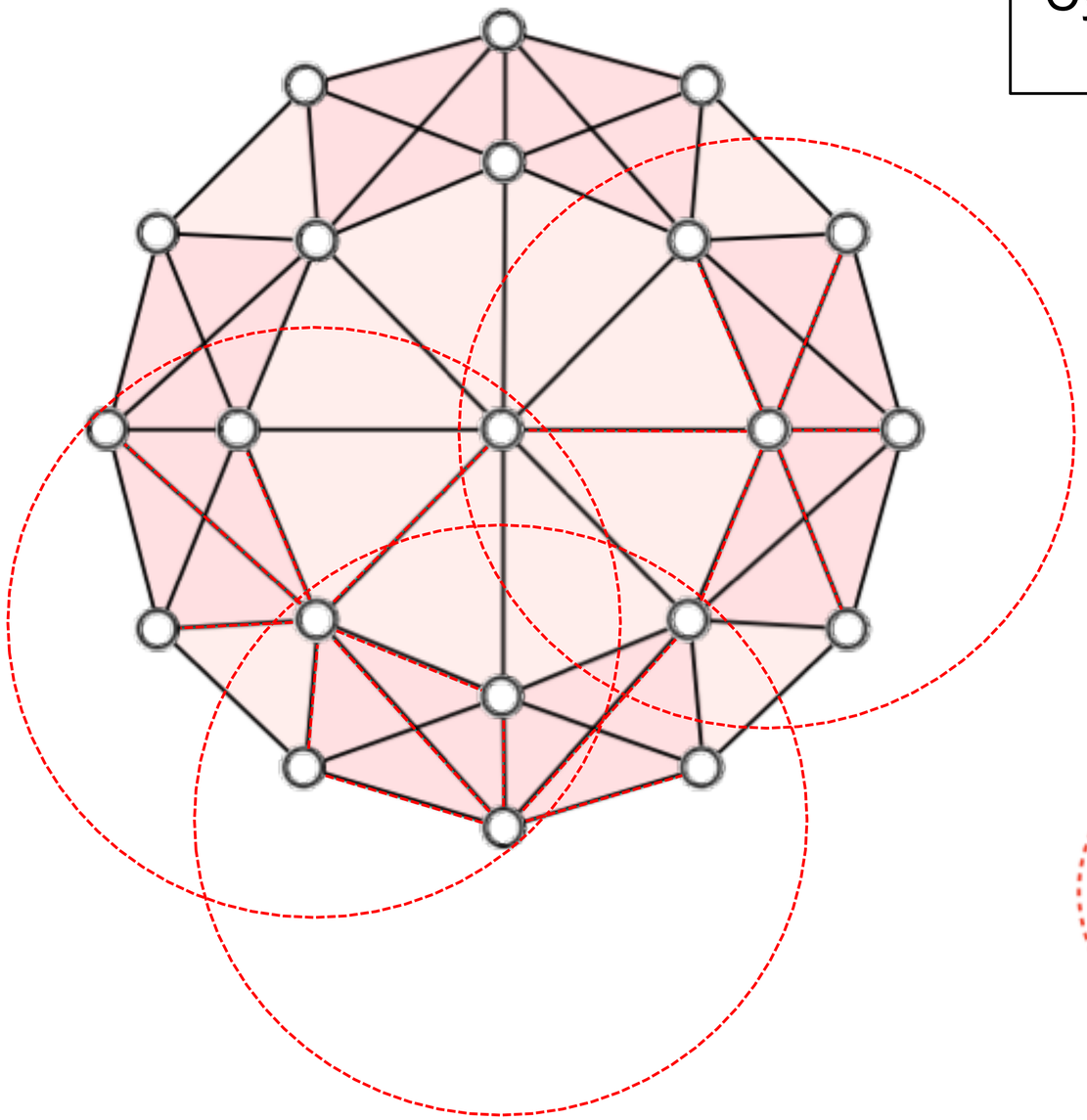
ε_0

Topological data analysis to identify cycles



Cycle is found with birth time ε_1

Topological data analysis to identify cycles



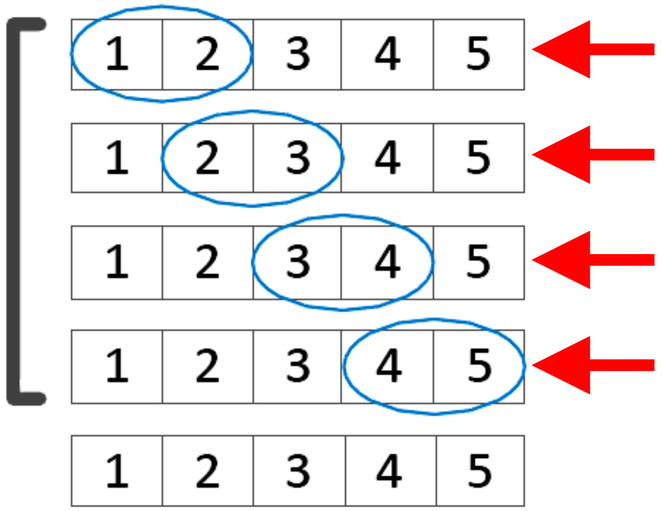
Cycle has died at time ε_2

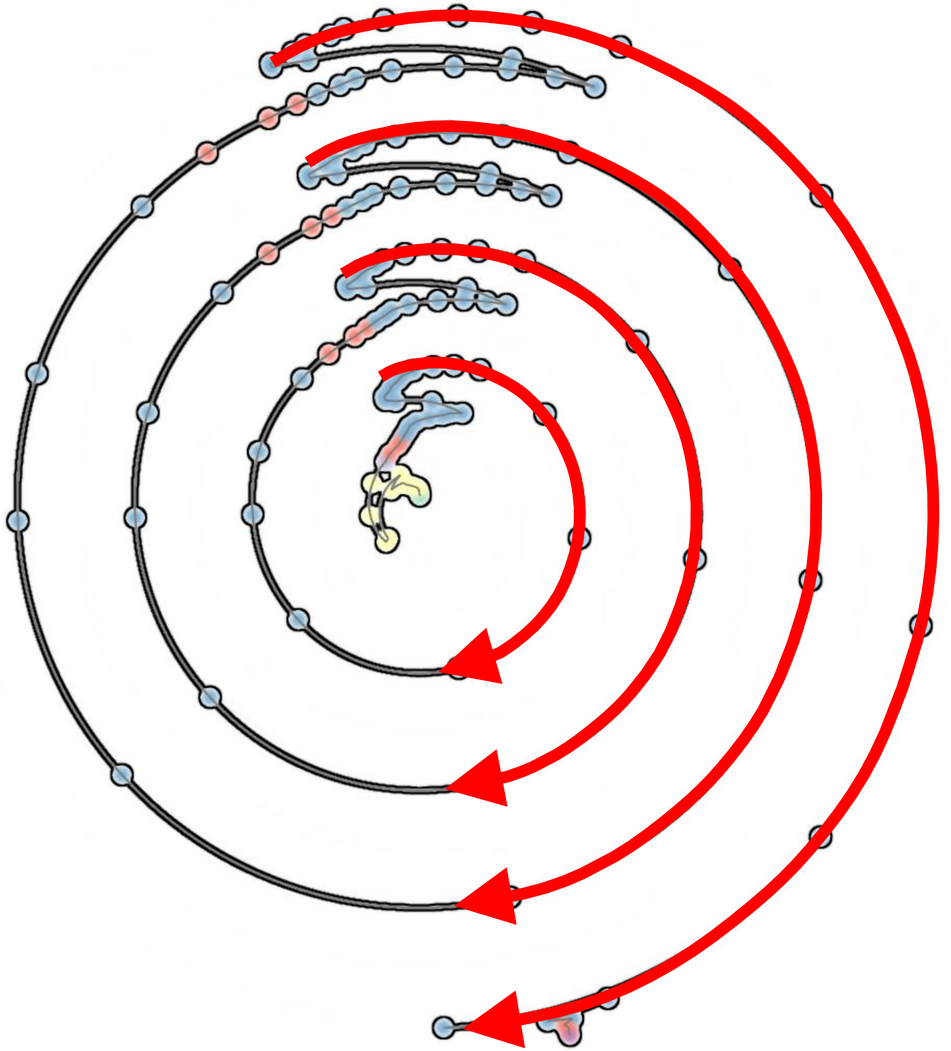
ε_2

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:}
```

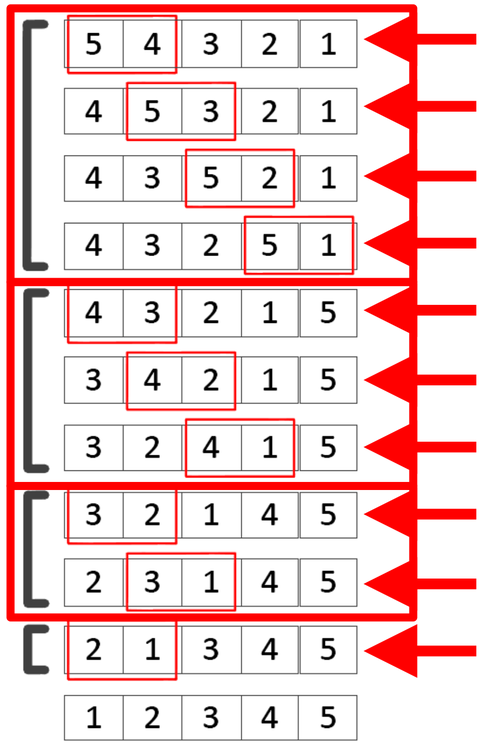


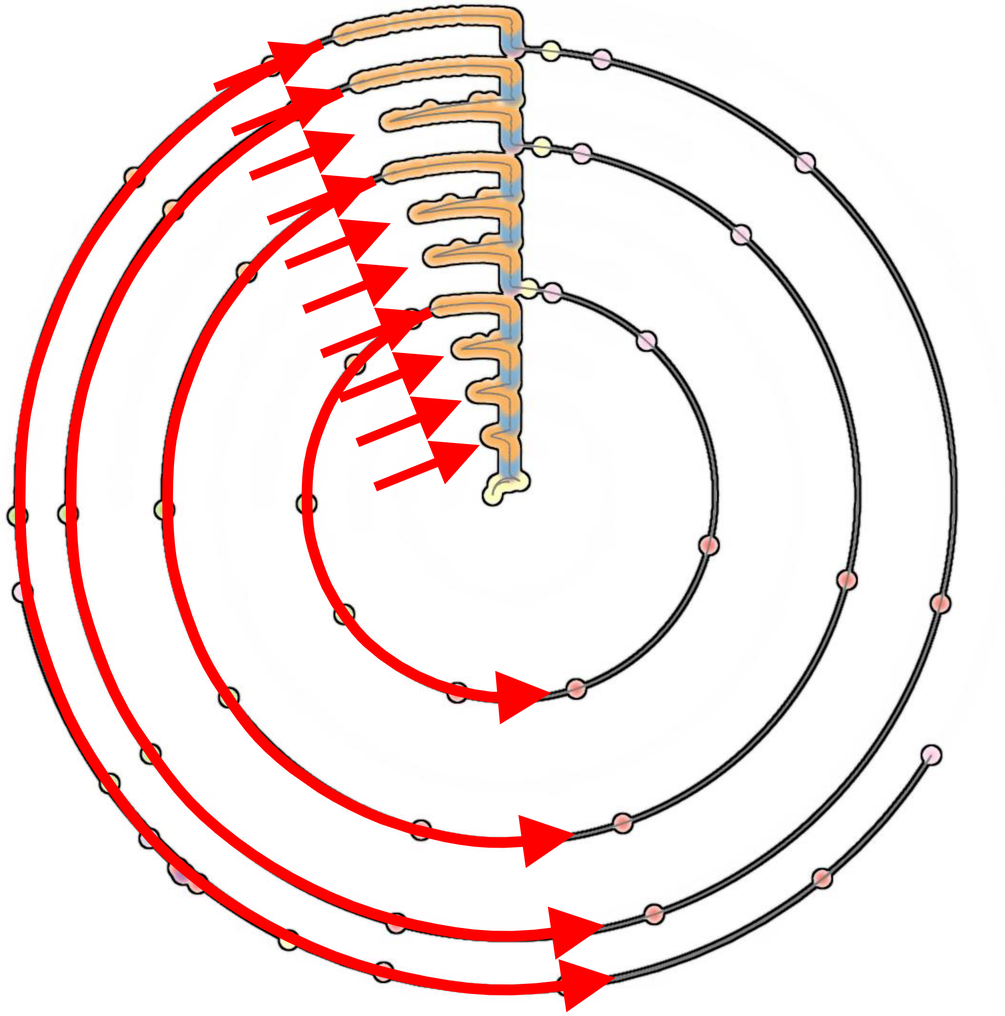


Data dependent structures

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:}
```

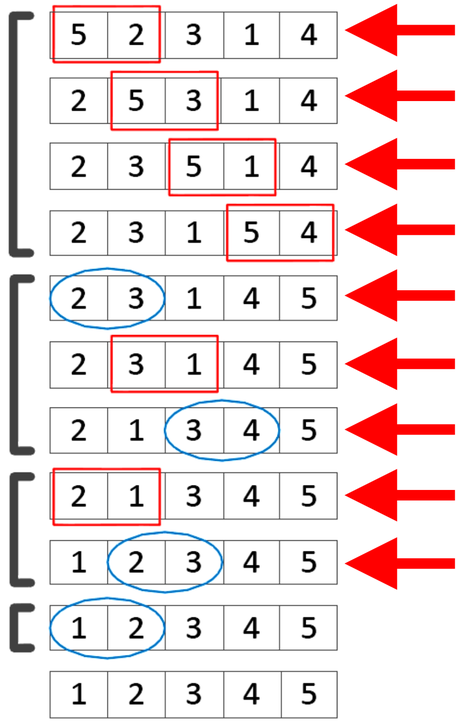


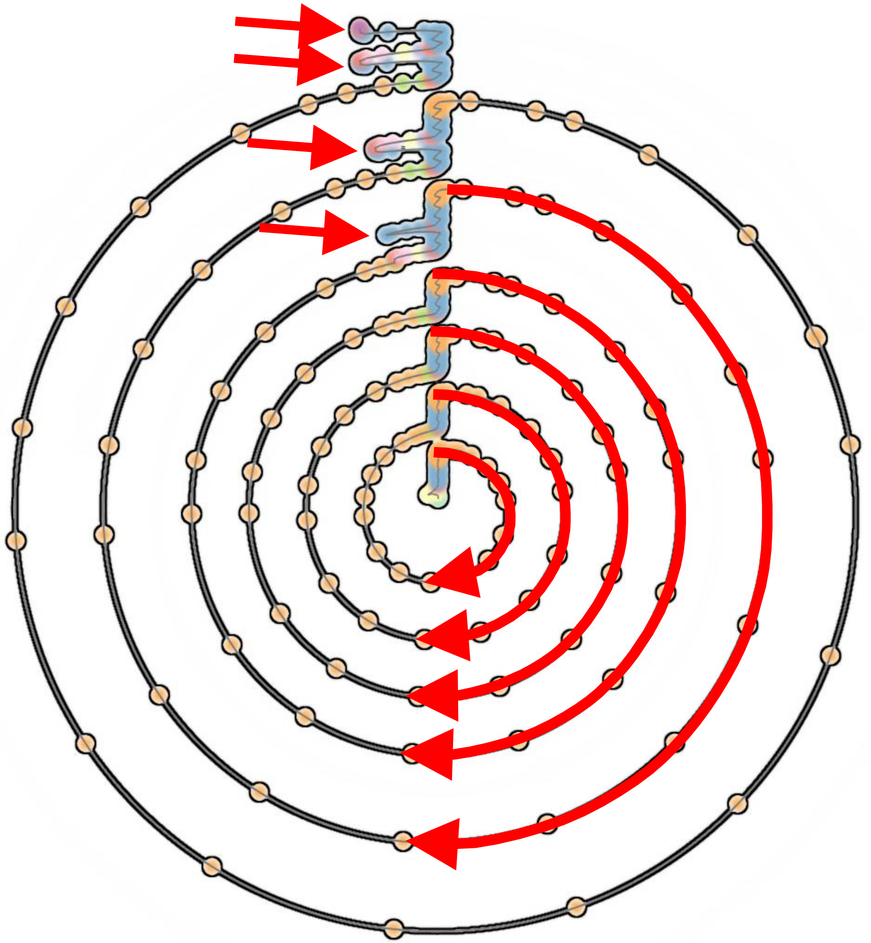


Data dependent structures

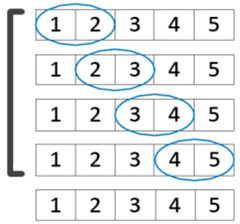
File: sort.cpp

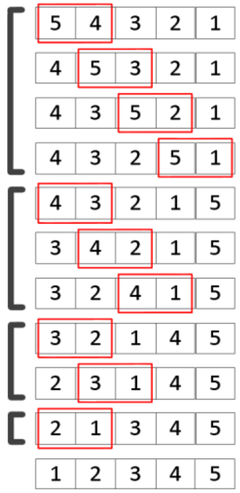
```
 1:void bubblesort(std::vector<double>& v){
 2:  for(unsigned end=v.size()-1; end >= 0; end--){
 3:    bool swapped = false;
 4:    for(unsigned i=0; i<end; i++){
 5:      if(v[i] > v[i+1]){
 6:        std::swap(v[i], v[i+1]);
 7:        swapped = true;
 8:      }
 9:    }
10:    if(!swapped) break;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

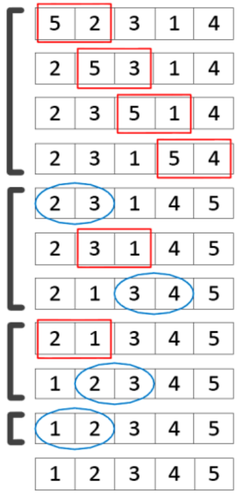


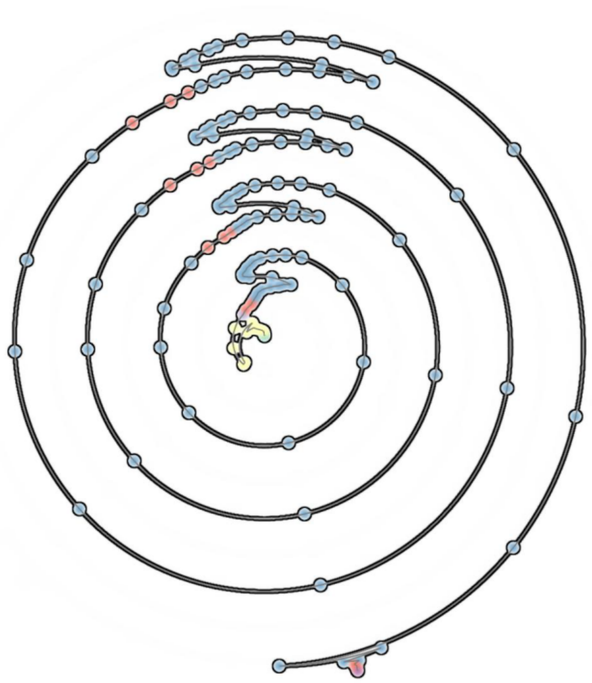


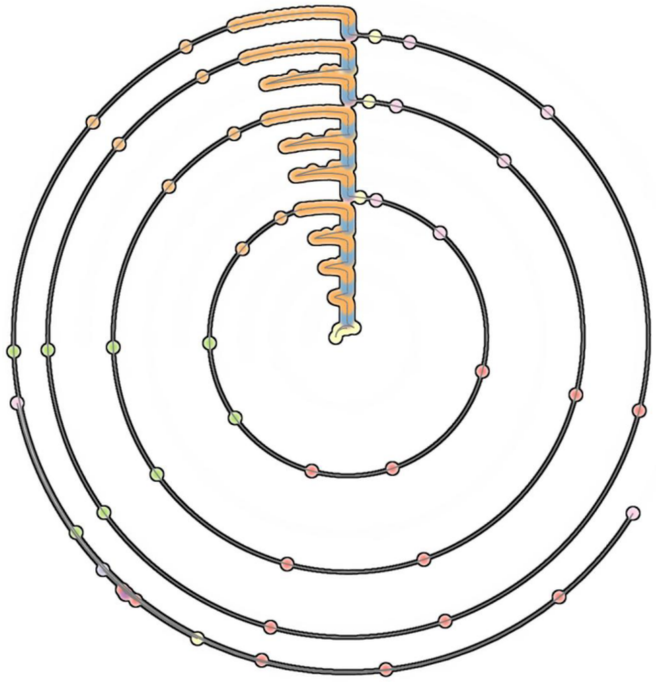
Data dependent structures

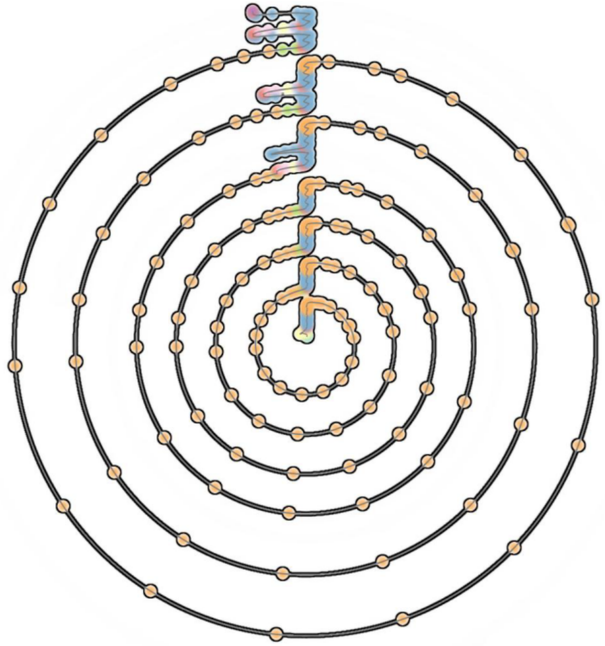








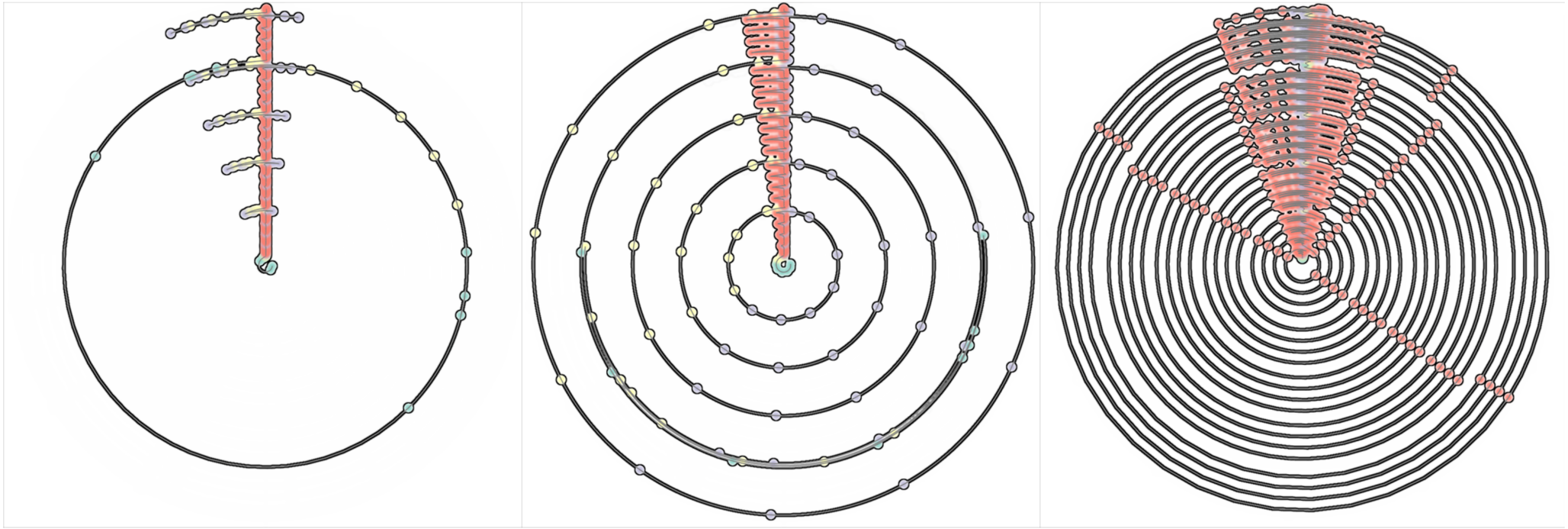




Algorithm dependent structures

File: matmult.cpp

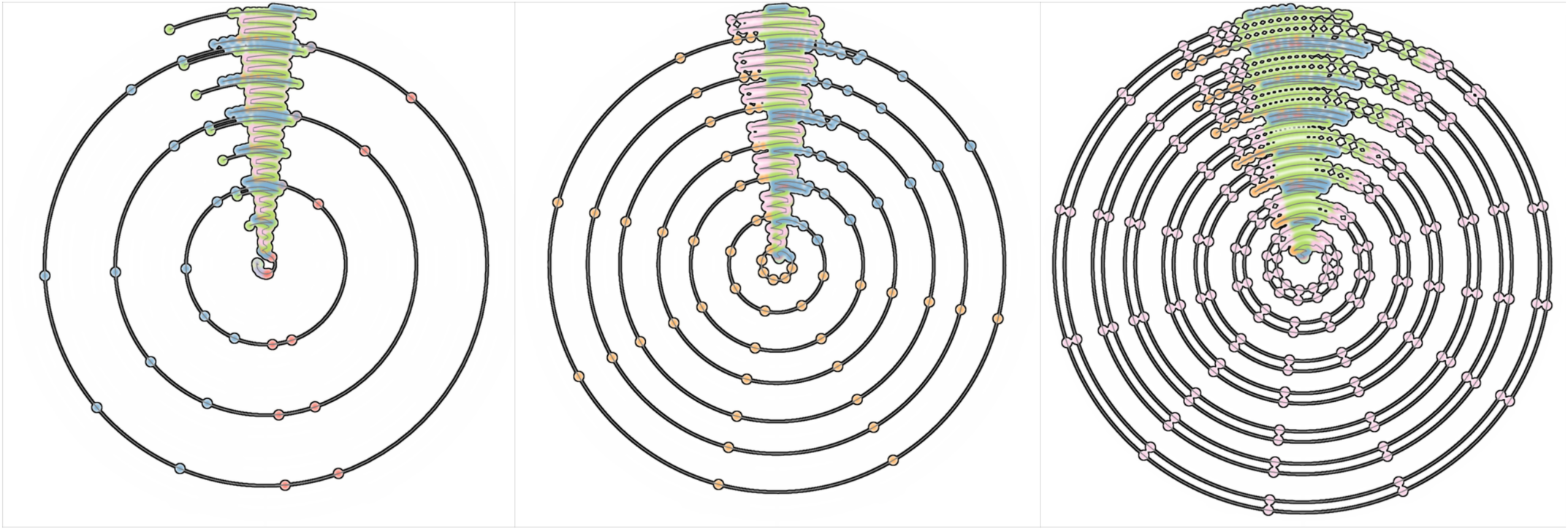
```
1:  unsigned int i, j, k;
2:  for (i = 0; i < N; i++)
3:    for (j = 0; j < N; j++)
4:      for (k = 0; k < N; k++)
5:        linC[i*N + j] += linA[i*N + k] * linB[k*N + j];
```



Algorithm dependent structures

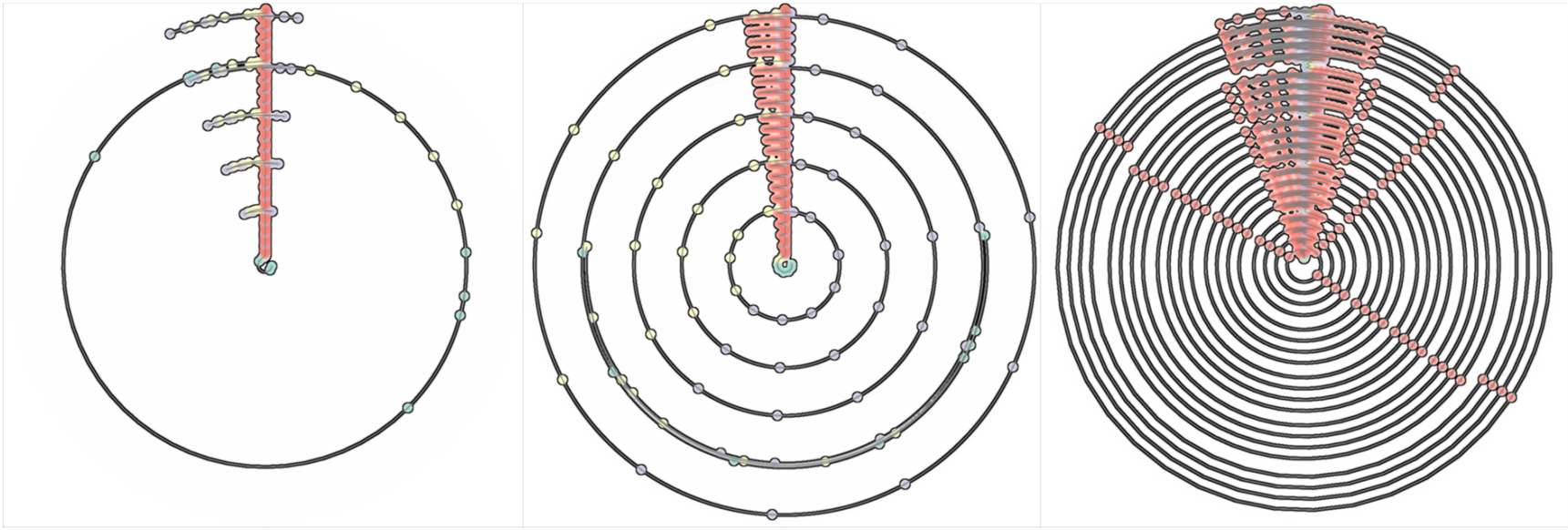
File: blocked-matmult.cpp

```
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:        }
```

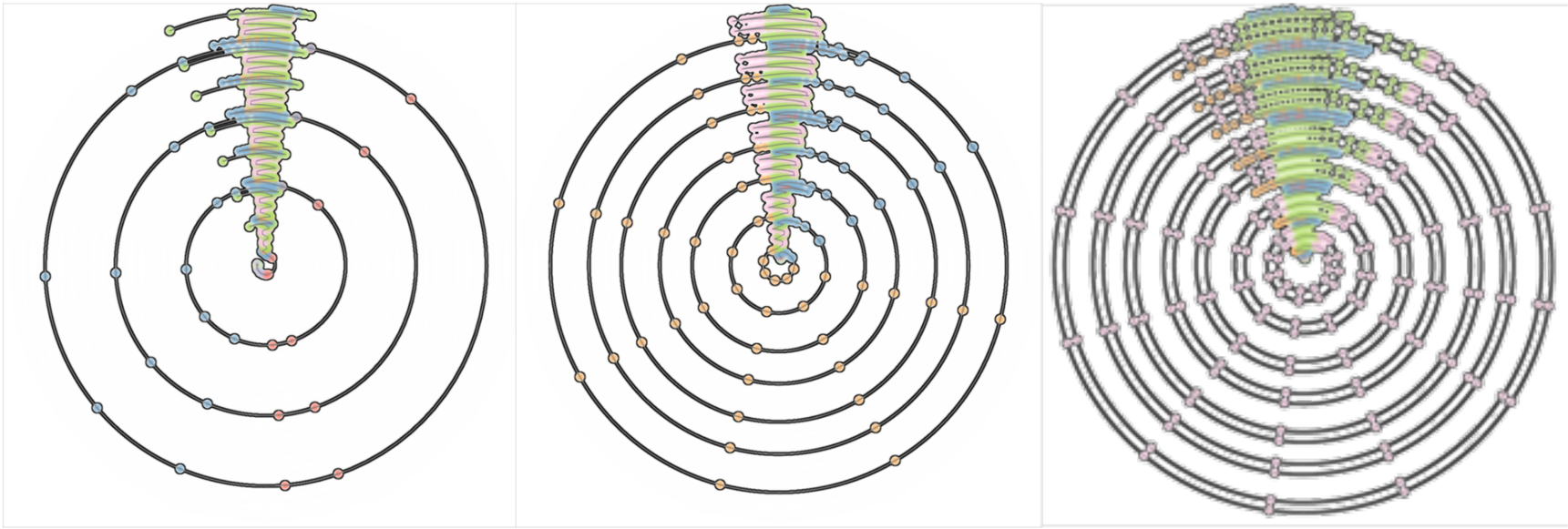


Algorithm dependent structures

Naïve Matrix Multiply



Blocked Matrix Multiply



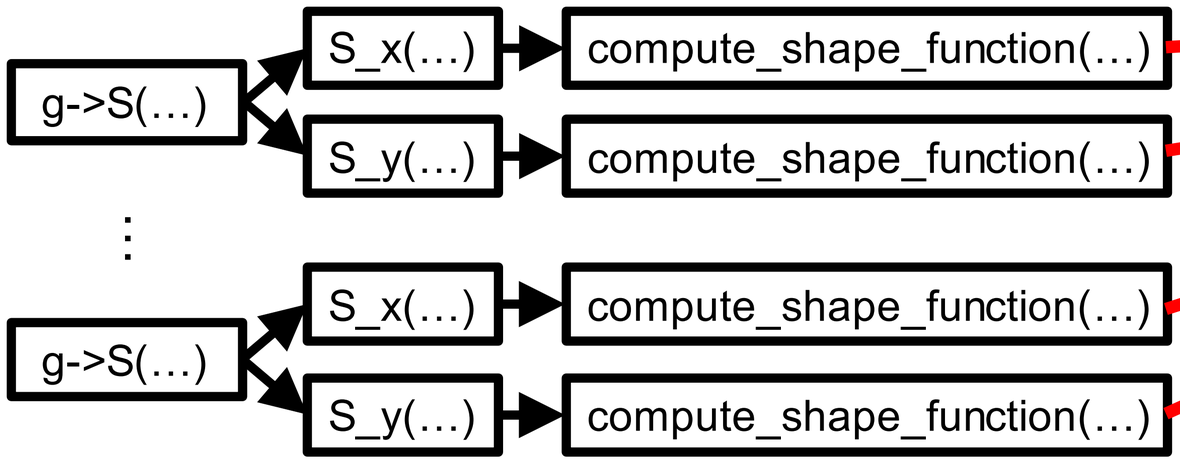
Non-loop based structures

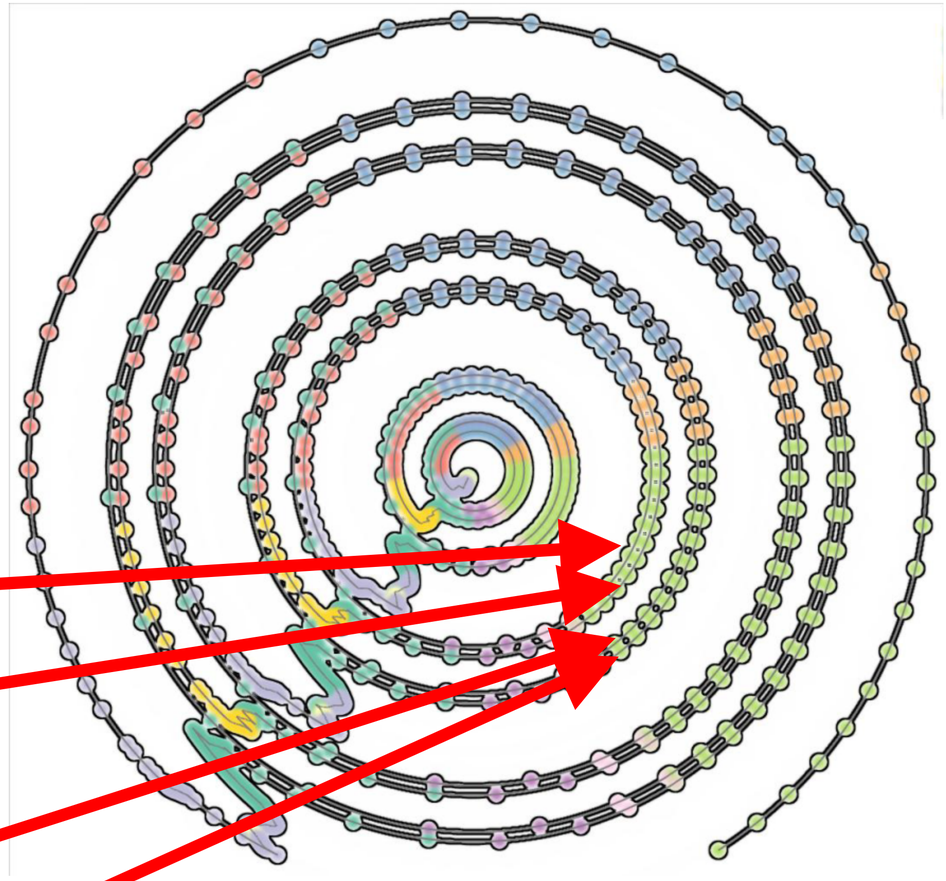
File: MPM.cpp

```
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

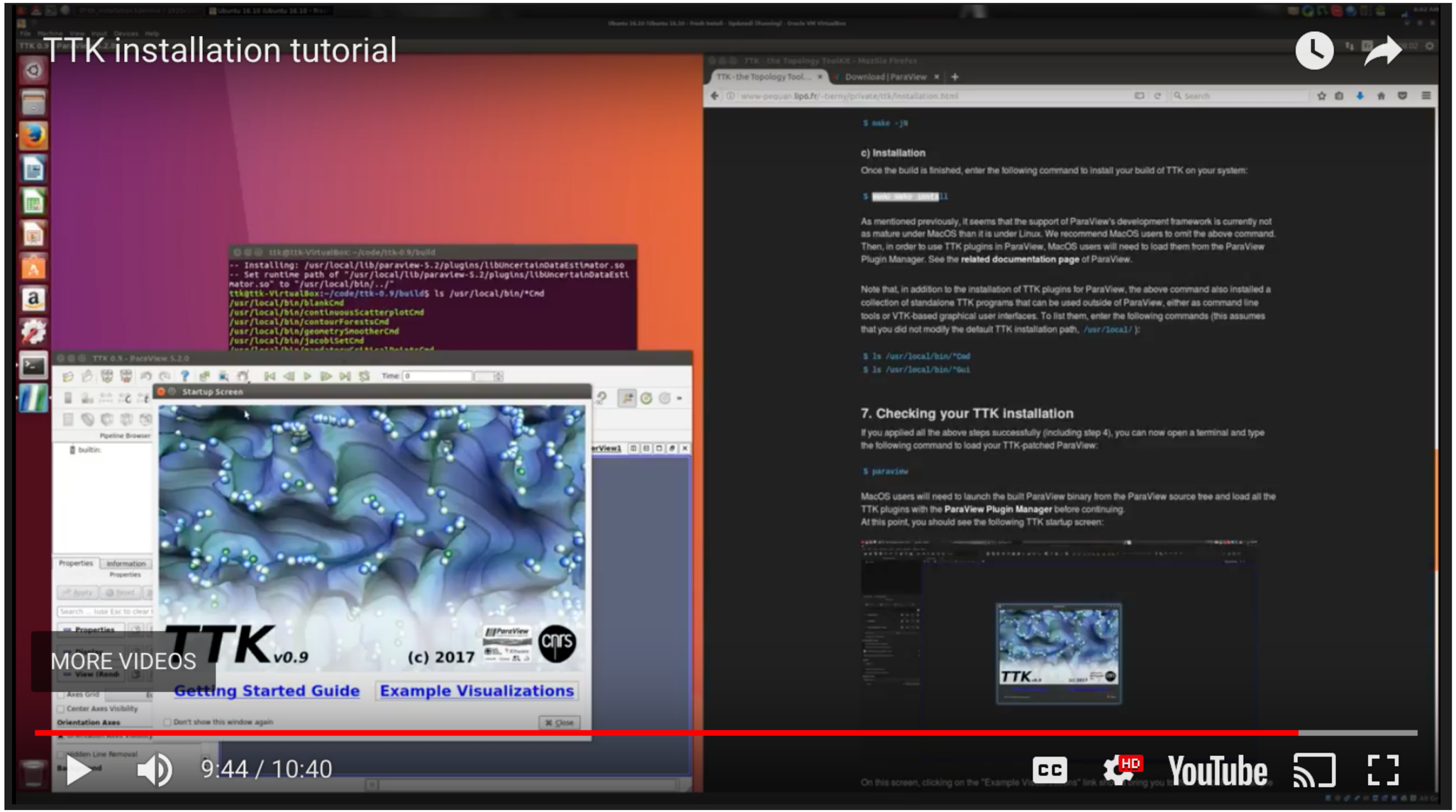
```
1:double S(int i, int j, const Point& p){ ... }
2:unsigned indexify(unsigned i, unsigned j) const { ... }
3:double S_x(int i, double x){ ... }
4:double S_y(int j, double y){ ... }
5:static double compute_shape_function(int cell, double position, double cell_size){
6:  // This is the distance of "position" 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:}
```





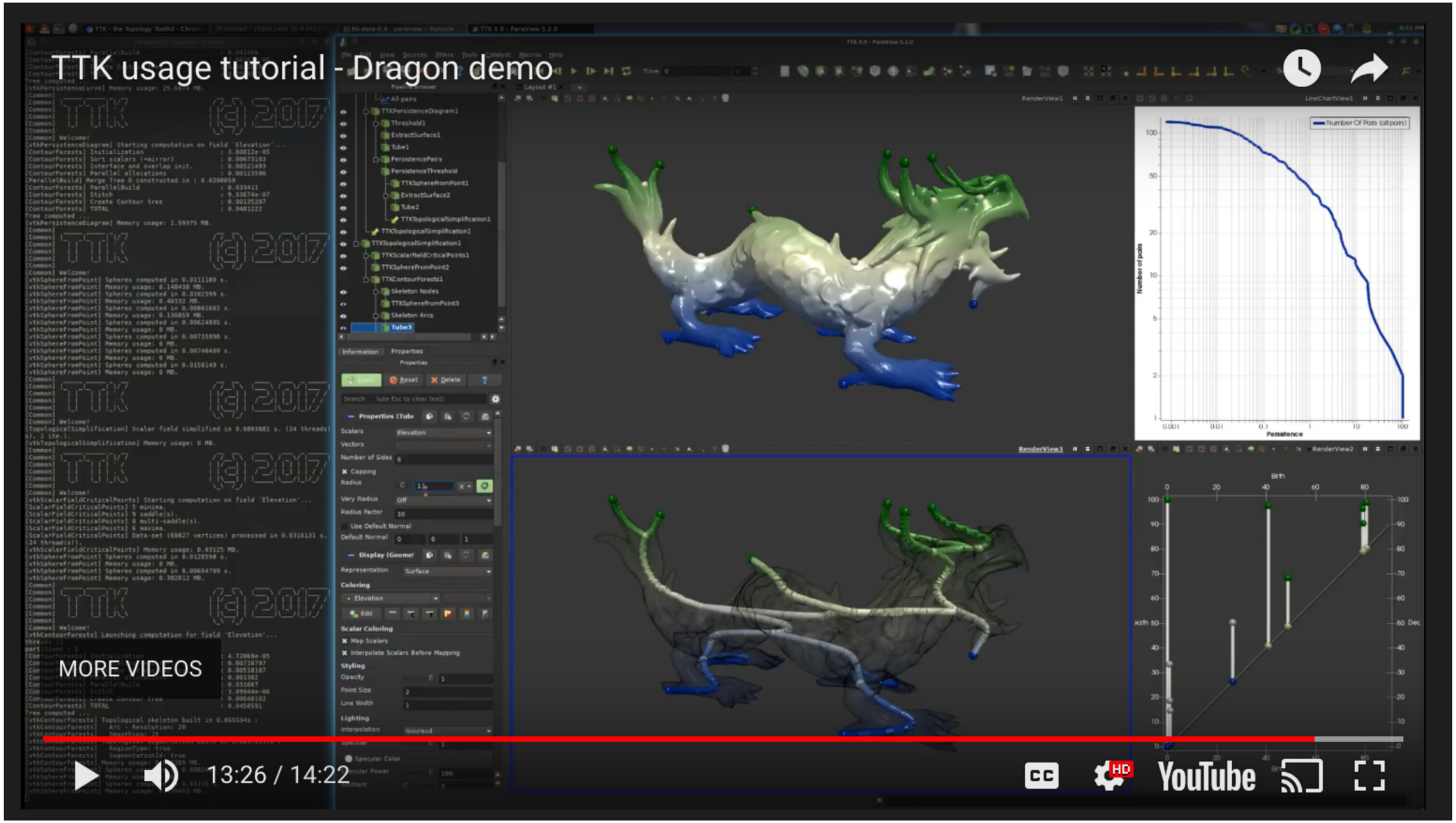
Topology Tool Kit
TTK

Installation Demo



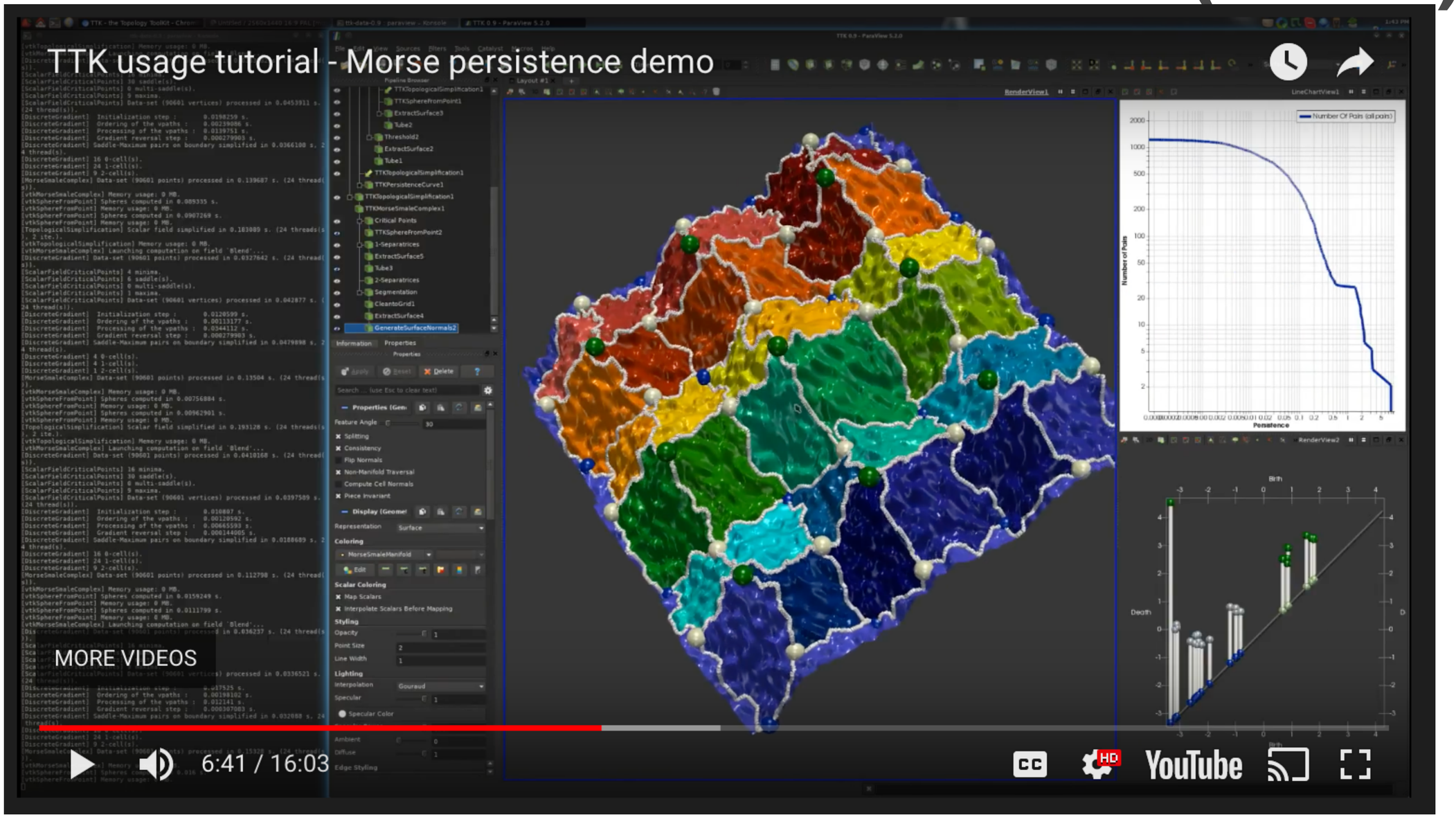
https://youtu.be/etAe13KEWsk

Dragon Demo (contour tree)



https://youtu.be/YVk9vRKIEX8

Morse Persistence Demo (MSC)



https://youtu.be/xjKh6YTq5RA

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