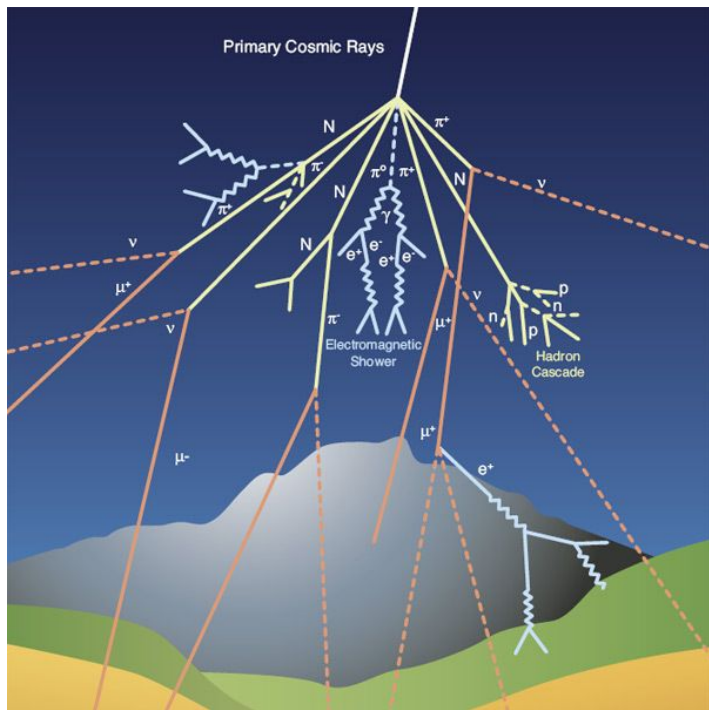


Cosmic Ray detection using clustering techniques

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



Giant Radio Array for Neutrino Detection : GRAND

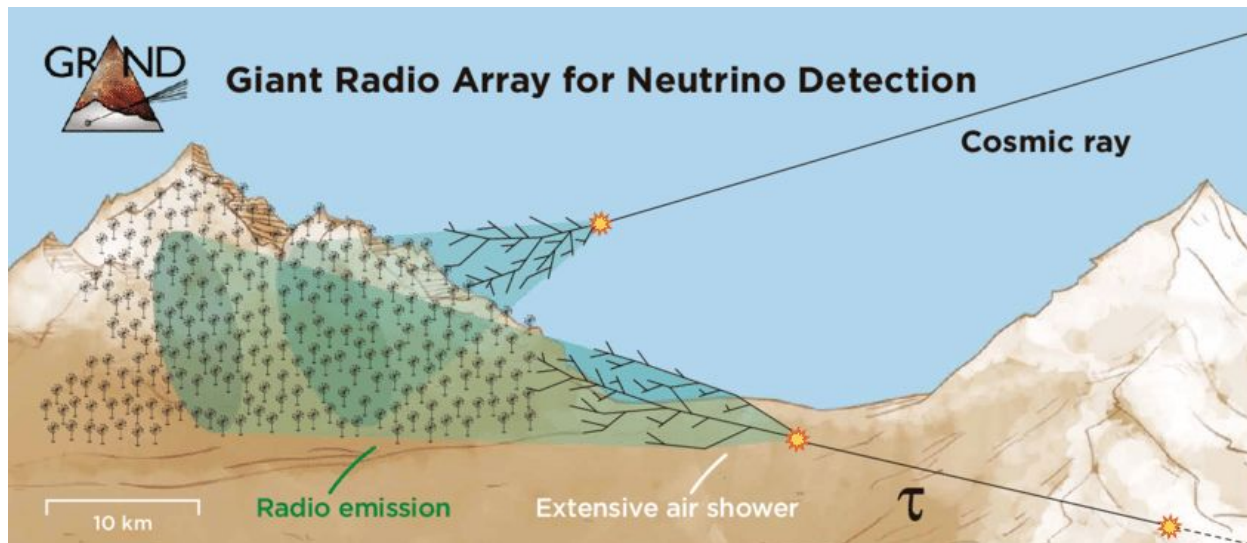


- Cosmic Rays enter the Earth's atmosphere
- Collision with nitrogen and oxygen \rightarrow creation of an air shower (cascade of secondary particles)



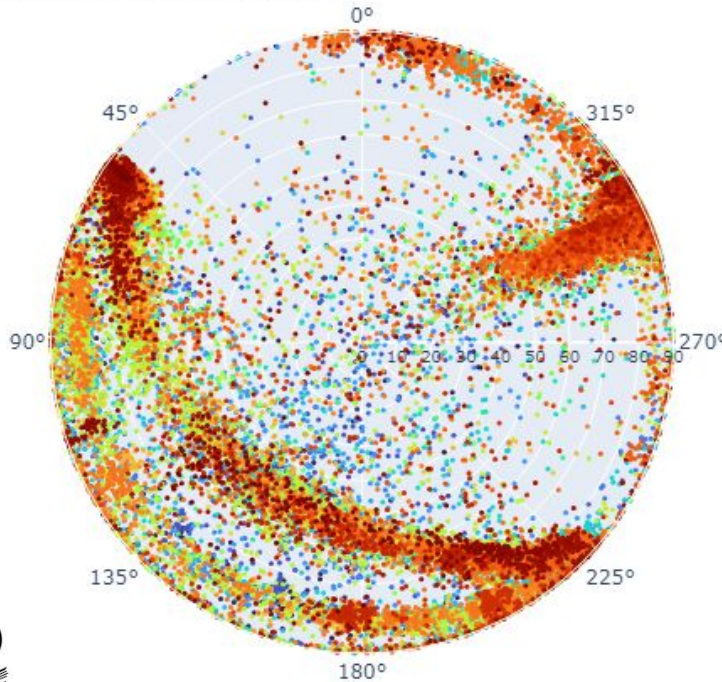
Giant Radio Array for Neutrino Detection : GRAND

- Air shower \rightarrow Electromagnetic wave \rightarrow detection with a network of radio antennas
- GRAND : 300 antennas over 200 km²



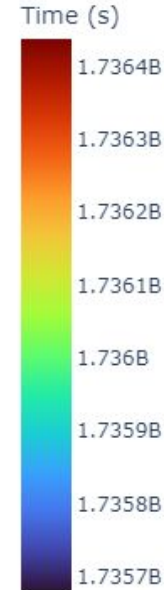
January 2025 data

Arrival Direction (PWF reconstruction)



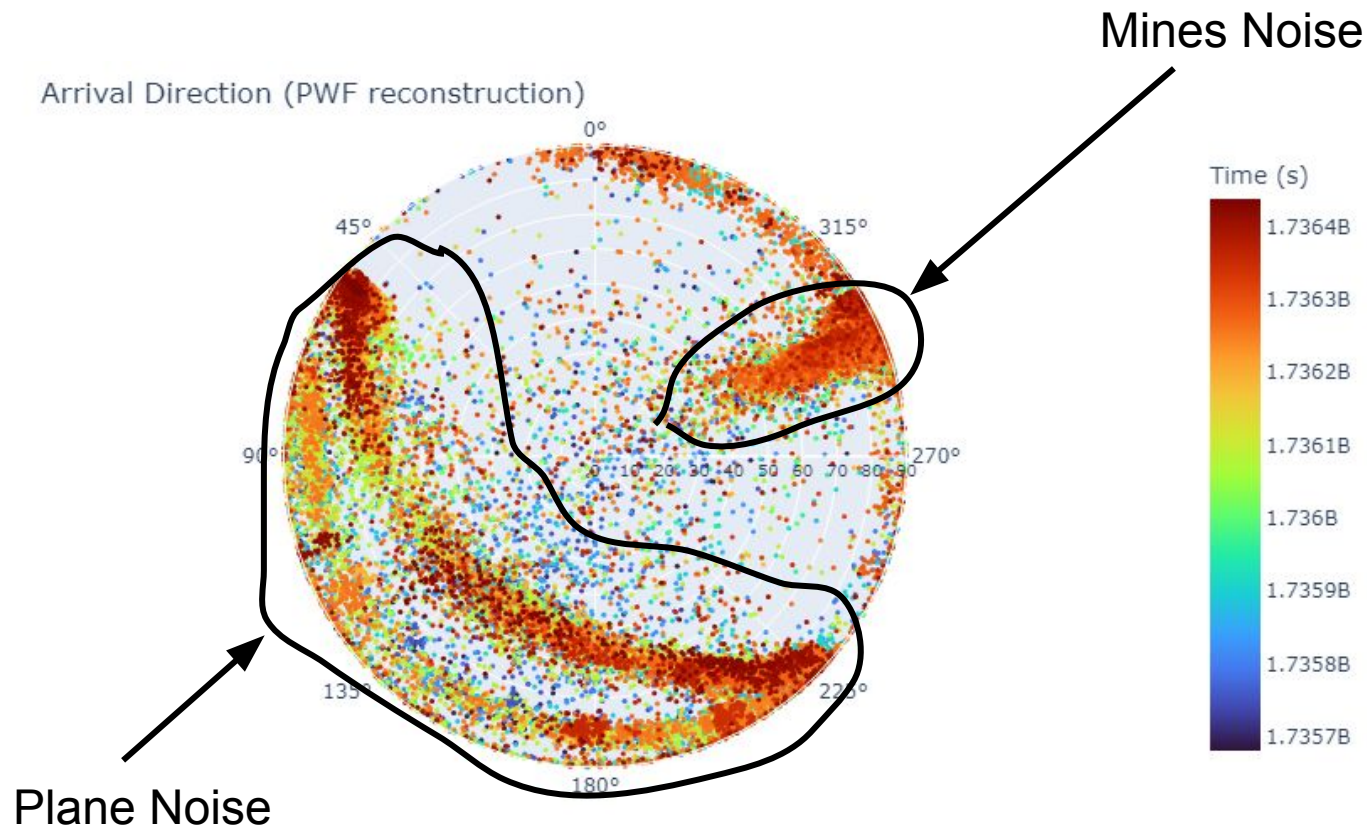
Three types of Noise :

- Galactic Noise
- Mines
- Planes



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

January 2025 data

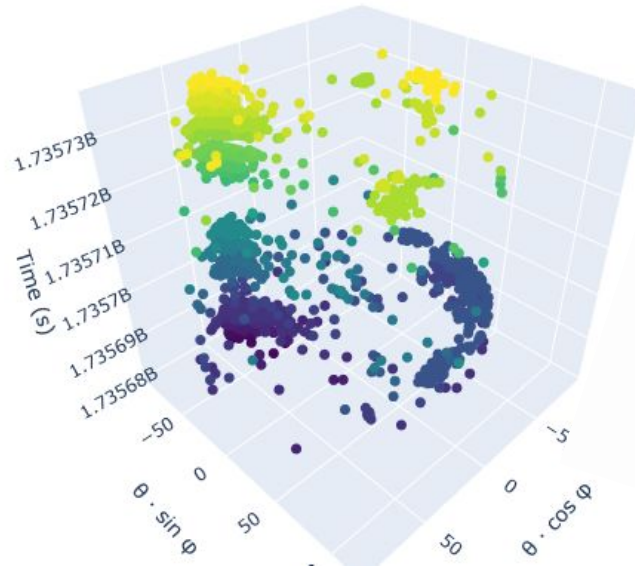
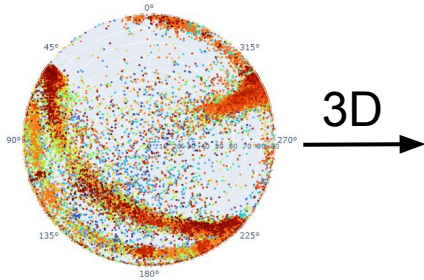


→ **Clustering** :
detect patterns
→ work on
mines and on
planes
individually

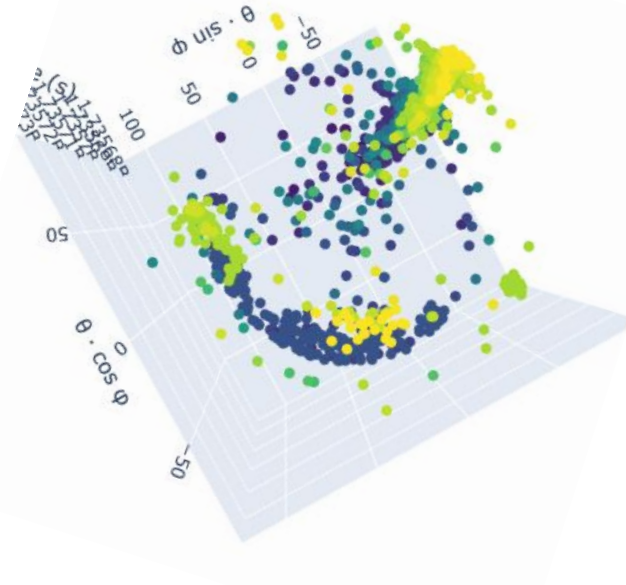


January 3D visualisation

Arrival Direction (PWF reconstruction)



Side view

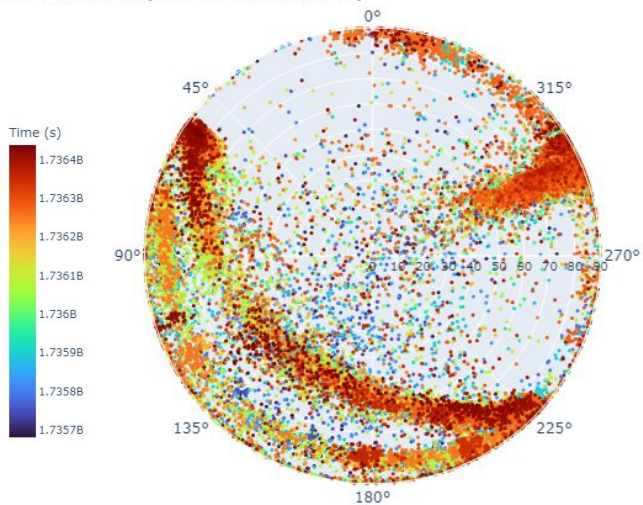


Top view

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

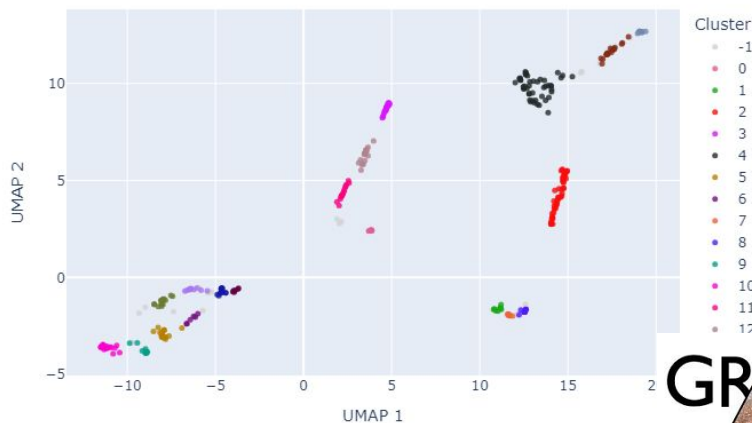
Arrival Direction (PWF reconstruction)



UMAP



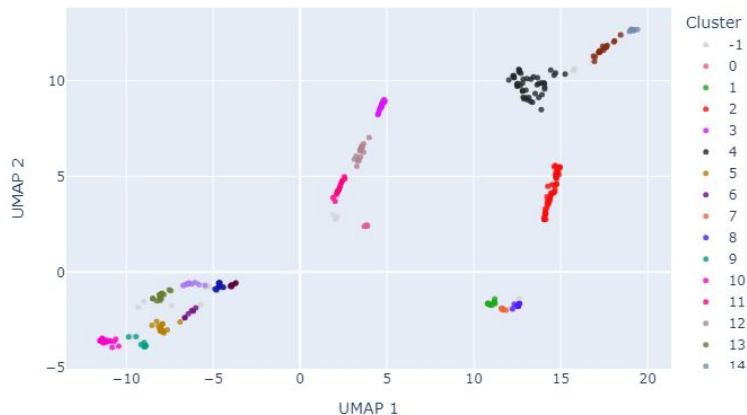
UMAP(clustered) - 2025-01-04 - <function time_weighted_distance at 0x7f8



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

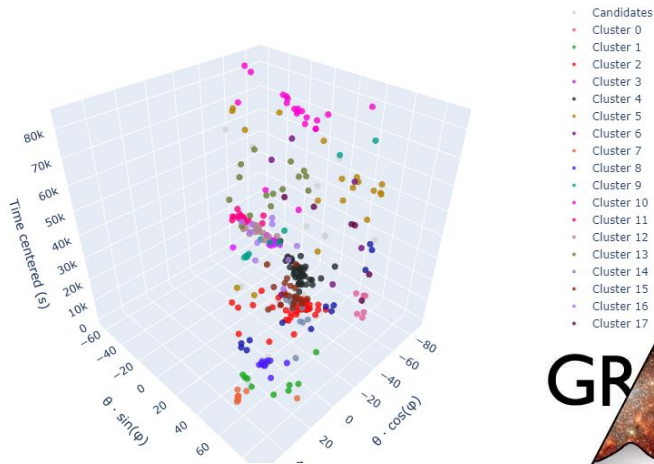
- Density-based **clustering**
- ++ : Handles varying densities and automatically detects outliers (= potential candidates)

UMAP(clustered) – 2025-01-04 – <function time_weighted_distance at 0x7f8



HDBSCAN

HDBSCAN – 2025-01-04 – <function time_weighted_distance at 0x7f805887d9d0>



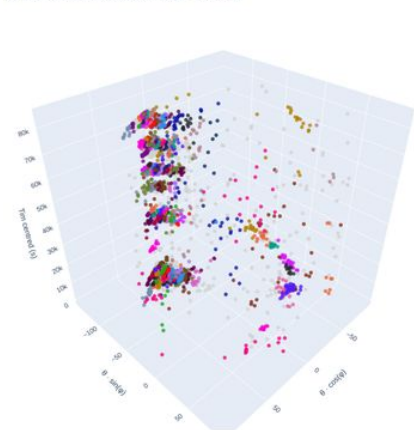
Optimisation of Clustering on Planes Data

1) Selection of Planes Data

→ $50^\circ < \varphi < 230^\circ$

→ Goal : detect planes pattern (little Δt and high $\Delta \text{angular}$: Plane signals appear as **horizontal slices** in the data space)

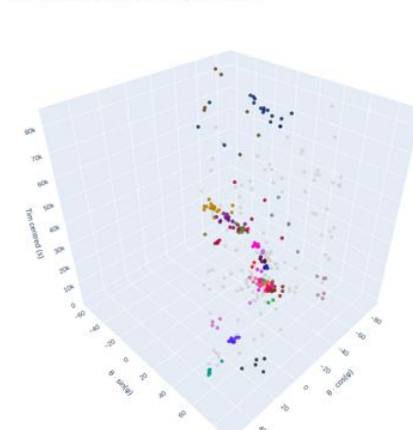
All - DBSCAN 3D
3D polar visualization by cluster (All Data) - 2025-01-04



- Noise
- Cluster 0
- Cluster 1
- Cluster 2
- Cluster 3
- Cluster 4
- Cluster 5
- Cluster 6
- Cluster 7
- Cluster 8
- Cluster 9
- Cluster 10
- Cluster 11
- Cluster 12
- Cluster 13
- Cluster 14
- Cluster 15
- Cluster 16
- Cluster 17
- Cluster 18
- Cluster 19
- Cluster 20
- Cluster 21
- Cluster 22
- Cluster 23
- Cluster 24
- Cluster 25
- Cluster 26
- Cluster 27
- Cluster 28
- Cluster 29
- Cluster 30
- Cluster 31
- Cluster 32
- Cluster 33
- Cluster 34
- Cluster 35
- Cluster 36
- Cluster 37
- Cluster 38
- Cluster 39



Planes - DBSCAN 3D
3D polar visualization by cluster (Data Planes) - 2025-01-04



- Noise
- Cluster 0
- Cluster 1
- Cluster 2
- Cluster 3
- Cluster 4
- Cluster 5
- Cluster 6
- Cluster 7
- Cluster 8
- Cluster 9
- Cluster 10
- Cluster 11
- Cluster 12
- Cluster 13
- Cluster 14
- Cluster 15
- Cluster 16
- Cluster 17
- Cluster 18
- Cluster 19
- Cluster 20
- Cluster 21
- Cluster 22



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

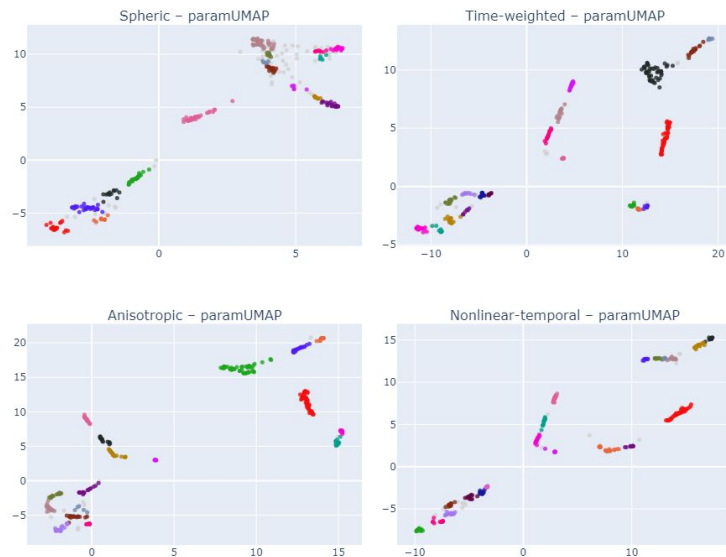
4° spherical metric $d(\mathbf{a}, \mathbf{b}) = \alpha \cdot (\Delta t)^2 + [\arccos(\sin \theta_a \cdot \sin \theta_b \cdot \cos(\phi_a - \phi_b) + \cos \theta_a \cdot \cos \theta_b)]^2$

Optimisation of Clustering on Planes Data

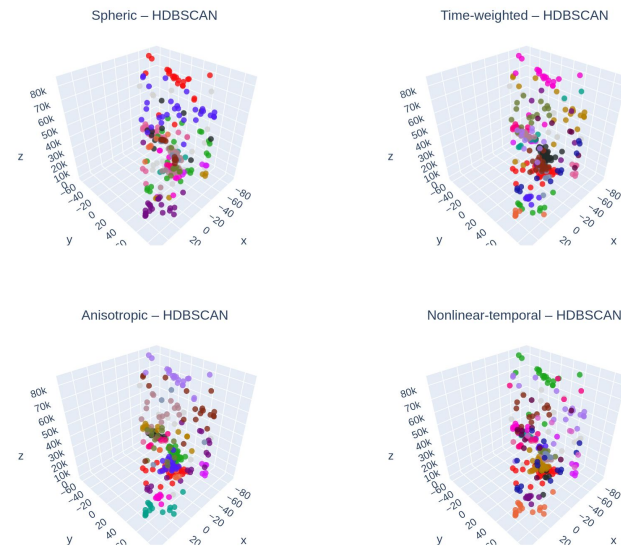


- 1) Selection of Planes Data
- 2) Introduce custom UMAP metrics
- 3) **Implement parametric UMAP + HDBSCAN**

UMAP 2D - Different metrics - 20250104



HDBSCAN 3D - Different metrics - 20250104



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**

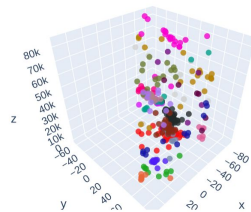
- HDBSCAN

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

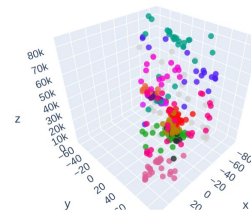
→ **min_cluster_size = 5**

HDBSCAN 3D – Time-weighted metric – varying min_cluster_size (20250104)

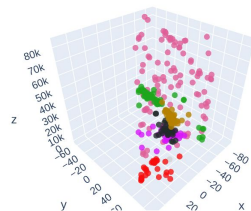
min_cluster_size=5 – HDBSCAN 3D



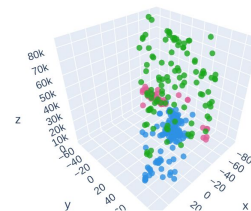
min_cluster_size=10 – HDBSCAN 3D



min_cluster_size=20 – HDBSCAN 3D



min_cluster_size=50 – HDBSCAN 3D



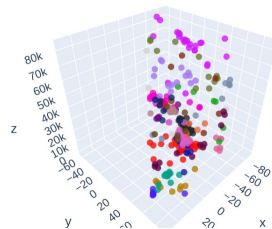
Optimisation of Clustering on Planes Data



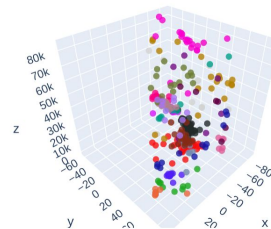
- 1) Selection of Planes Data
- 2) Introduce custom UMAP metrics
- 3) Implement parametric UMAP + HDBSCAN
- 4) **Optimize different parameters**

HDBSCAN 3D – varying `n_neighbors` – `time_weighted_distance` – 20250104

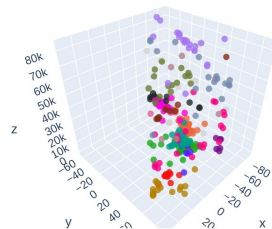
`n_neighbors=5` – HDBSCAN 3D



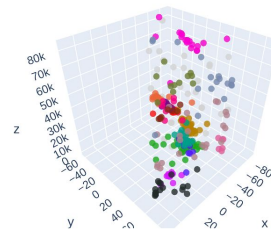
`n_neighbors=10` – HDBSCAN 3D



`n_neighbors=20` – HDBSCAN 3D



`n_neighbors=50` – HDBSCAN 3D



- HDBSCAN : *min_cluster_size* = 5

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

→ **`n_neighbors` = 10**

Optimisation of Clustering on Planes Data

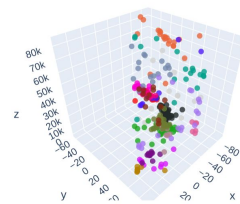


- 1) Selection of Planes Data
- 2) Introduce custom UMAP metrics
- 3) Implement parametric UMAP + HDBSCAN
- 4) **Optimize different parameters**

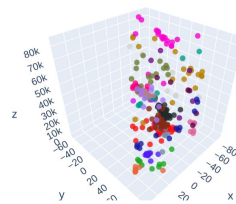
- HDBSCAN : $min_cluster_size = 5$
 $n_neighbors = 10$
- UMAP : min_dist (0.01 , 0.01, 0.3 ,0.5)
→ **$min_dist = 0.1$**

HDBSCAN 3D – varying min_dist – time_weighted_distance – 20250104

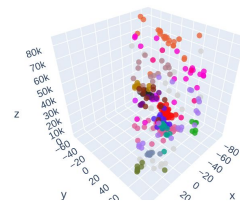
$min_dist=0.01$ – HDBSCAN 3D



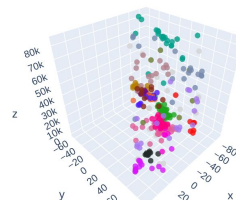
$min_dist=0.1$ – HDBSCAN 3D



$min_dist=0.3$ – HDBSCAN 3D



$min_dist=0.5$ – HDBSCAN 3D



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: **$(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{t})$** using PWF reconstruction

