Uncovering the topological features for the characterisation of Parkinson's disease

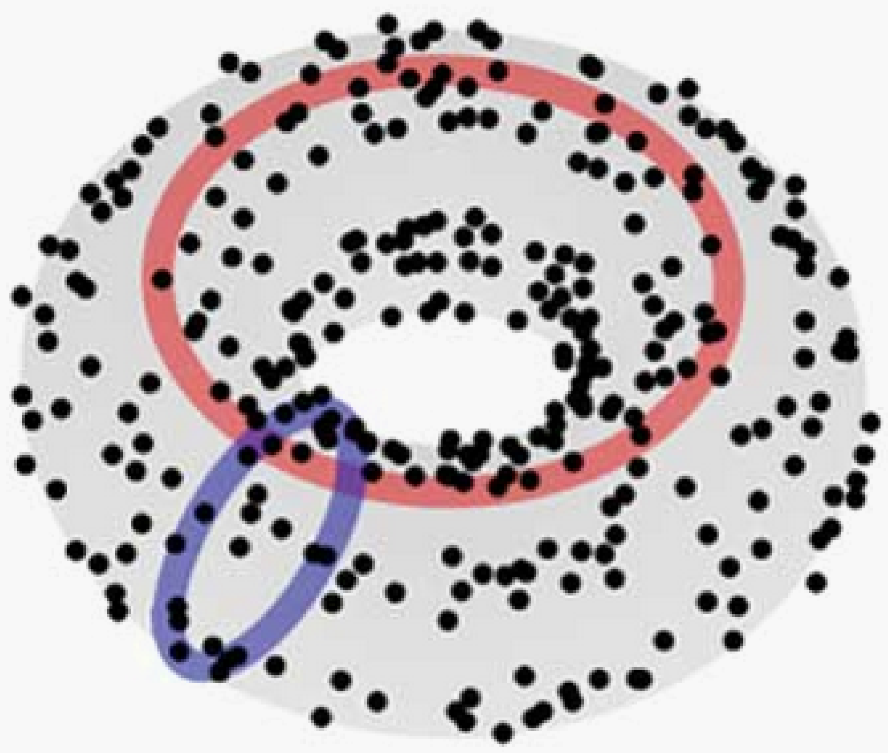
El Missaoui Imane

Masters student in Healthcare data science

Supervisors : Amael Broustet, Anne-Sophie Rolland

Hypothesis in geometric data analysis

Data has a shape: points are drawn from a geometric object that exists in a higher-dimensional space. Obtaining the properties of these shapes provides valuable information about the data

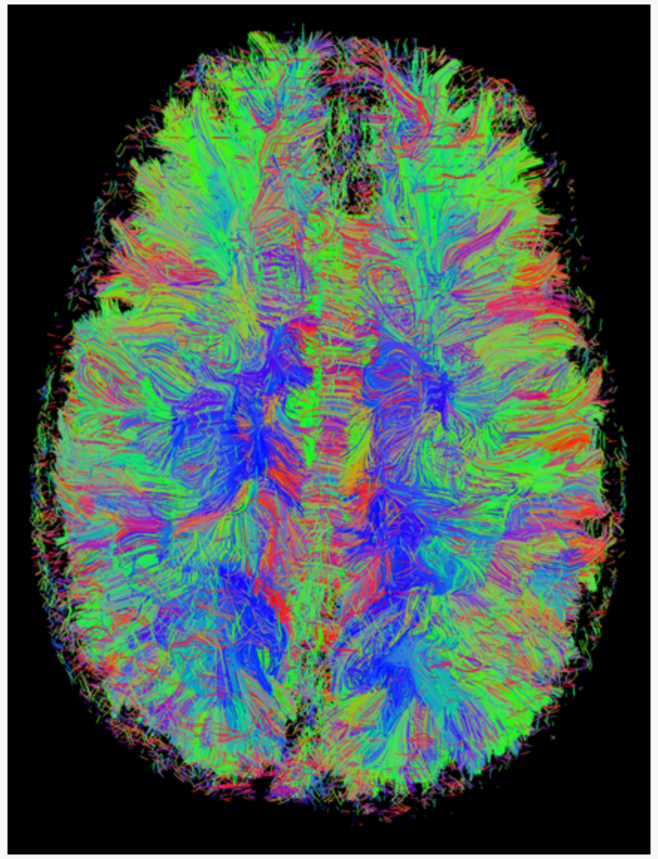


Contexte and Objective

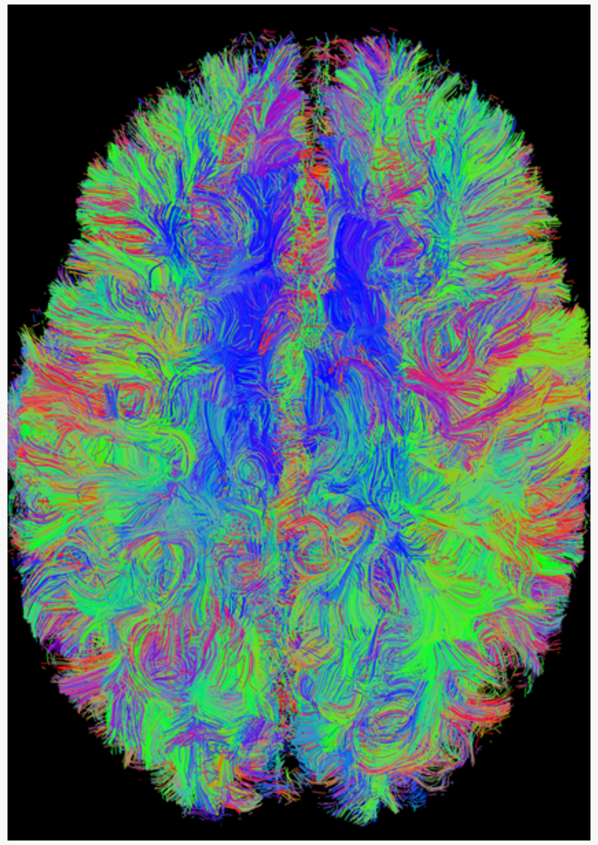
Neurodegenerative diseases impact the morphology and connectivity of the brain, making networks a suitable tool for studying and modeling their effects.

⇨ The brain is modelled as a complex network of interacting regions, two regions are connected by a fibre tract

PD



HEALTHY



Measure a difference between multiple stages of a neurodegenerative disease

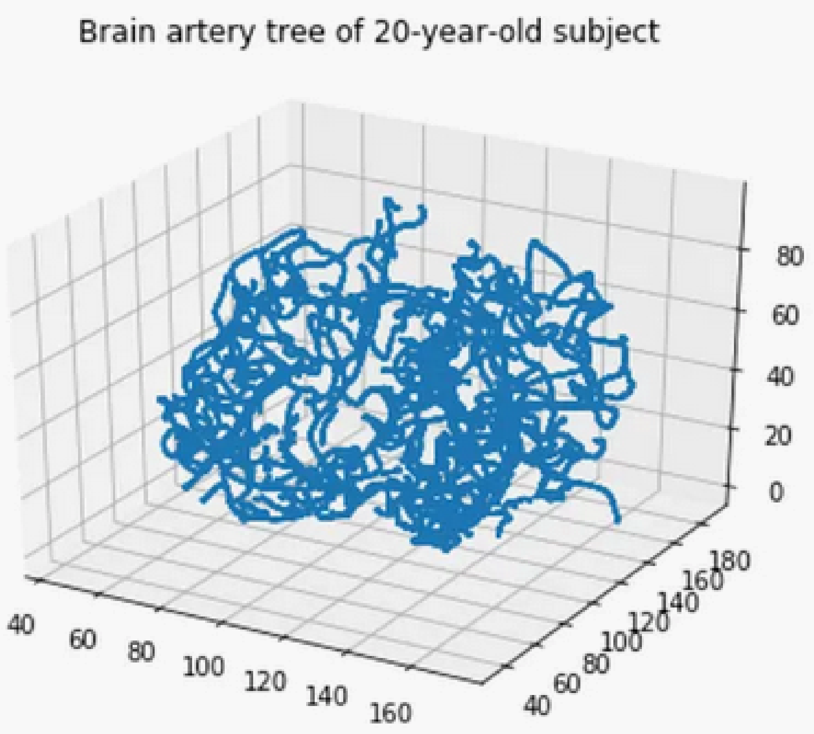
Example : analysis of brain artrees

Subjects of the study :

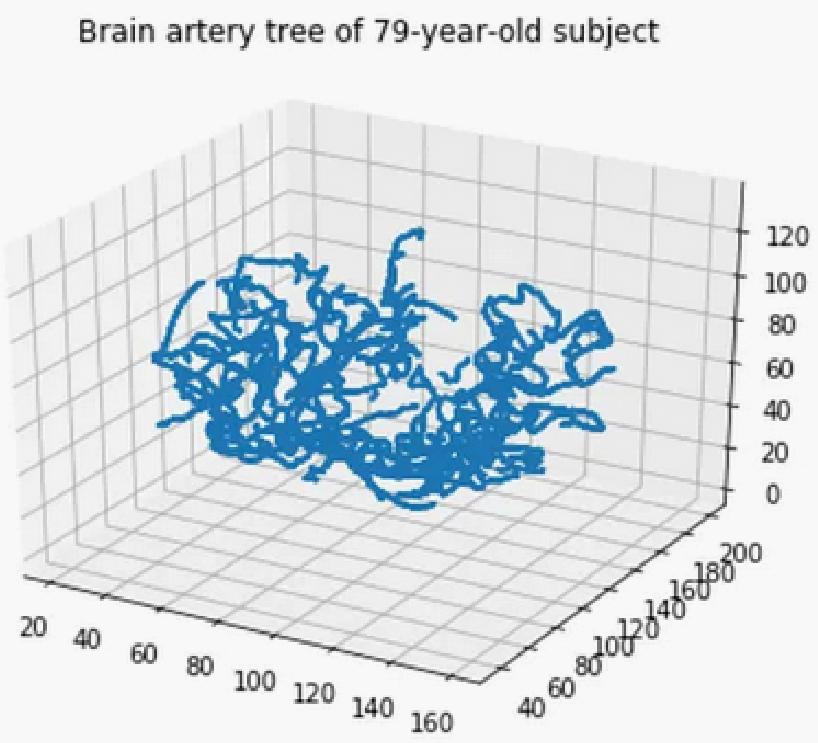
- non-pathological cases
- 98 individuals of age between 18 and 79

Results : distinguished between brain artery trees coming from 2 different age group

Group 1 : age < 45

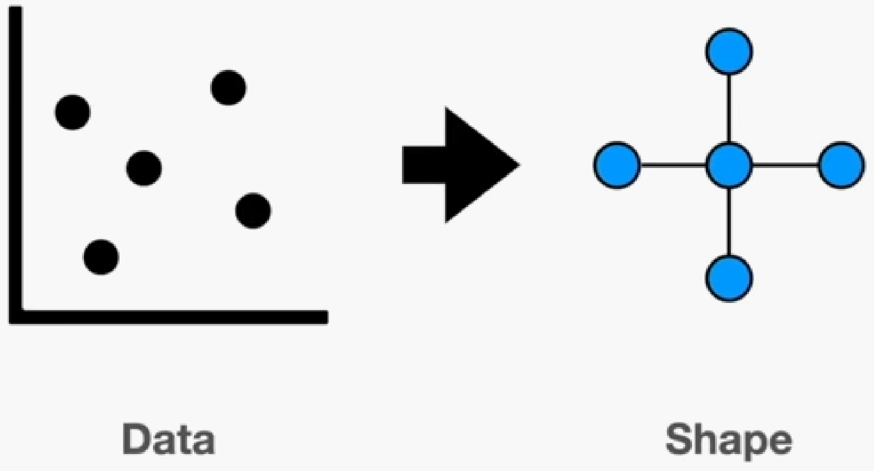


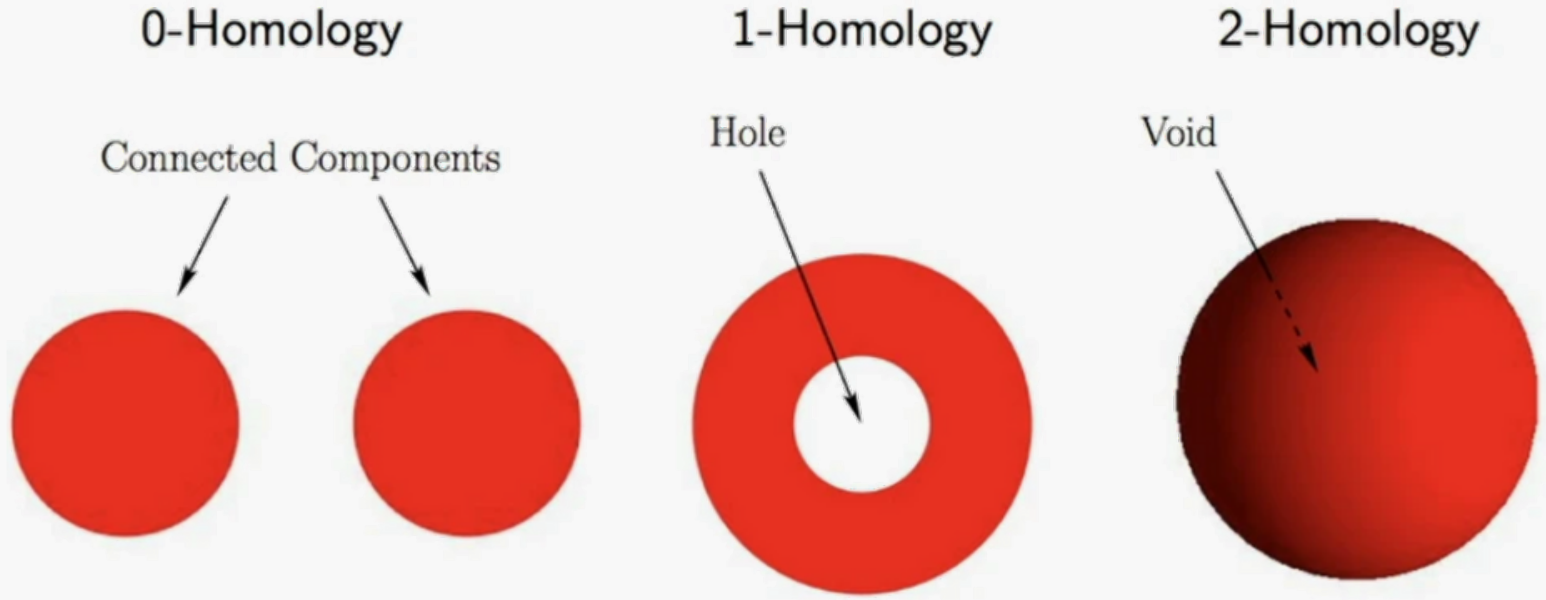
Group 2 : age > 45



[1] P. Bendich, J. S. Marron, E. Miller, A. Pieloch, and S. Skwerer, "Persistent Homology Analysis of Brain Artery Trees," Ann Appl Stat, vol. 10, no. 1, pp. 198–218, 2016, doi: 10.1214/15-AOAS886.

Topological data analysis (TDA)



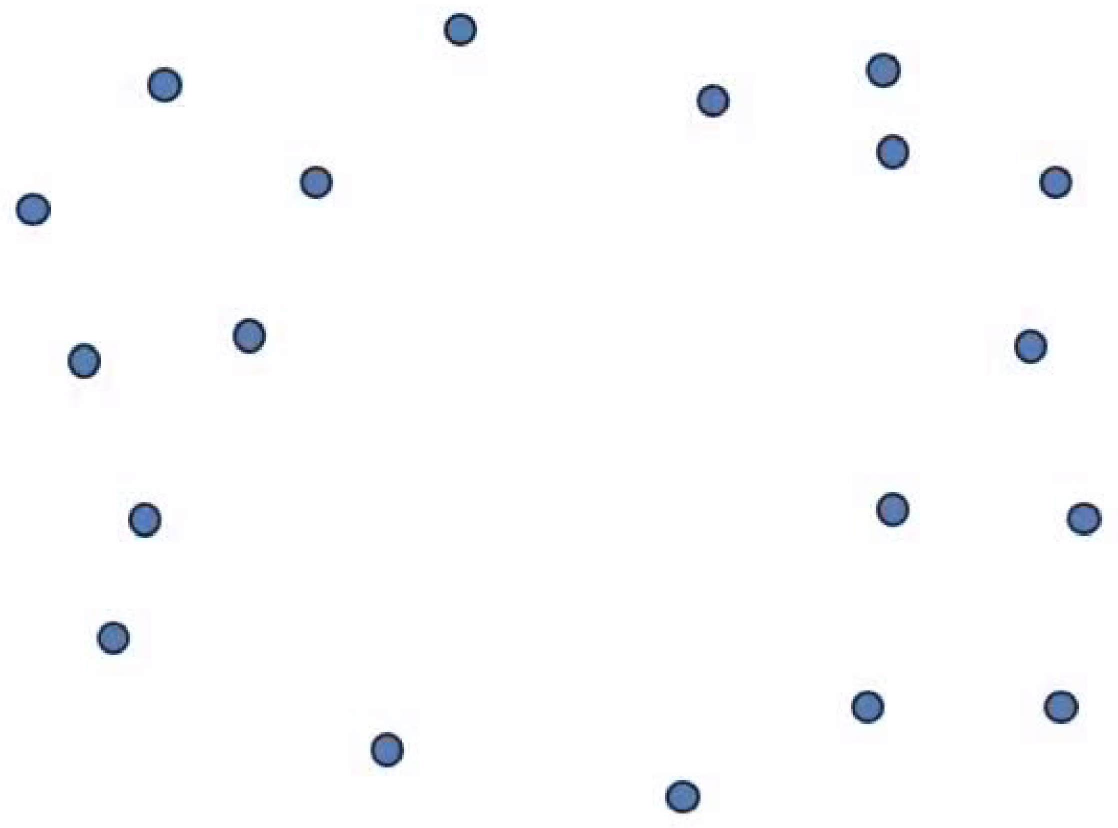


Why topological data analysis?

- Data-Driven Approach
- Studying complex high dimensional data without any assumptions
- **Shape has Meaning**; extracting **shapes**(**patterns**) of data
- Invariant to smooth deformations (stretching, bending, scaling)

Goal : compute and encode the topology of the data, by recording the **connected components**, **loops**, **cavities**, and **higher-dimensional structures**

Example:

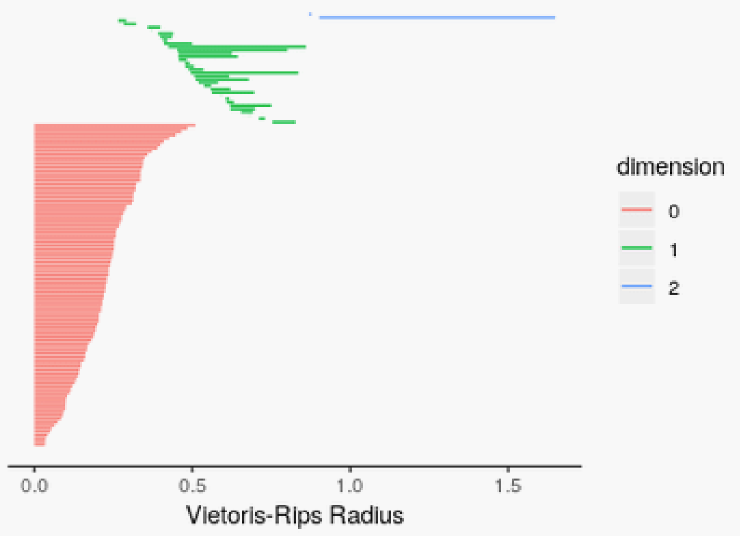


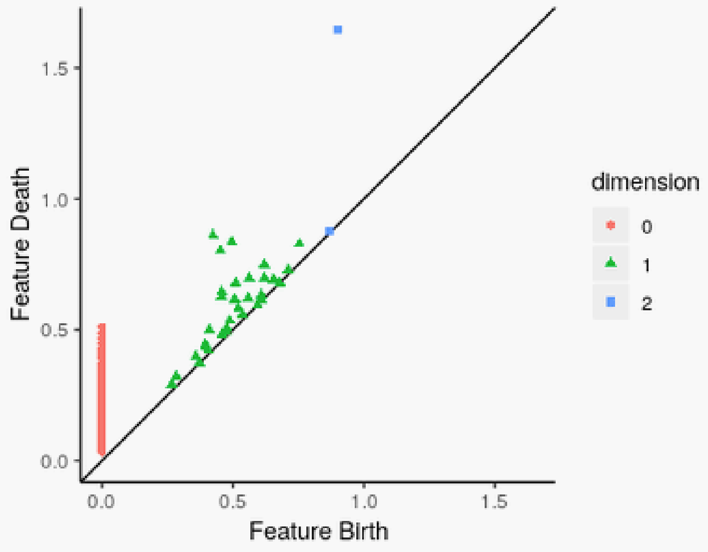
Persistent Homology

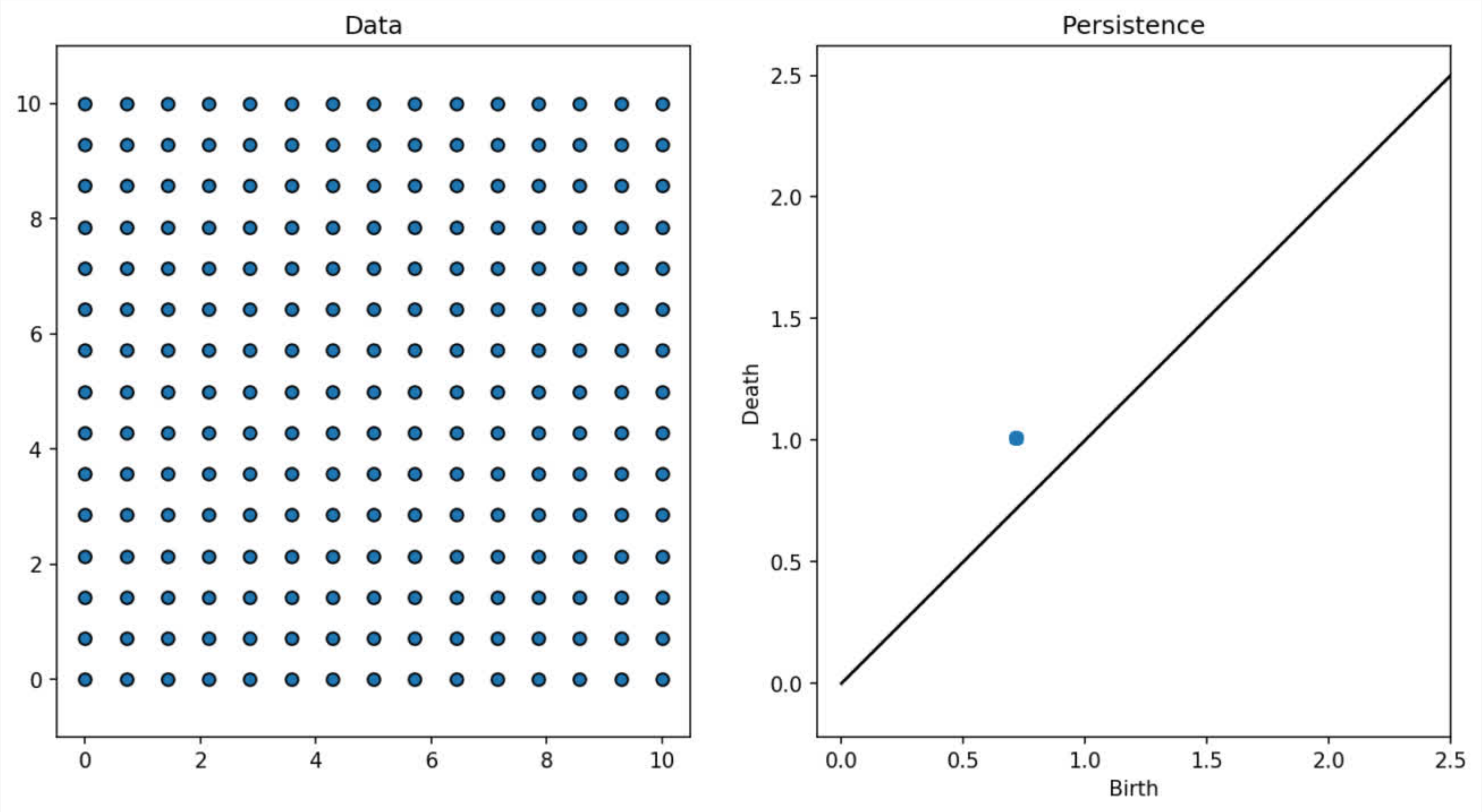
- Persistent homology study how points are connected to each other and how different topological features, such as loops, voids, or connected components, emerge and evolve as we vary a parameter called the "scale" or "resolution."

Represented as :

Barcode **or** Persistence diagram

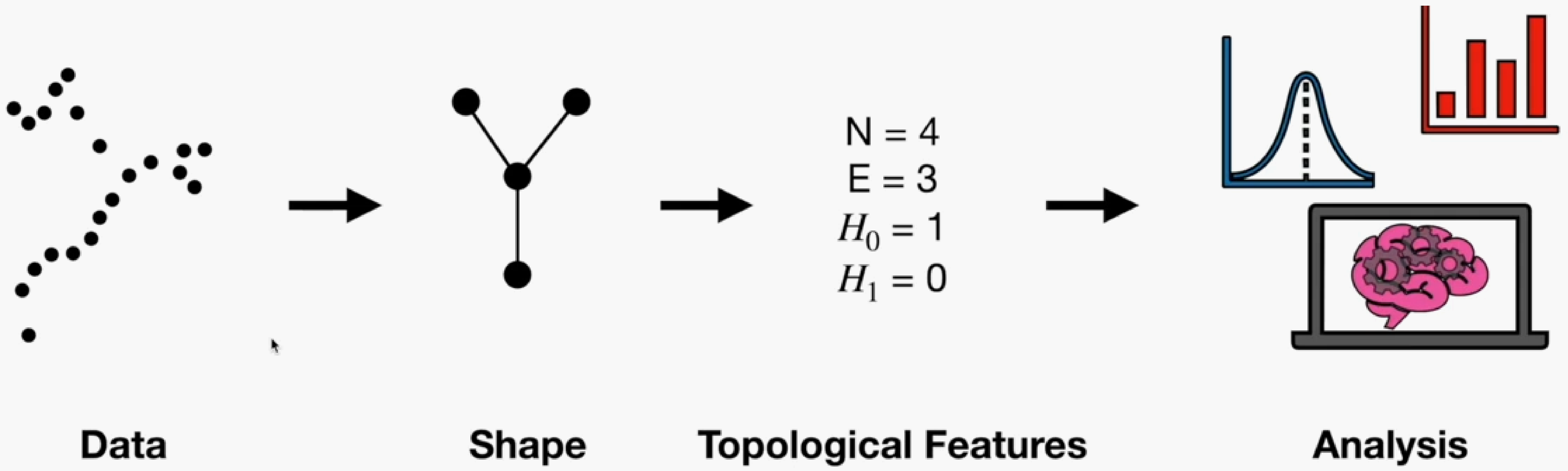




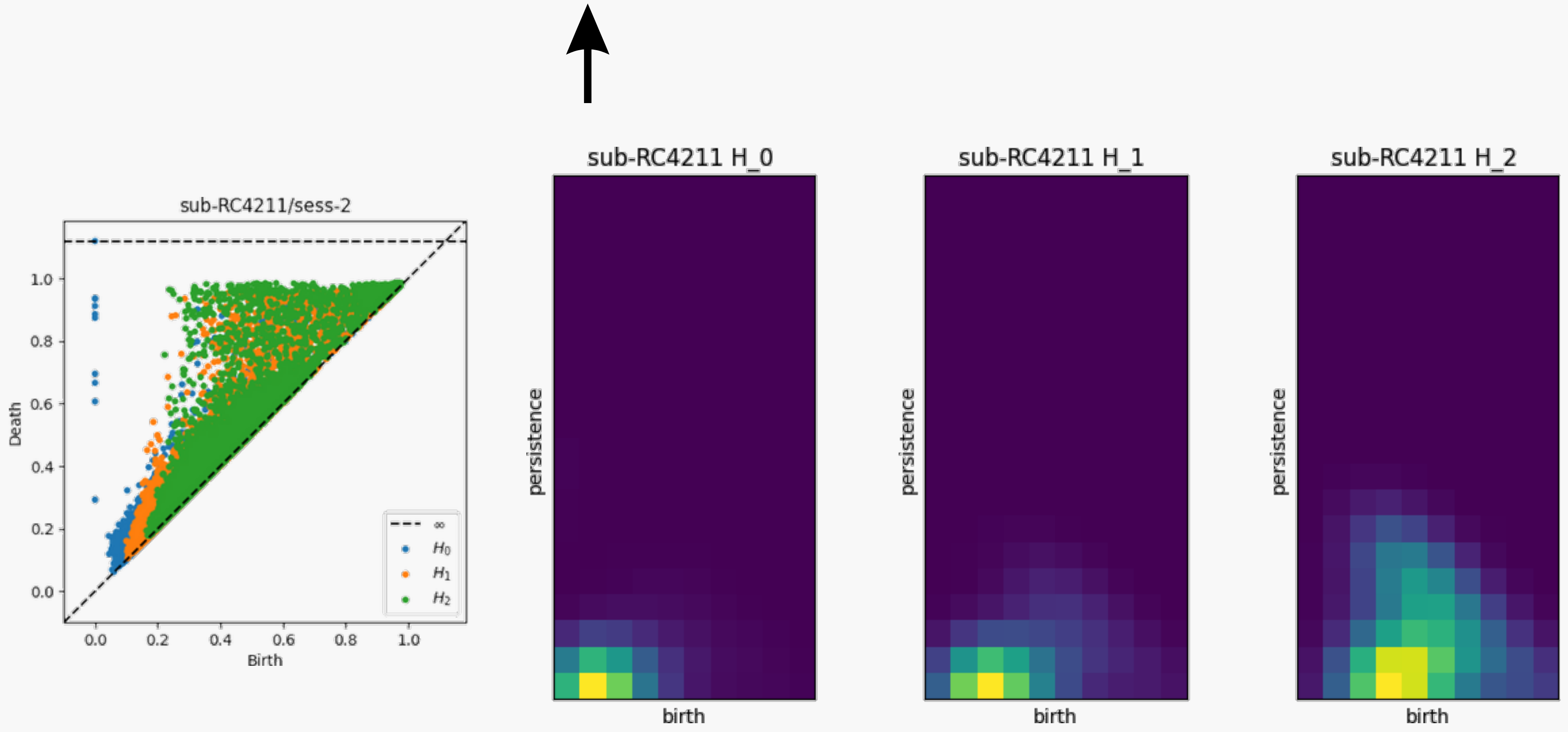


By examining the birth and death of topological features in this exemple, persistent homology provides a way to measure their persistence or robustness.

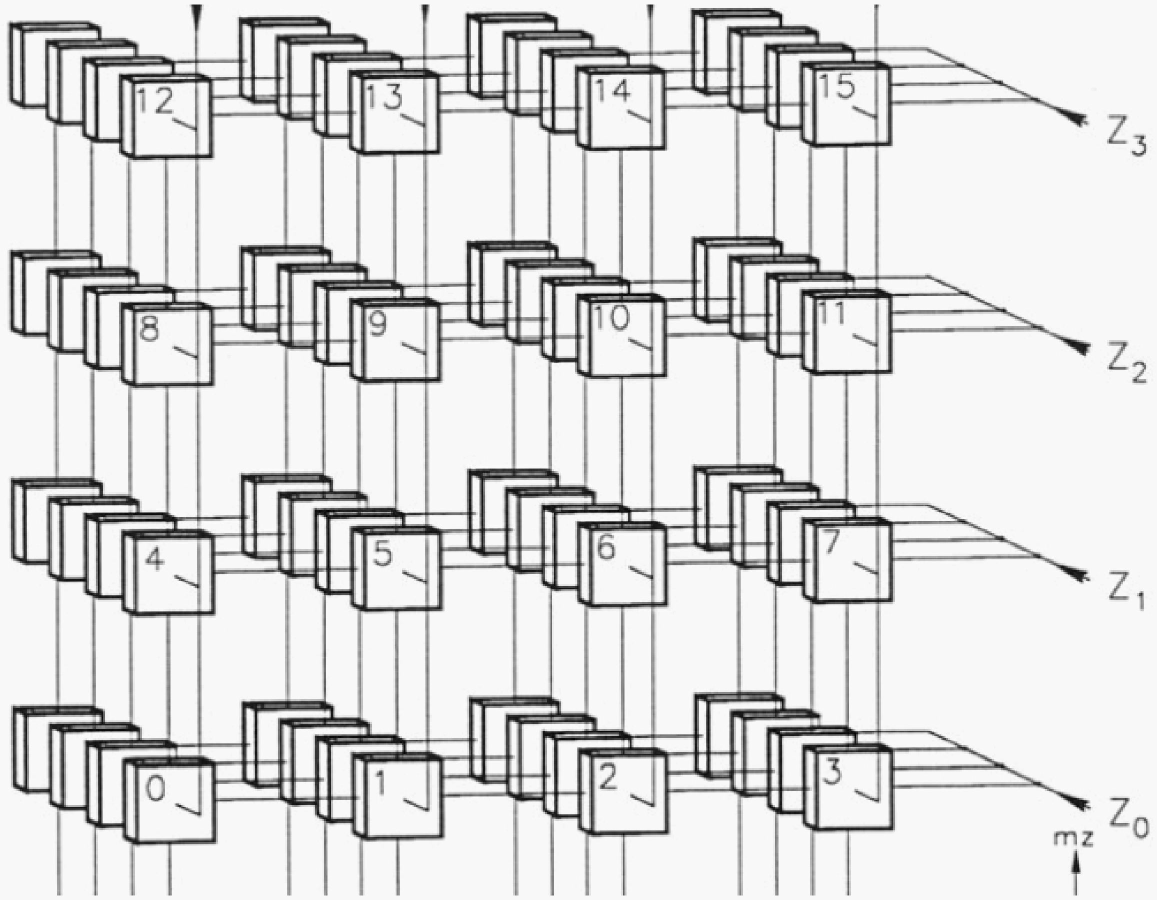
TDA Pipeline

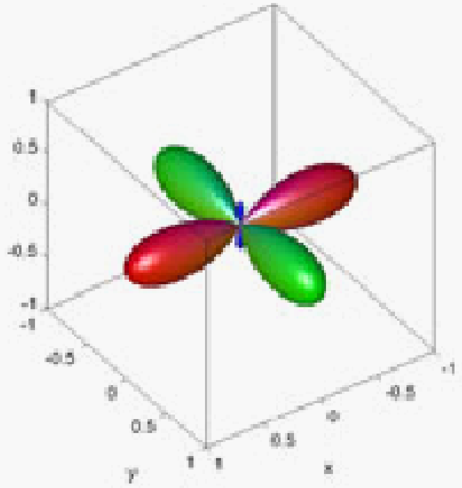


[1]F. Chazal and B. Michel, "An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists," arXiv:1710.04019 [cs, math, stat], Oct. 2017

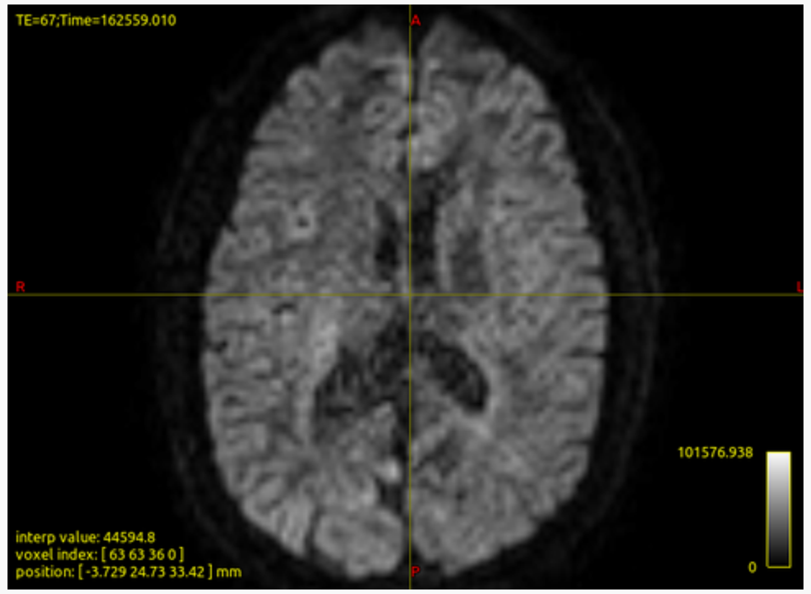


Why not use TDA on voxels?

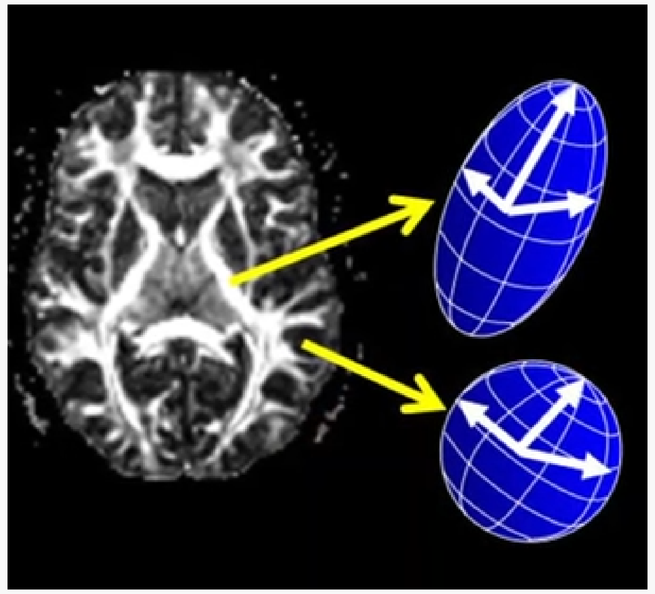




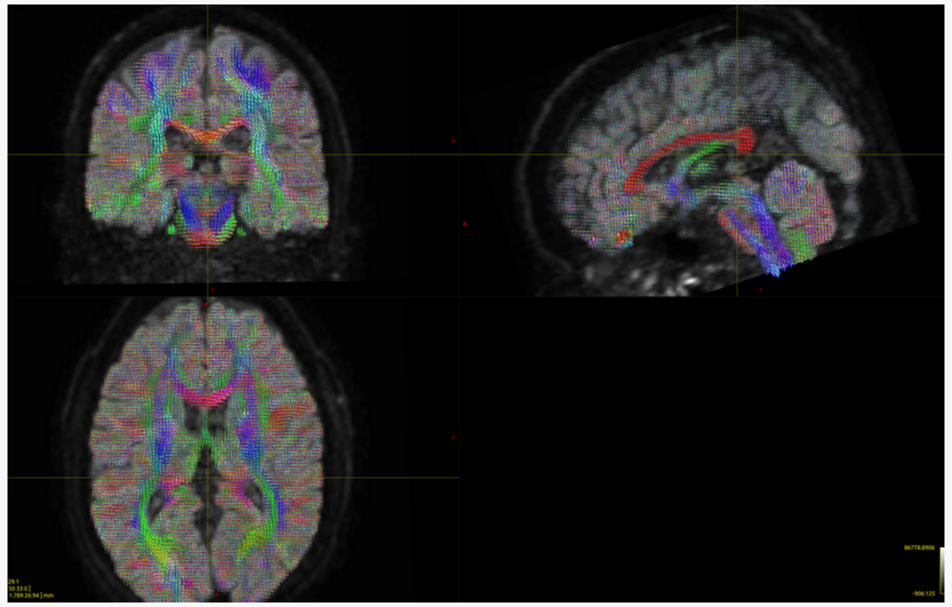
FODF : fiber orientation distribution function

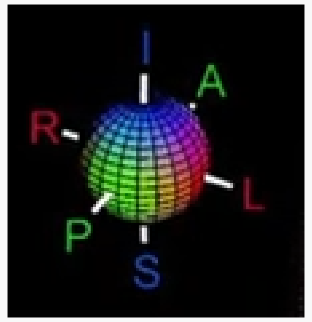


Preprocessed DWI →

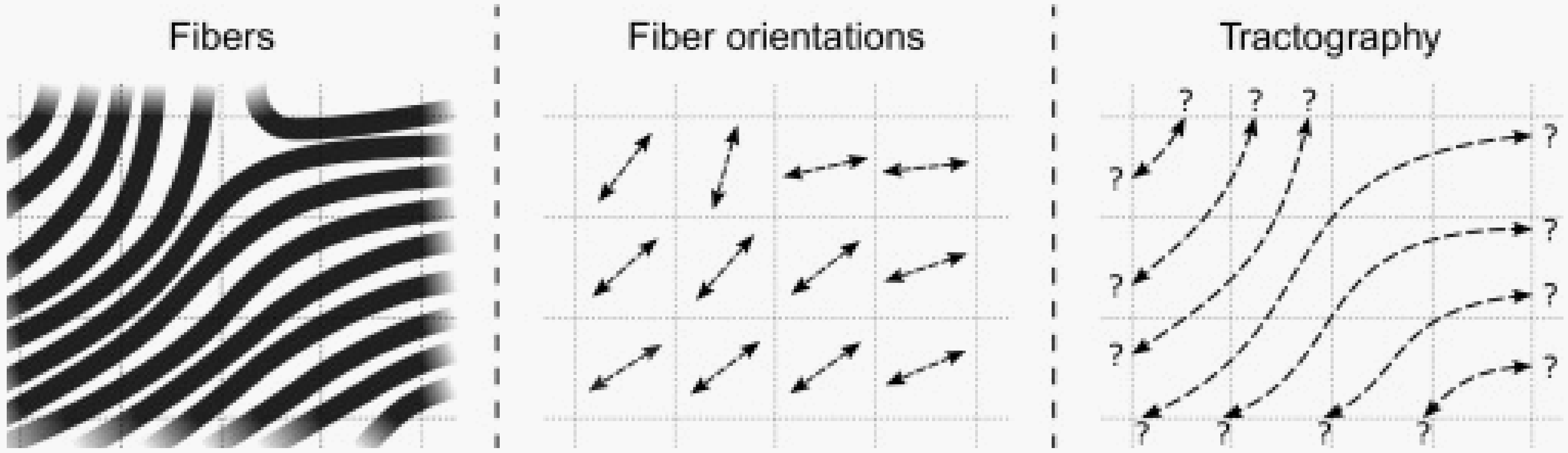


FODs estimation →

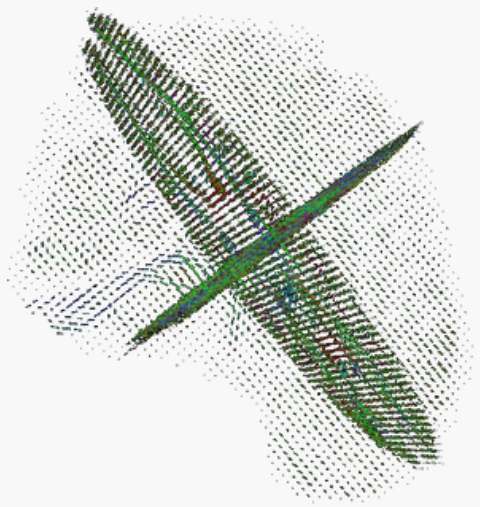


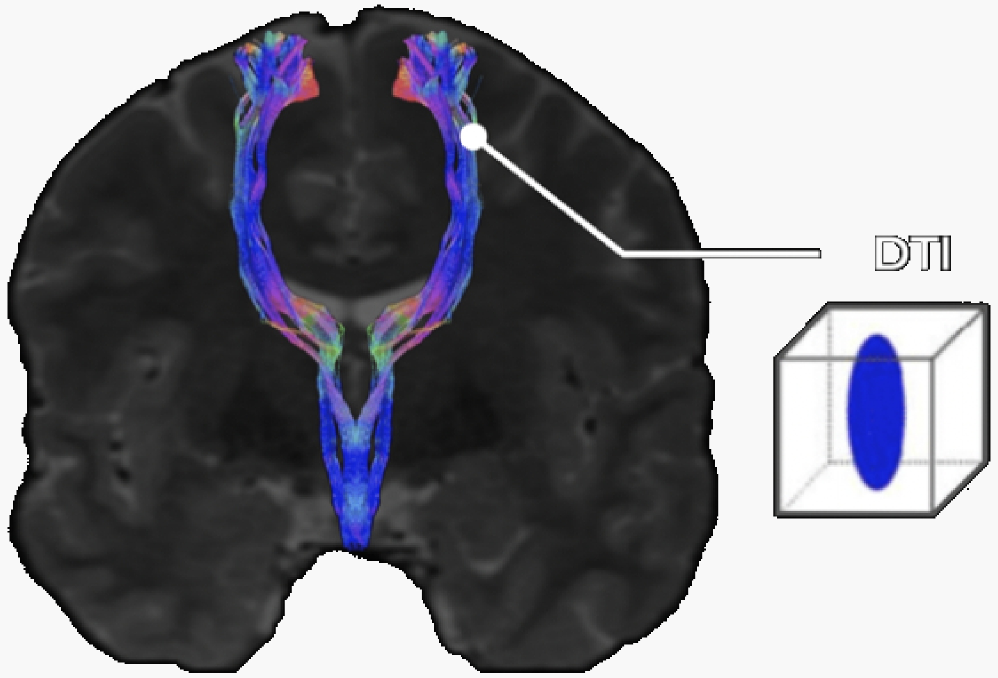


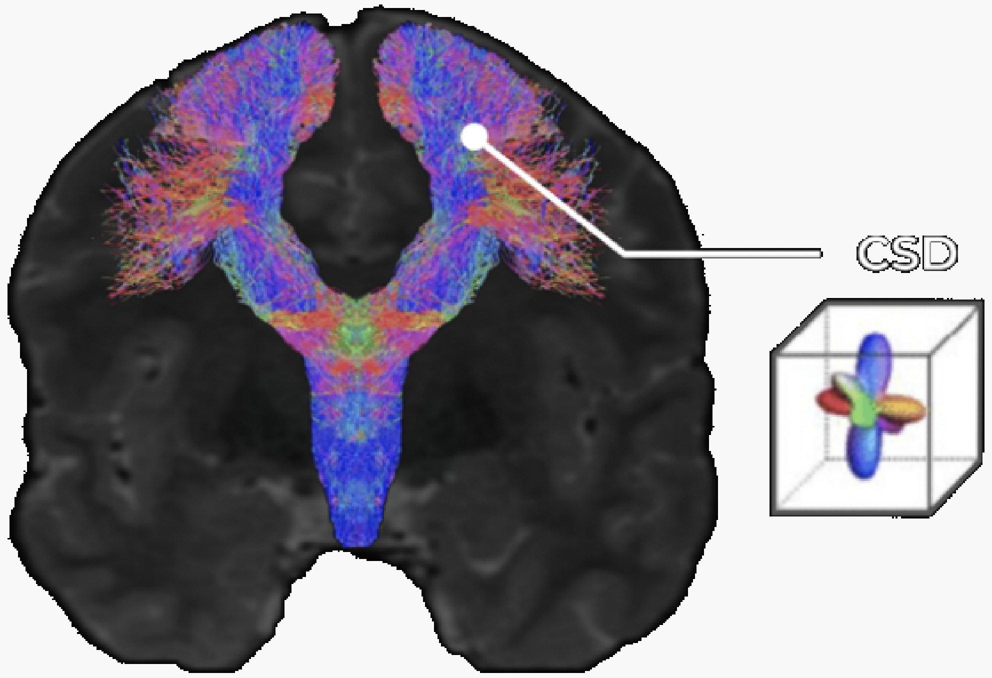
Fitting the model to the brain



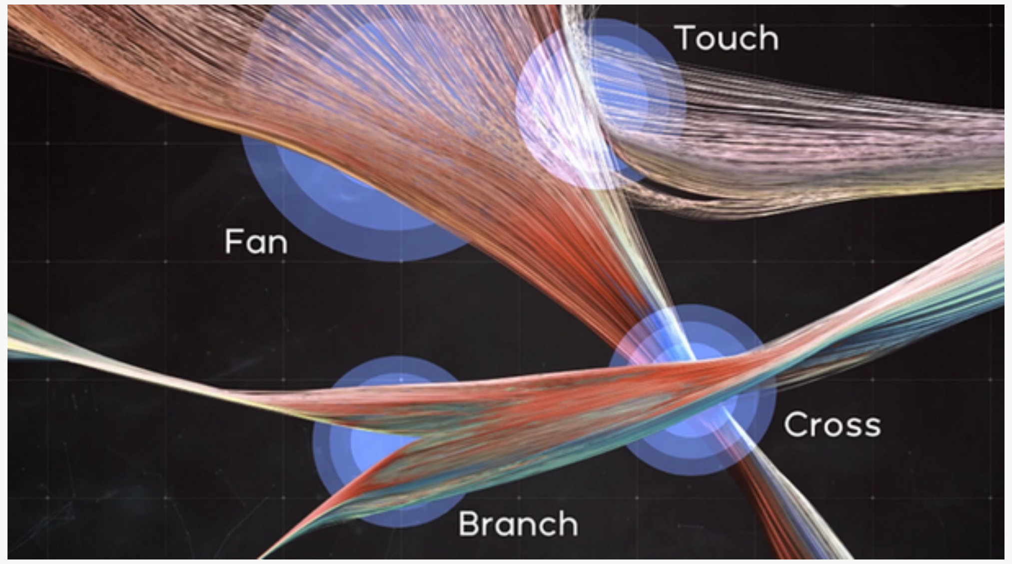
Constrained spherical deconvolution model vs DTI model







- DTI is a conventional tractography technique that is limited when modelling more complex structures of the brain
- CSD is a new technique that offers vastly greater detail when mapping complex brain network connections



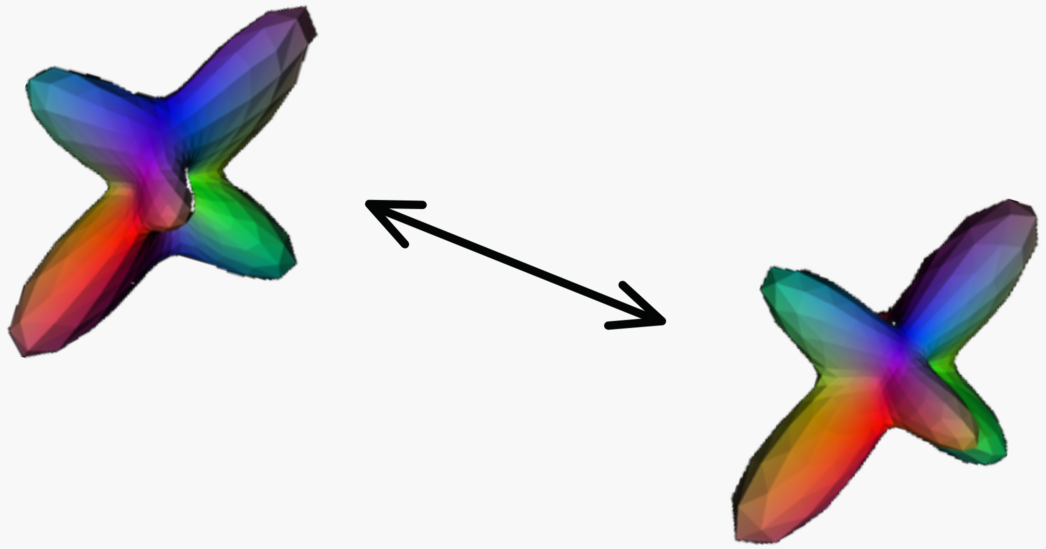
- Tractography : process of mapping the brain's white matter connections based on the evaluation of distribution of fibers in each voxel

Growth distance

Growth distances is defined for a pair of convex objects as a measure of how much each of the objects must be grown, outward from fixed seed points in their interiors, so that they just touch

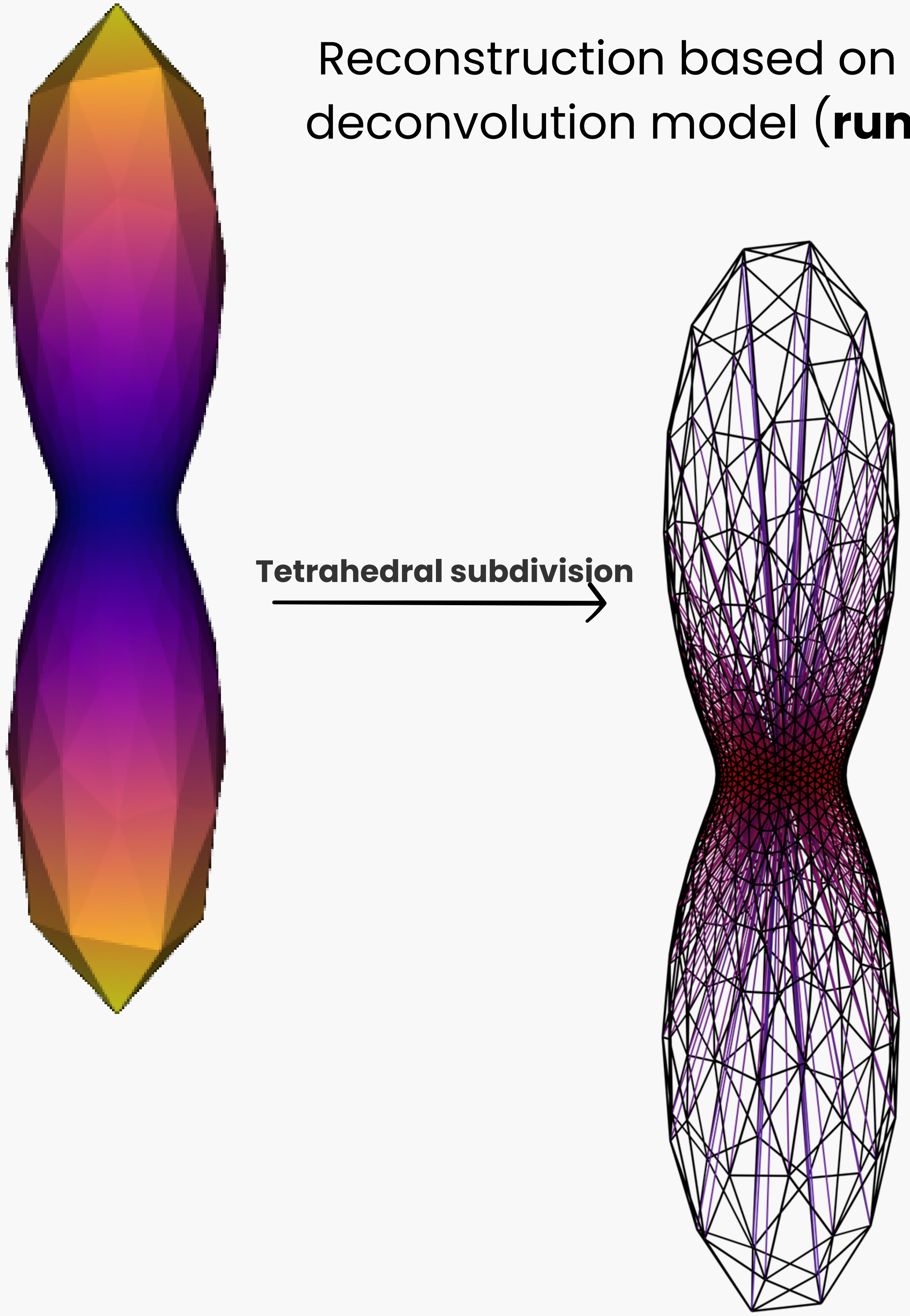
[3]C. J. Ong and E. G. Gilbert, "IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 12, NO. 6, DECEMBER 1996".

Growth distance between two odfs is how much we need to dilate each odf to make them touch



Growth distance of fodfs

Reconstruction based on a robust and unbiased spherical deconvolution model (**rumba-sd**) from DIPY



DIPY

Growth distance between 2 fodfs

```
# Retrieve the tetrahedrons for ODF centers (0, 0, 0) and (0, 0, 1)
tetrahedrons_A = odf_cartesian_faces_dict[(0, 0, 0)]
tetrahedrons_B = odf_cartesian_faces_dict[(0, 1,0)]

# Calculate the optimized sigma_star
optimized_sigma_star = calculate_optimized_sigma_star(tetrahedrons_A, tetrahedrons_B)

# Print or use the optimized_sigma_star as desired
print("Optimized sigma_star:", optimized_sigma_star)

#run time 8:56
```

```
Optimized sigma_star: 1.982345336731888
```

Computational complexity

1444 tetrahedrons per ODF
128*128*32 = 524288 voxels
524288 * 1444 = 757071872 tetrahedrons in total

757071872^2 calculations between 2 odfs

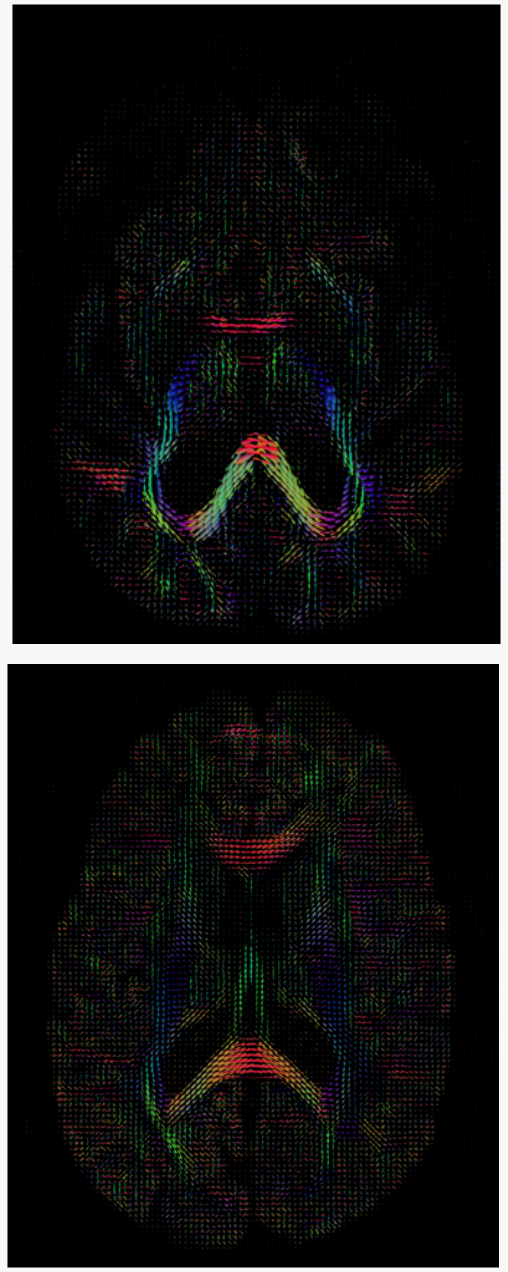
Processing Flow

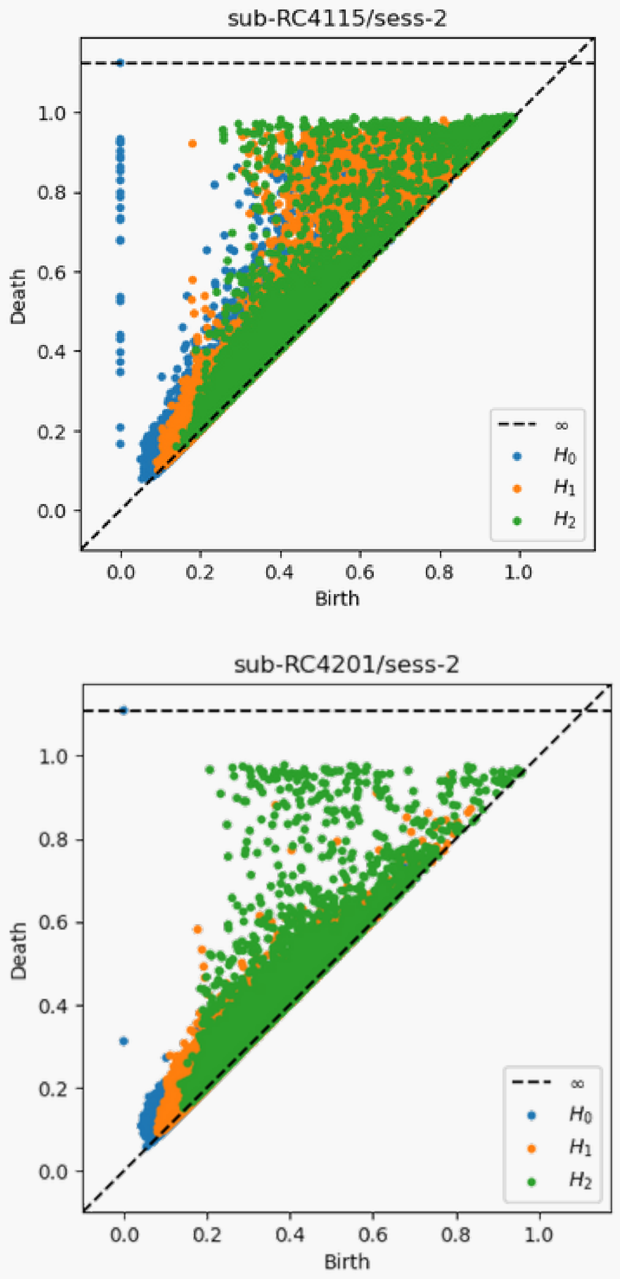
fOdfs reconstruction

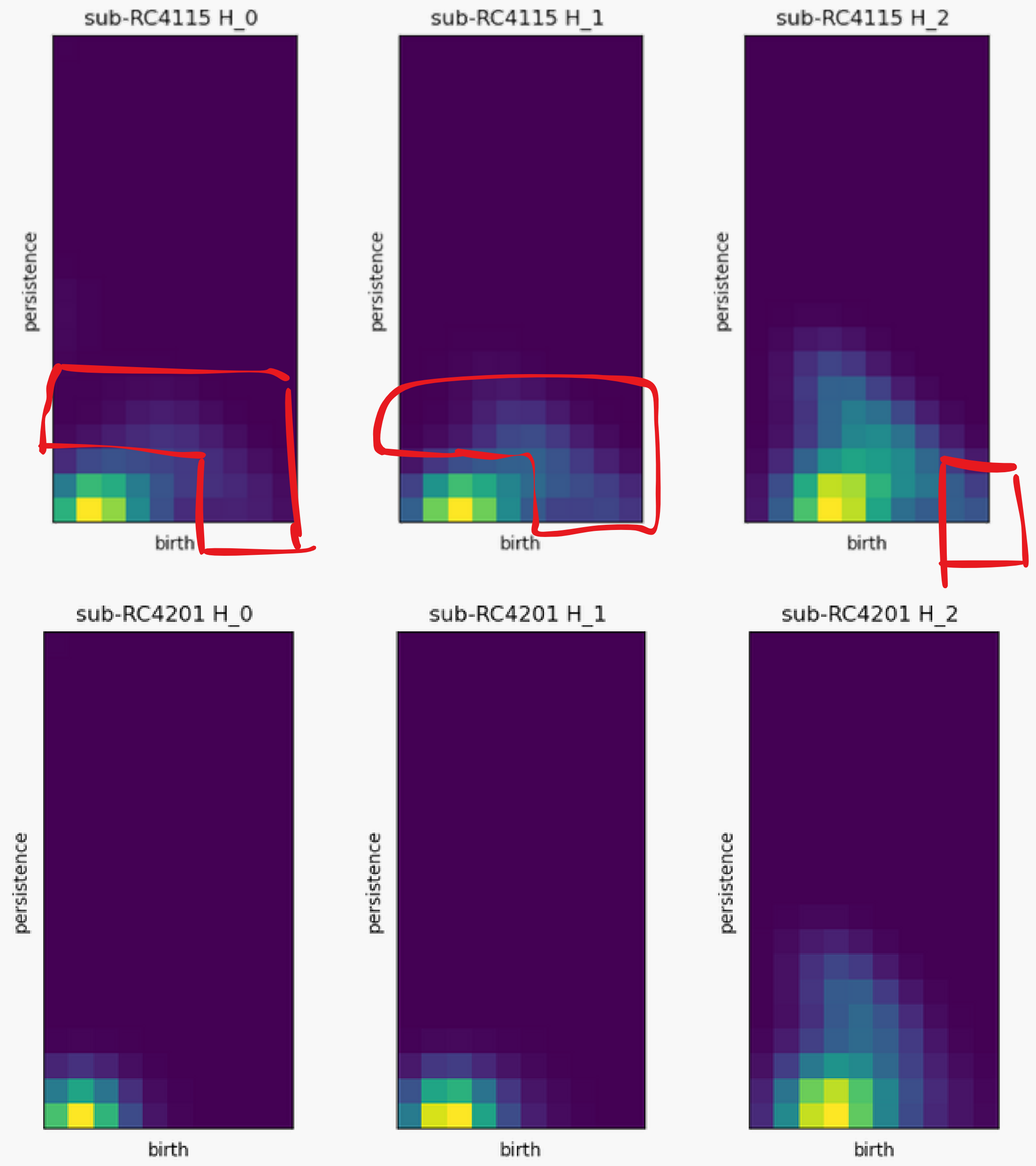
fodfs dilatation and construction of persistence diagrams

Conversion into persistence images

Classification



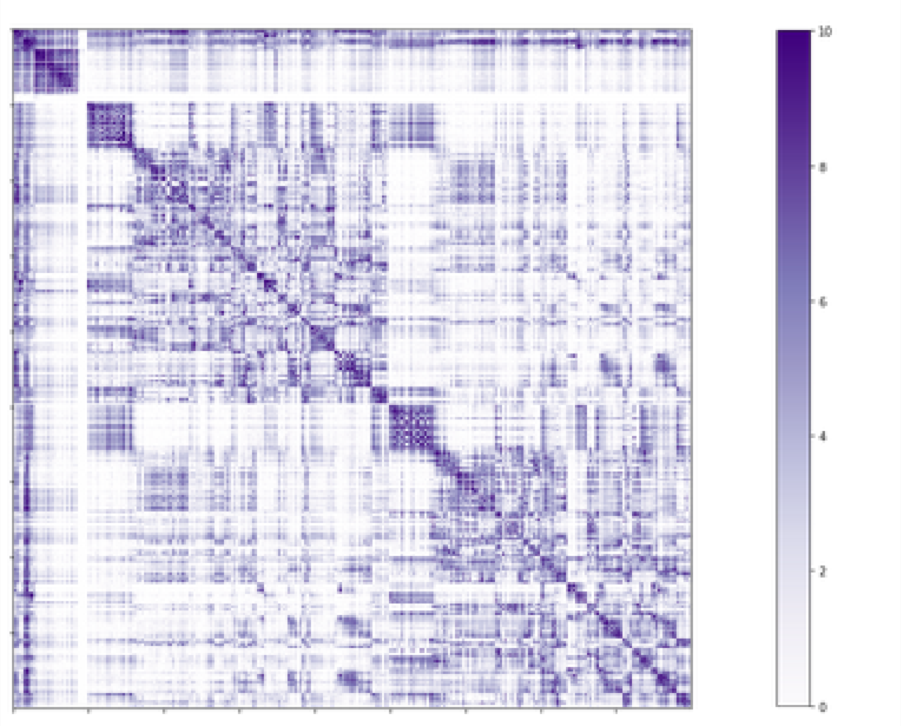




Healthy

Parkinsonian

Outlook



Structural Connectome

- TDA on connectivity matrices
- PPMI cohort (prodromal patients)
- FairparkII (de novo patients) / Predistim (advanced stage patients) cohorts

Thank you for listening :)