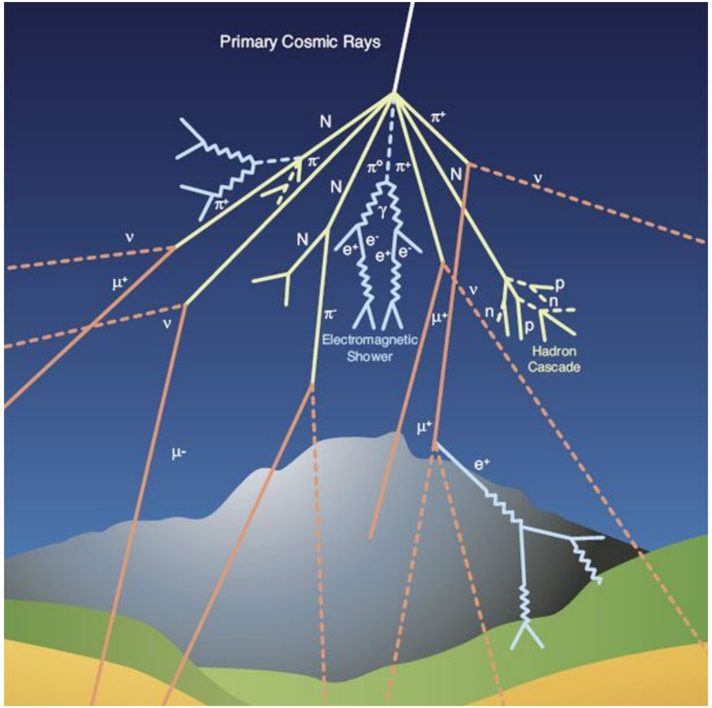
Cosmic Ray detection using clustering techniques

Sophie Carlier - Under the supervision of Jolan Lavoisier, Arsène Ferriere, Kumiko Kotera and Takashi Sako

GRAND

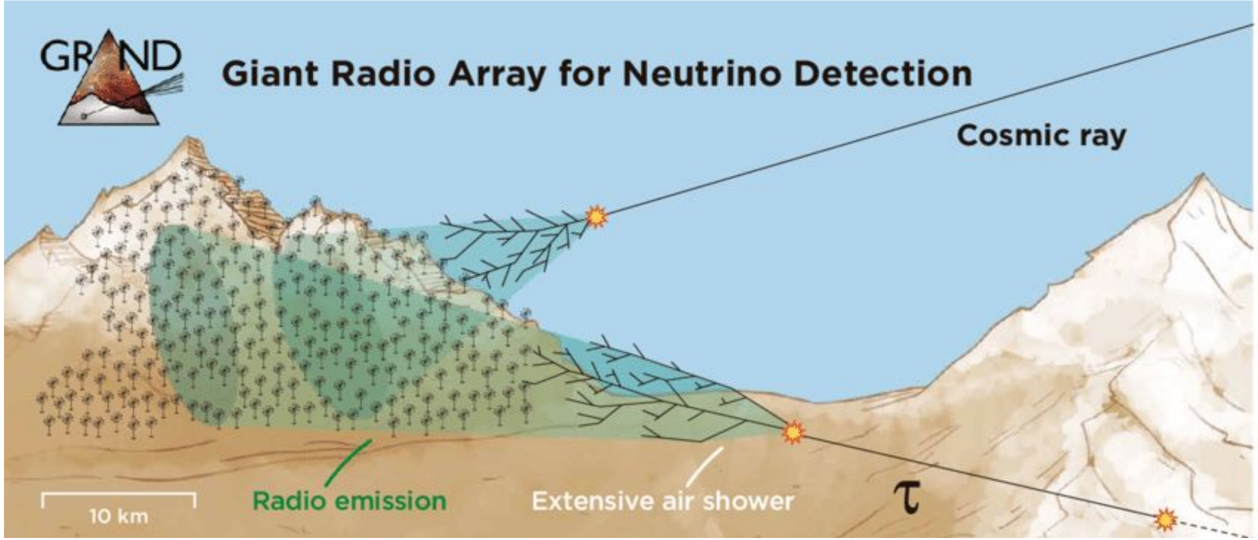
Giant Radio Array for Neutrino Detection : GRAND



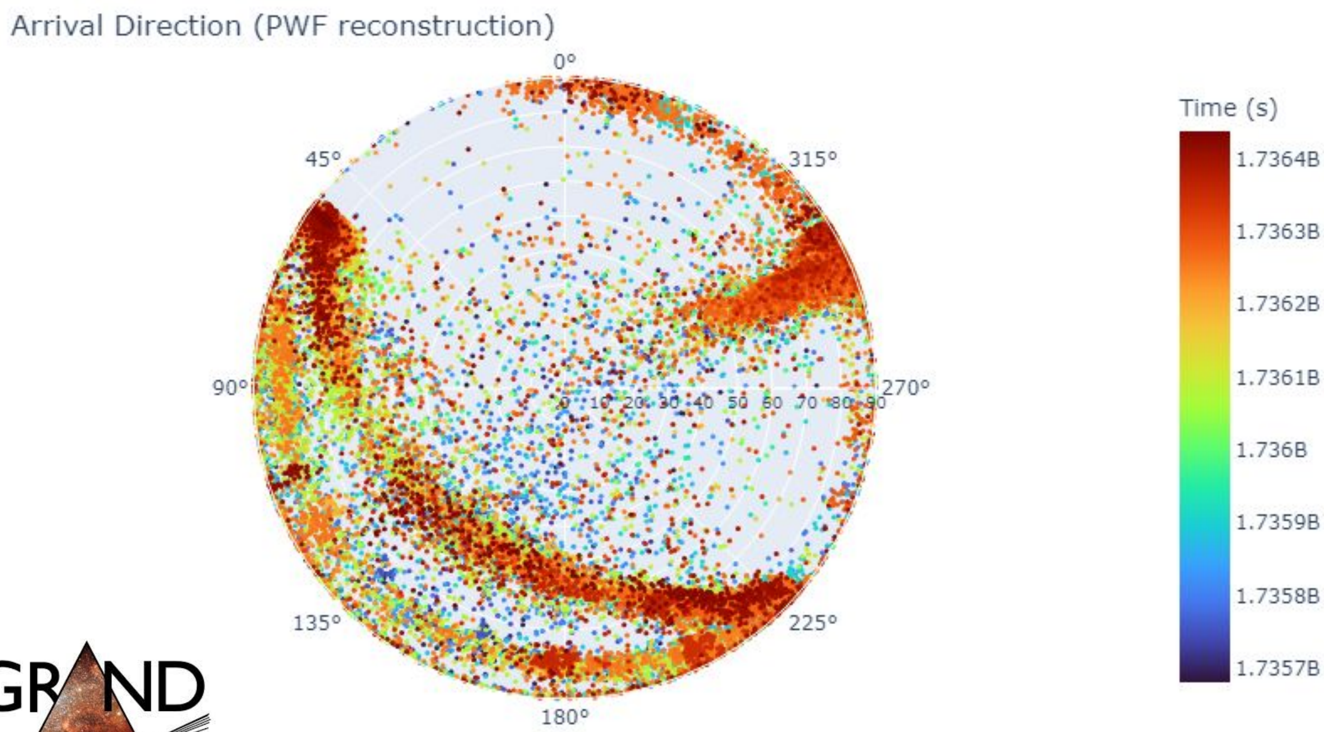
- Cosmic Rays enter the Earth’s atmosphere
- Collision with nitrogen and oxygen → creation of an air shower (cascade of secondary particles)

Giant Radio Array for Neutrino Detection : GRAND

- AIr shower → Electromagnetic wave → detection with a network of radio antennas
- GRAND : 300 antennas over 200 km2



January 2025 data

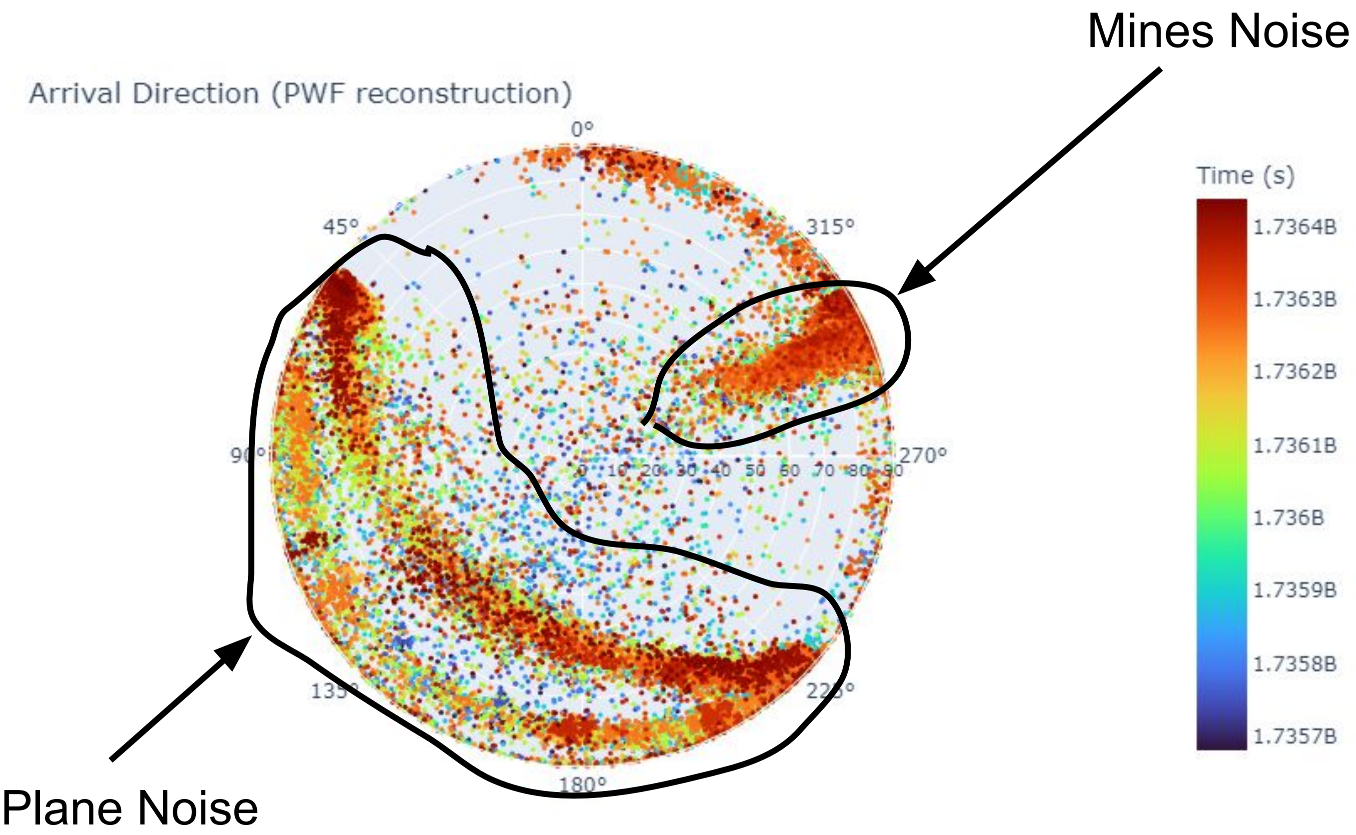


Three types of Noise :

- Galactic Noise
- Mines
- Planes

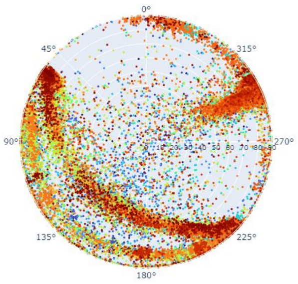
Goal : Use ML techniques to remove noise → identify candidates for Cosmic Rays

January 2025 data

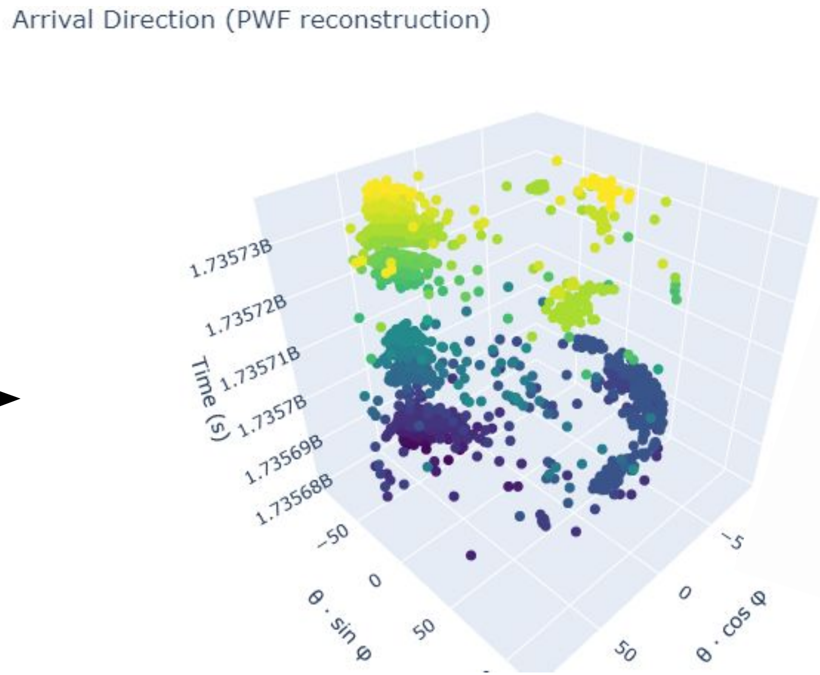


→ **Clustering** : detect patterns
→ work on mines and on planes individually

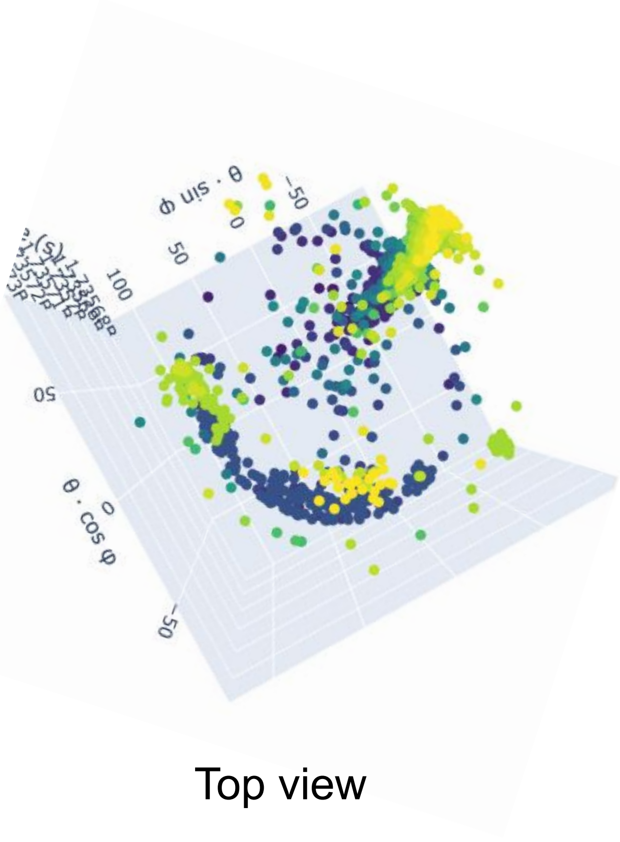
January 3D visualisation



3D



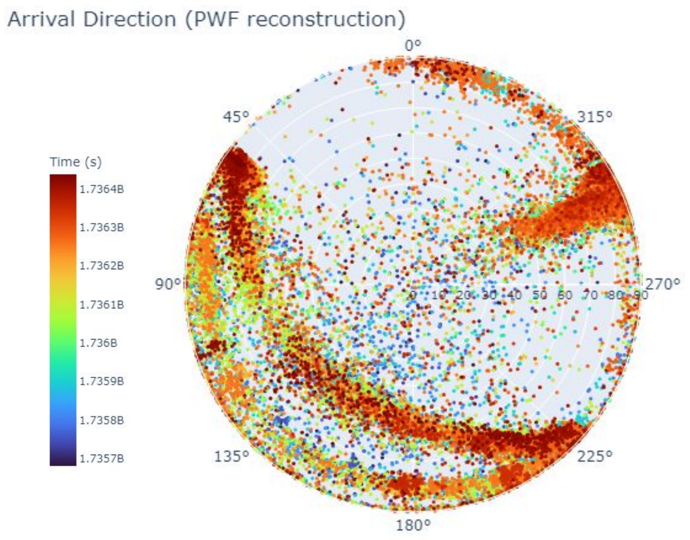
Side view



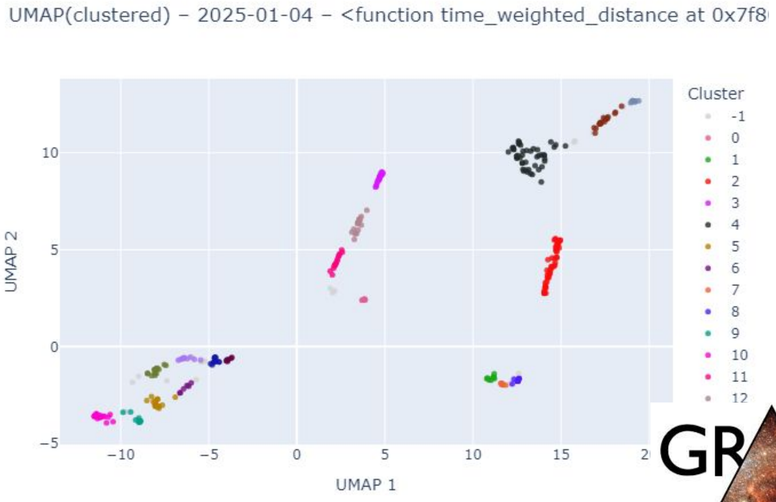
Top view

Step 1 : Uniform Manifold Approximation and Projection (UMAP)

- Dimensionality **reduction technique** : preserves local and global structure
- Projects the original 3D space [time, theta, phi] into a 2D space [Umap1, Umap2]
- Parametric UMAP : neural network → can learn and generalize to new data

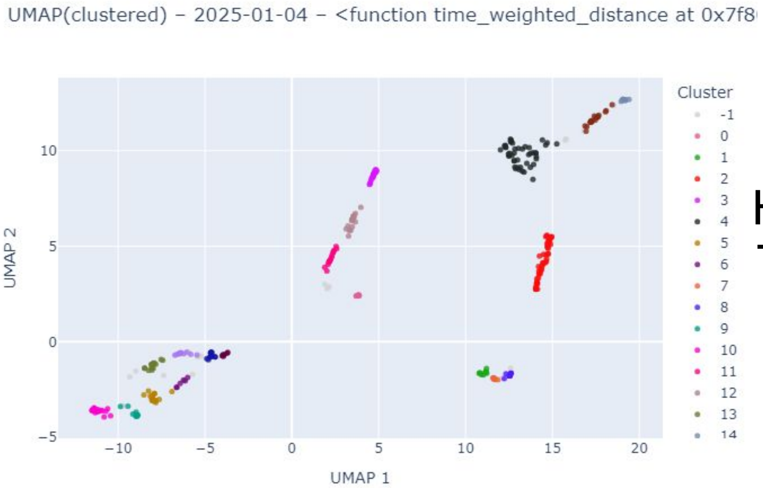


UMAP →

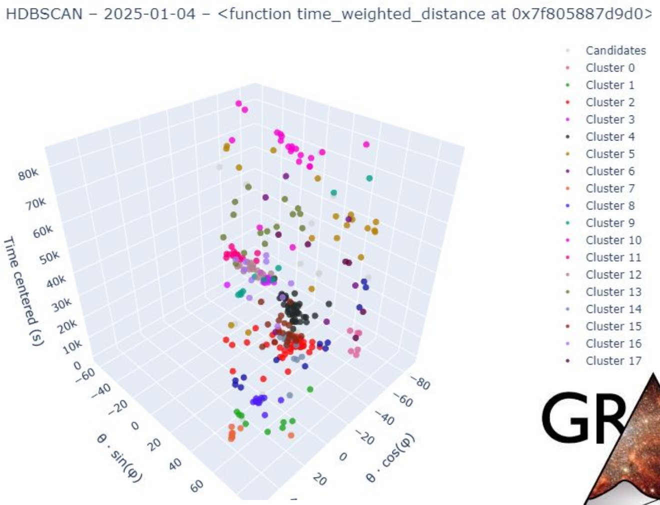


Step 2 : Hierarchical Density Based Spatial Clustering of Application with Noise (HDBSCAN)

- Density-based **clustering**
- ++ : Handles varying densities and automatically detects outliers (= potential candidates)



HDBSCAN

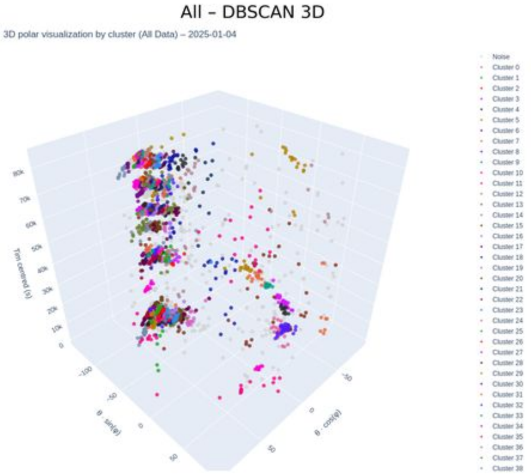


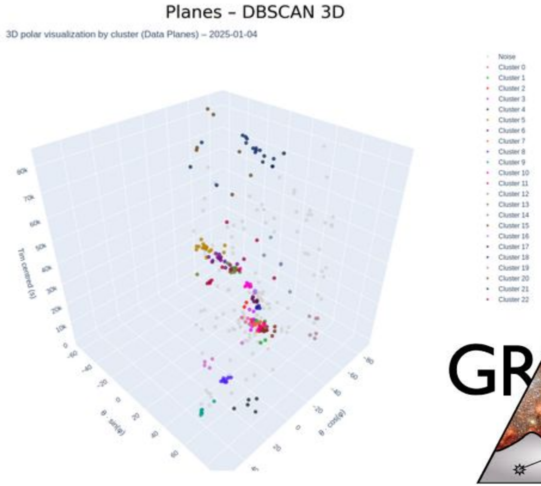
Optimisation of Clustering on Planes Data

1) **Selection of Planes Data**

→ 50° < **φ** < 230°

→ Goal : detect planes pattern (little Δt and high Δangular : Plane signals appear as **horizontal slices** in the data space)





Optimisation of Clustering on Planes Data

1) Selection of Planes Data
2) **Introduce custom UMAP metrics**

→ give more importance to time than to angular variables

$$a = (t_a, \theta_a, \phi_a), \quad b = (t_b, \theta_b, \phi_b)$$

$$\Delta t = |t_a - t_b|, \quad \Delta\theta = |\theta_a - \theta_b|, \quad \Delta\phi = |\phi_a - \phi_b|$$

1° linear metric $d_1(a,b) = 5 \cdot \Delta t + \Delta\theta + \Delta\phi$

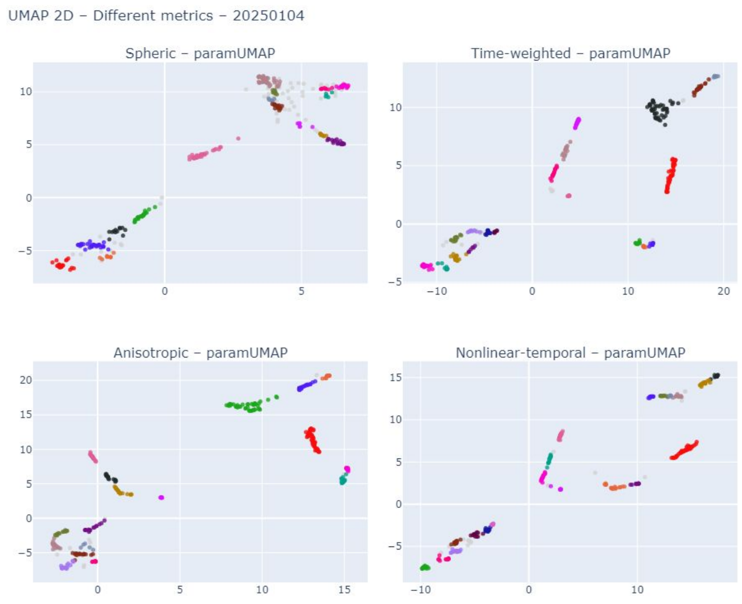
2° anisotropic metric $d_2(a,b) = \Delta t + 0.3 \cdot \sqrt{\Delta\theta^2 + \Delta\phi^2}$

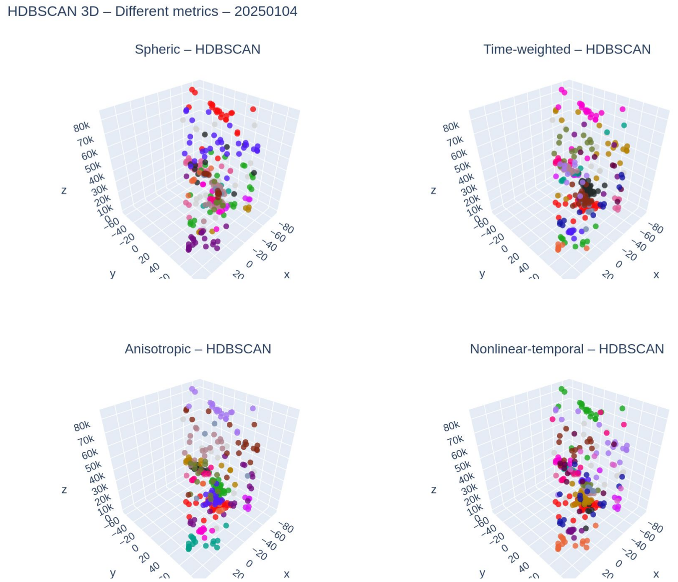
3° exponential metric $d_3(a,b) = \Delta t + e^{-1/\Delta t} \cdot (\Delta\theta + \Delta\phi)$

4° spherical metric $d(\mathbf{a},\mathbf{b}) = \alpha \cdot (\Delta t)^2 + \left[\arccos\left(\sin\theta_a \cdot \sin\theta_b \cdot \cos(\phi_a - \phi_b) + \cos\theta_a \cdot \cos\theta_b\right)\right]^2$

Optimisation of Clustering on Planes Data

1) Selection of Planes Data
2) Introduce custom UMAP metrics
3) **Implement parametric UMAP + HDBSCAN**





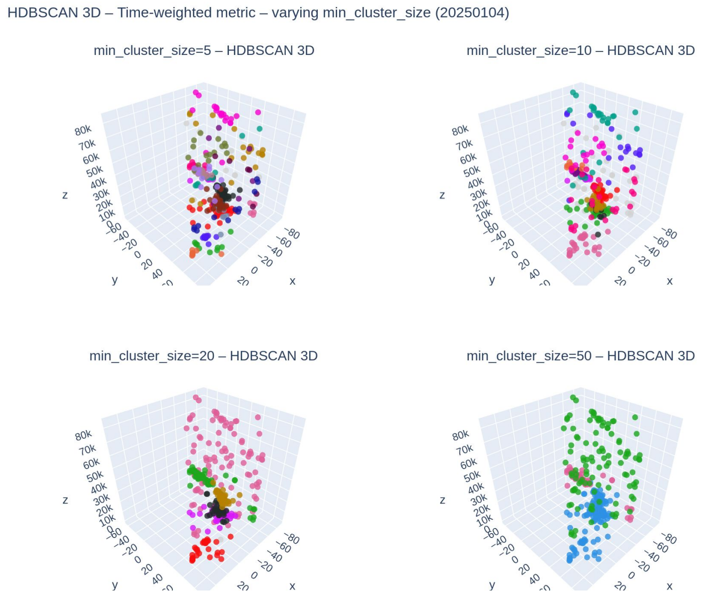
Optimisation of Clustering on Planes Data

1) Selection of Planes Data
2) Introduce custom UMAP metrics
3) Implement parametric UMAP + HDBSCAN
4) **Optimize different parameters**

- HDBSCAN

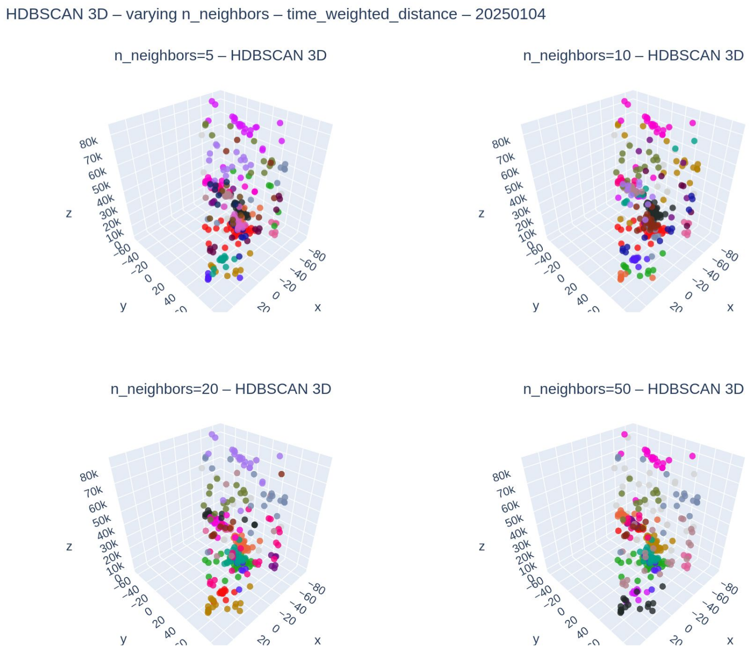
  test min_cluster_size : (5,10,20,50)

  → **min_cluster_size = 5**



Optimisation of Clustering on Planes Data

1) Selection of Planes Data
2) Introduce custom UMAP metrics
3) Implement parametric UMAP + HDBSCAN
4) **Optimize different parameters**



- HDBSCAN : *min_cluster_size = 5*

test n_neighbors : (5, 10, 20, 50)

→ **n_neighbors = 10**

Optimisation of Clustering on Planes Data

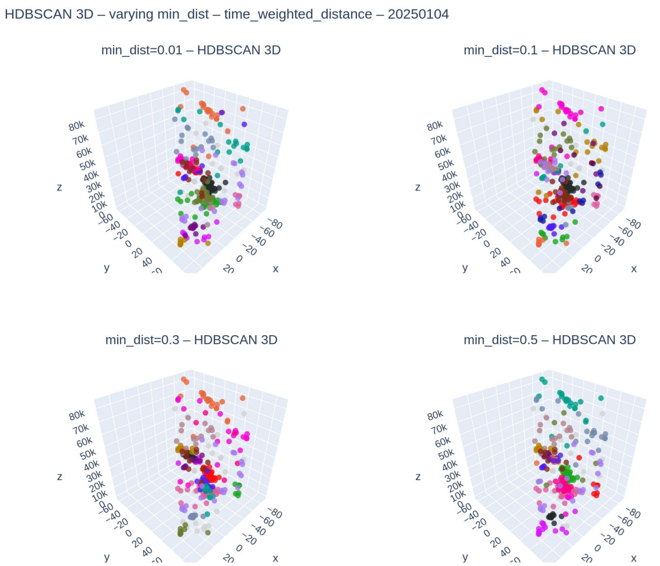
1) Selection of Planes Data
2) Introduce custom UMAP metrics
3) Implement parametric UMAP + HDBSCAN
4) **Optimize different parameters**

- HDBSCAN : *min_cluster_size = 5*

  n_neighbors = 10

- UMAP : min_dist (0.01 , 0.01, 0.3 ,0.5)

  → **min_dist = 0.1**



And now?

Done :

- Optimized clustering algorithm for planes

  → Clear separation of clusters and outliers

Next steps and improvements :

- Apply the same methodology to mines data and **merge both datasets**

- Switch to a new representation space

  → Current analysis uses SWF: (θ, φ, t)

  → Next step: move to 4D space: **(x, y, z, t)** using PWF reconstruction