ANOMALY DETECTION USING UNSUPERVISED ALGORITHM FOR PRODUCTION OF RADIO-ISOTOPE

Charly LASSALLE

05/11/2020

ARRONAX Activities

- A tool to produce radionuclides for research in nuclear medicine
  - Imaging: β+ radioelements for PET (ex: $^{82}Sr/^{82}Rb$, $^{44m/44}Sc$, ^{52}Fe, ^{64}Cu ...)
  - Therapy: α immunotherapy (^{211}At), ⍰$^{-}$ radioelements : ^{67}Cu, ^{47}Sc

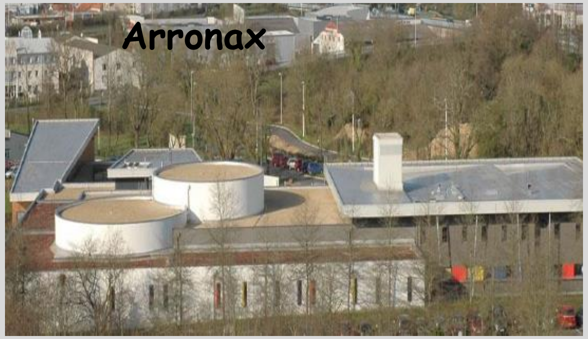
- A tool for radiochimistry & radiobiology research
  - specifically alpha radiolysis of water (eg nuclear waste storage)
  - radiobiology

- A tool for physics research
  - Particularly studies of material under irradiation
  - Development of detection system
  - Measurements of nuclear data

- A tool for training and education
  - University of Nantes
  - École des mines of Nantes (IMT Atlantique)
  - CHU (academic hospital) of Nantes
  - Permanent and dedicated trainings

- An industrial production site for medical needs
  - Standards:
    - ISO 9001 (Quality)
    - GMP (Good manufacturing practice)
    - APUI (internal usage pharmacy): on going

Arronax is a public interest group formed by national and regional institutes:



^{82}Sr production example

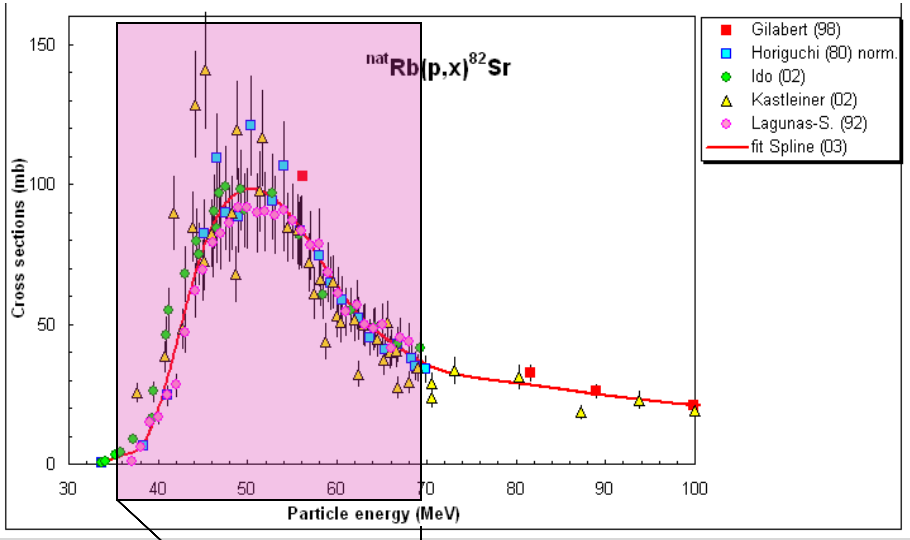
Reaction and Cross section

- Production of ^{82}Sr is obtained via:

  ^{nat}Rb + p → ^{82}Sr + x

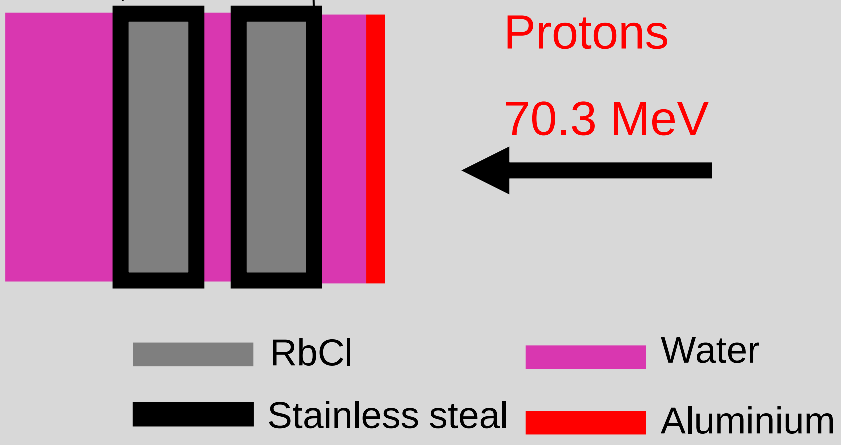
- Decay of ^{82}Sr (EC, 25.34d) gives ^{82}Rb, used in cardiology

Energy range of interest 40 MeV-70 MeV



2 small encapsulated targets

→Increase cooling capability
→limitate melting of RbCl



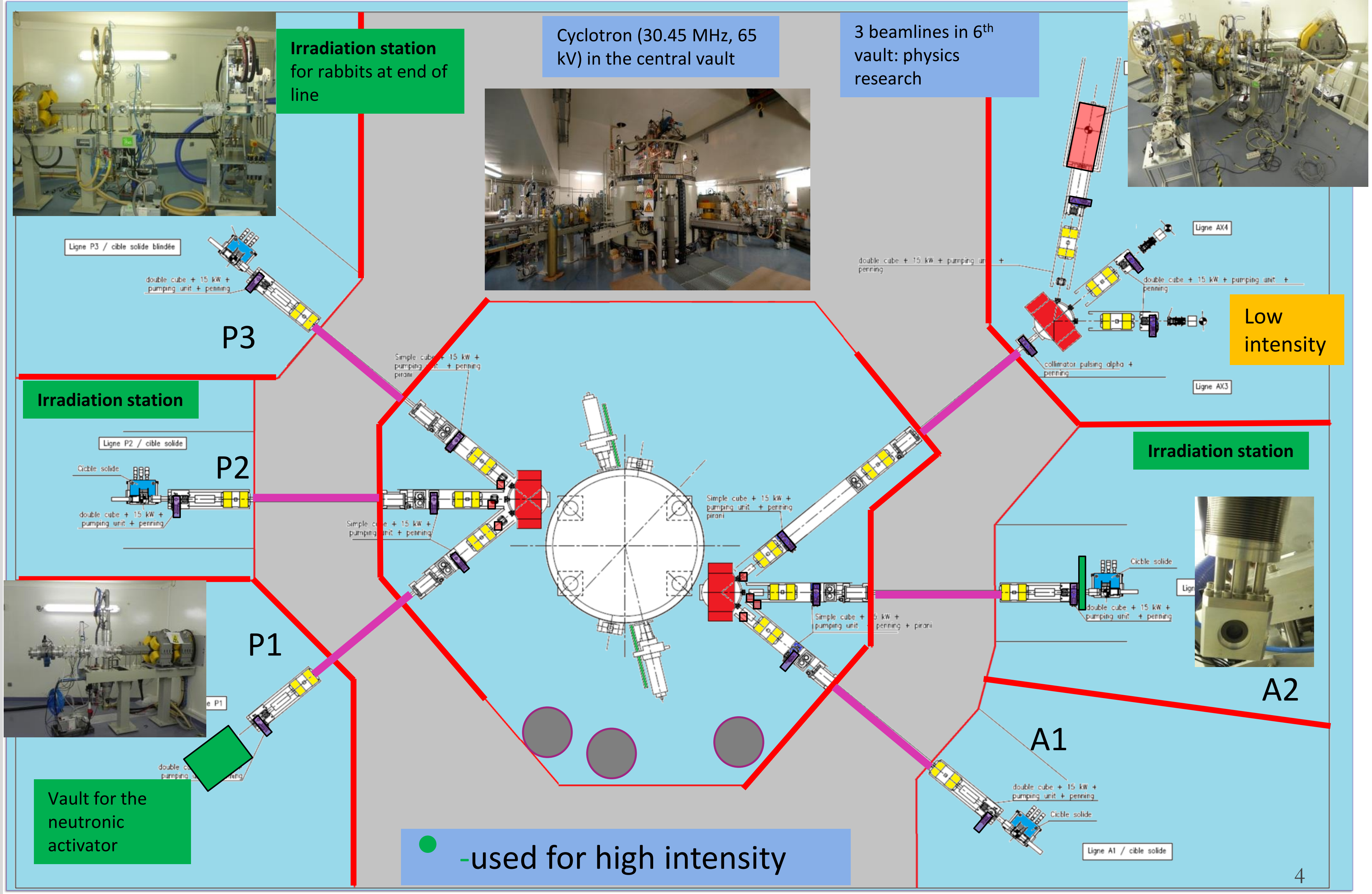
We have achieved 100µA on RbCl target for ^{82}Sr production

In 2016, we switched to RbMetal and increased intensity on target to ~130 uA

Though damages occurs on target → need studies on operation parameters through distinction of unusual events (anomalies)

Beamlines

- Cyclotron
  - 30.45MHz
- 5 irradiation stations for high intensity (proton, 70MeV)
  - Named A1,A2, P1,P2,P3
- 1 vault for low intensity research



Irradiation station for rabbits at end of line

Cyclotron (30.45 MHz, 65 kV) in the central vault

3 beamlines in 6th vault: physics research

Low intensity

Irradiation station

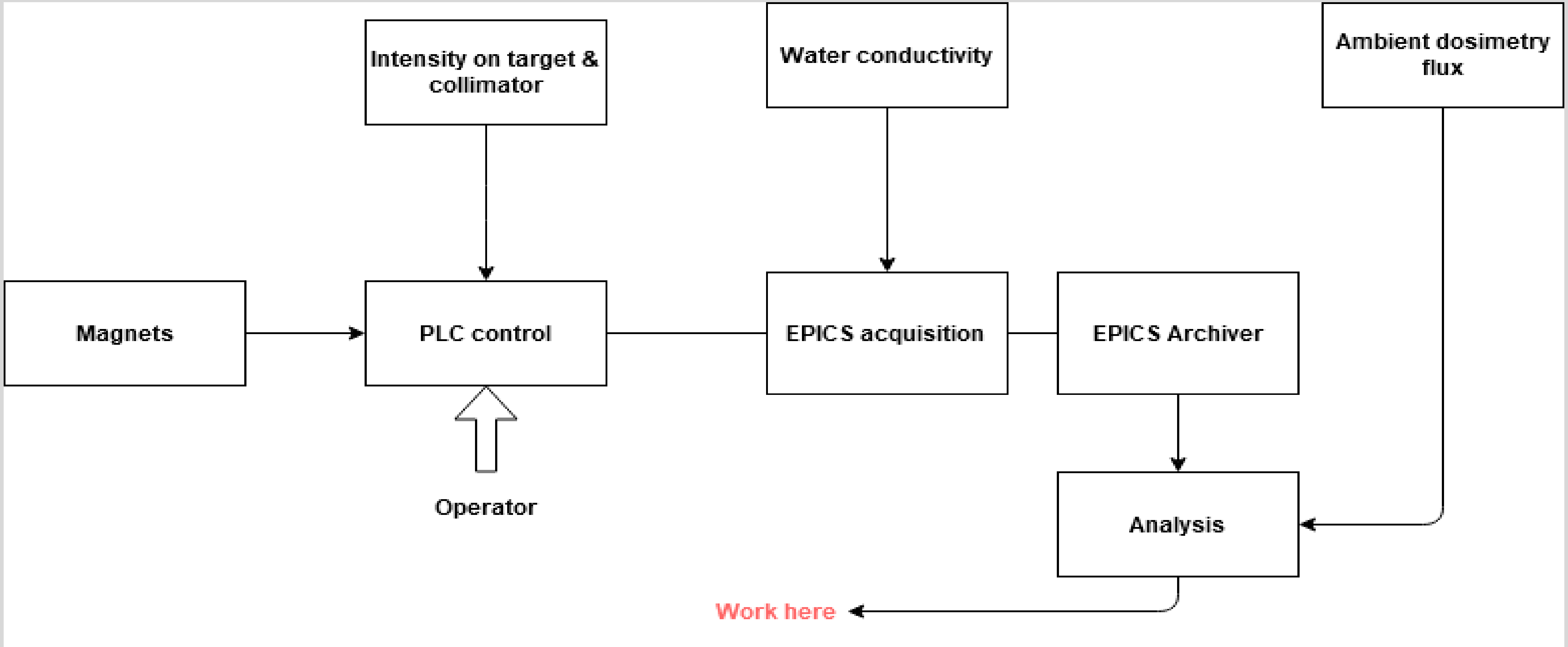
Irradiation station

Vault for the neutronic activator

● -used for high intensity

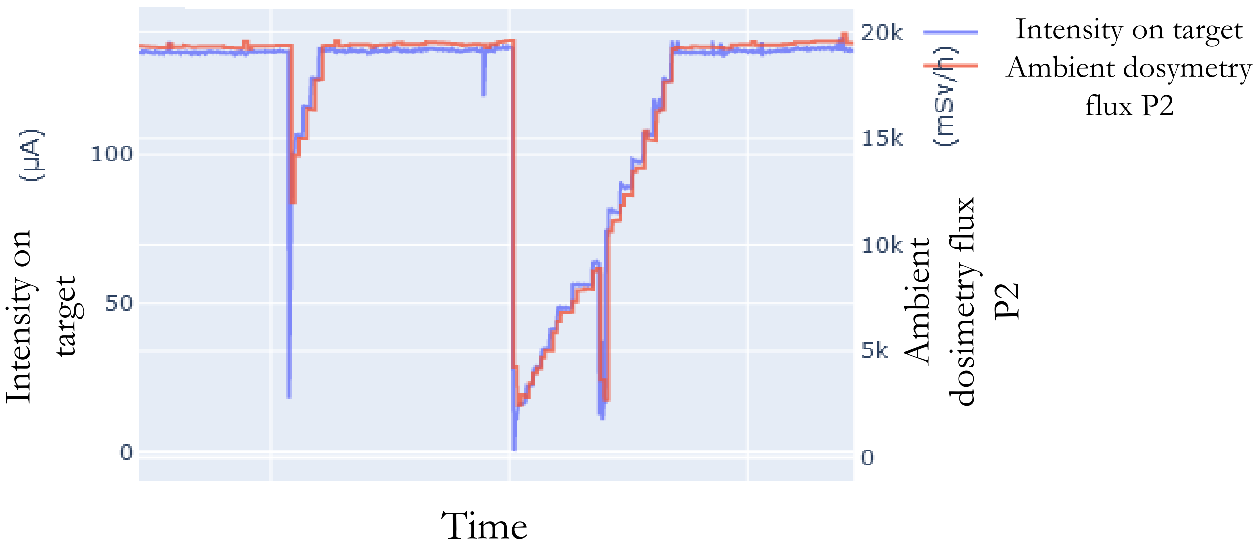
P3
P2
P1
A1
A2
Ligne P3 / cible solide blindée
double cube + 15 kW + pumping unit + penning
Ligne P2 / cible solide
Cible solide
double cube + 15 kW + pumping unit + penning
Simple cube + 15 kW + pumping unit + penning pirani
Simple cube + 15 kW + pumping unit + penning
Simple cube + 15 kW + pumping unit + penning pirani
Simple cube + 15 kW + pumping unit + penning + pirani
double cube + 15 kW + pumping unit + penning
Ligne AX4
Ligne AX3
collimator pulsing alpha + penning
Cible solide
double cube + 15 kW + pumping unit + penning
double cube + 15 kW + pumping unit + penning
Cible solide
Ligne A1 / cible solide

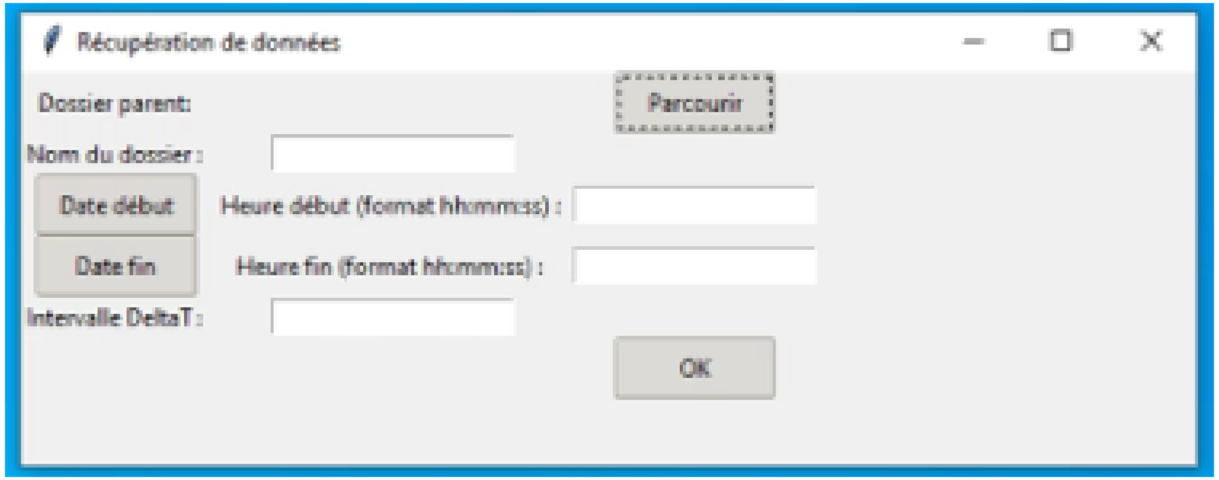
Collecting data



Data acquisition and archiving within EPICS environment

Intensity on target and Ambient dosymetry flux P2





Dossier parent:

Nom du dossier :

Heure début (format hh:mm:ss) :

Heure fin (format hh:mm:ss) :

Intervalle DeltaT :

Graphical interface for data retrieval

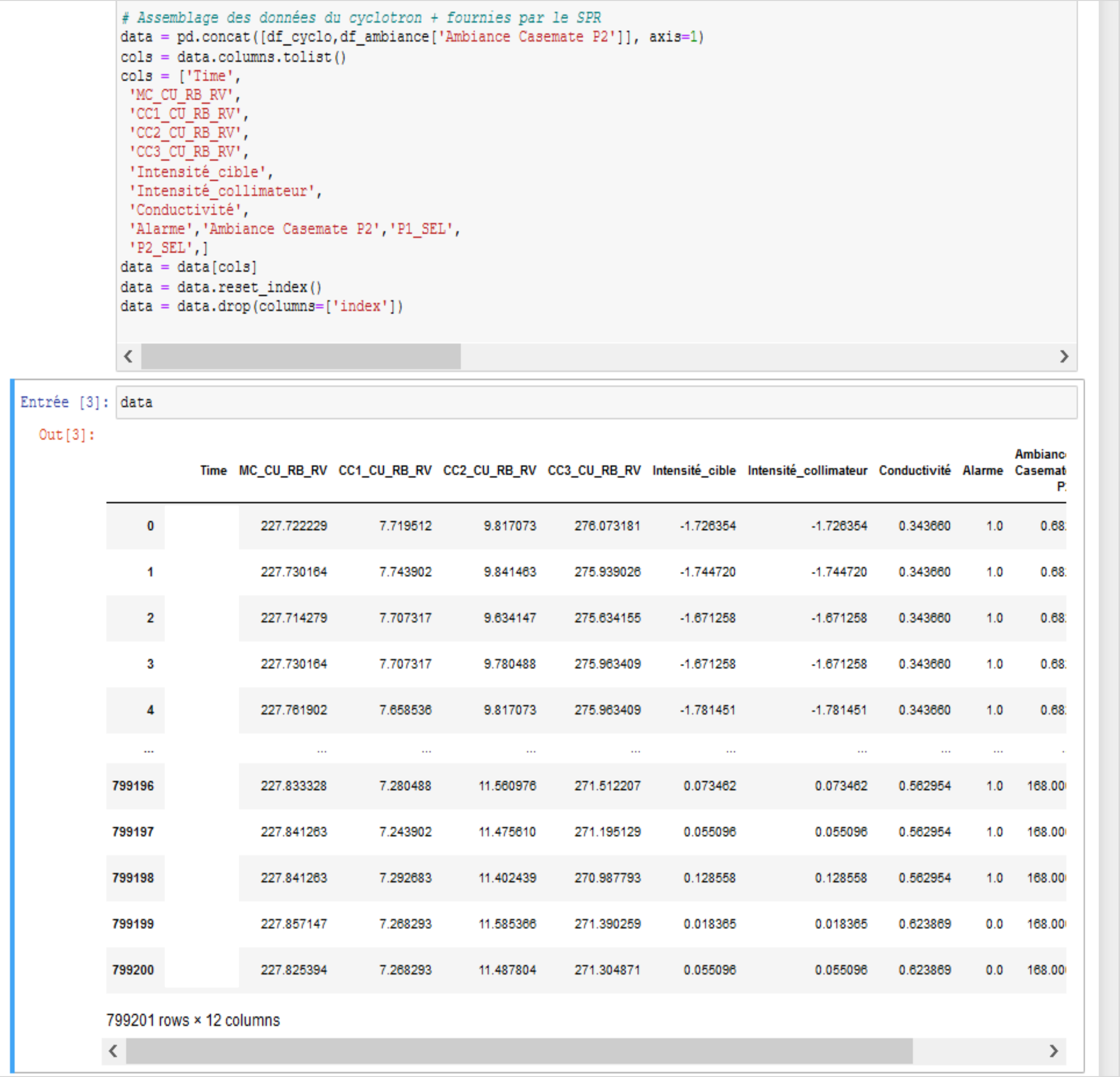
Data

- 3 datasets, each over couple of weeks, from Strontium 82 production :
  - **2 with suspected damage**
  - **1 without suspected damage**
- **But unsupervised learning**: a priori knowledge of damage is not used
- Data from several sources :
  - **Cyclotron data**
    - Retrieval from the archiver with a Python program
      - Intensity on target
      - Intensity on collimator
      - Intensity in coils
      - Water conductivity + Alarm(slope between t and t-1h)
      - Beamline selection
  - **Ambient dosimetry flux**
    - Synchronization with cyclotron data
    - Adding values ($\Delta t_{cyclo} = 1s \ vs \ \Delta t_{Amb.dos.flux} = 1 \ min$)

Software tools

In order to write the anomaly detection programs indicating significant events, the following software/libraries were used:

- Python (3.8.5)
- Jupyter with Anaconda environment (4.8.4)
  - Notebooks
- Libraries :
  - Pandas, Numpy, Scipy : data structure
  - Matplotlib, Seaborn, Plotly, Bokeh : visualization
  - Scikit-learn : machine learning



```
# Assemblage des données du cyclotron + fournies par le SPR
data = pd.concat([df_cyclo,df_ambiance['Ambiance Casemate P2']], axis=1)
cols = data.columns.tolist()
cols = ['Time',
 'MC_CU_RB_RV',
 'CC1_CU_RB_RV',
 'CC2_CU_RB_RV',
 'CC3_CU_RB_RV',
 'Intensité_cible',
 'Intensité_collimateur',
 'Conductivité',
 'Alarme','Ambiance Casemate P2','P1_SEL',
 'P2_SEL',]
data = data[cols]
data = data.reset_index()
data = data.drop(columns=['index'])
```

```
Entrée [3]: data
```

Out[3]:

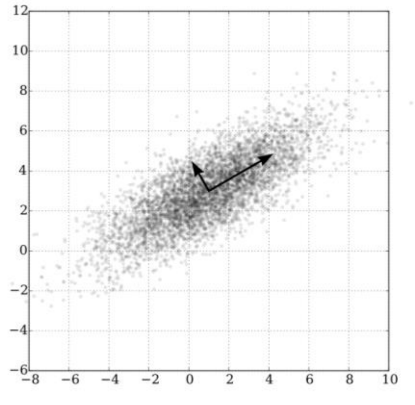
	Time	MC_CU_RB_RV	CC1_CU_RB_RV	CC2_CU_RB_RV	CC3_CU_RB_RV	Intensité_cible	Intensité_collimateur	Conductivité	Alarme	Ambianc Casemat P:
0		227.722229	7.719512	9.817073	276.073181	-1.726354	-1.726354	0.343660	1.0	0.68
1		227.730164	7.743902	9.841463	275.939026	-1.744720	-1.744720	0.343660	1.0	0.68
2		227.714279	7.707317	9.634147	275.634155	-1.671258	-1.671258	0.343660	1.0	0.68
3		227.730164	7.707317	9.780488	275.963409	-1.671258	-1.671258	0.343660	1.0	0.68
4		227.761902	7.658536	9.817073	275.963409	-1.781451	-1.781451	0.343660	1.0	0.68
...		...	...	...	...	...	...	...	...	.
799196		227.833328	7.280488	11.560976	271.512207	0.073462	0.073462	0.562954	1.0	168.00
799197		227.841263	7.243902	11.475610	271.195129	0.055096	0.055096	0.562954	1.0	168.00
799198		227.841263	7.292683	11.402439	270.987793	0.128558	0.128558	0.562954	1.0	168.00
799199		227.857147	7.268293	11.585366	271.390259	0.018365	0.018365	0.623869	0.0	168.00
799200		227.825394	7.268293	11.487804	271.304871	0.055096	0.055096	0.623869	0.0	168.00

799201 rows × 12 columns

Principal component analysis & K-means

Principal component analysis

- Allows a reduction in data size (number of variables)
- Decorrelation of variables



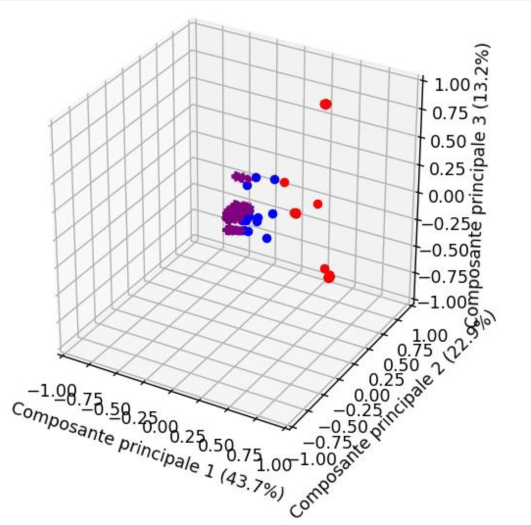
K-means

The simplest clustering algorithm : only 1 hyperparameter (number of clusters k)

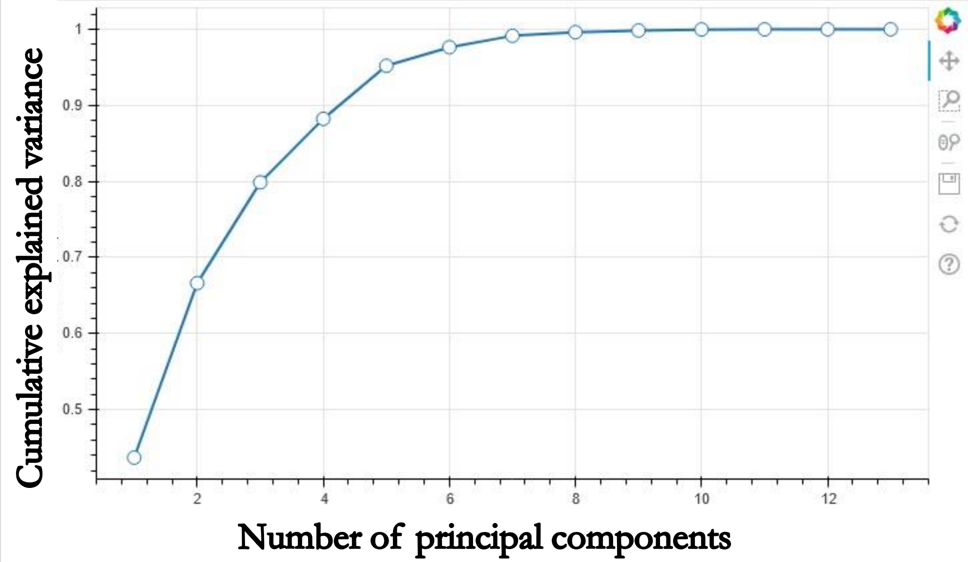
$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \left\|\mathbf{x}_j - \boldsymbol{\mu}_i\right\|^2$$

- Example of application ($\Delta t_{data} \approx 10$ min) of PCA + K-means

Display of data in the space of the 3 principal components providing the most information with K-means clustering (warning : blue and purple points belong to the same cluster, the colors differentiate the points according to the values of the intensity on the target (comparison with a threshold))

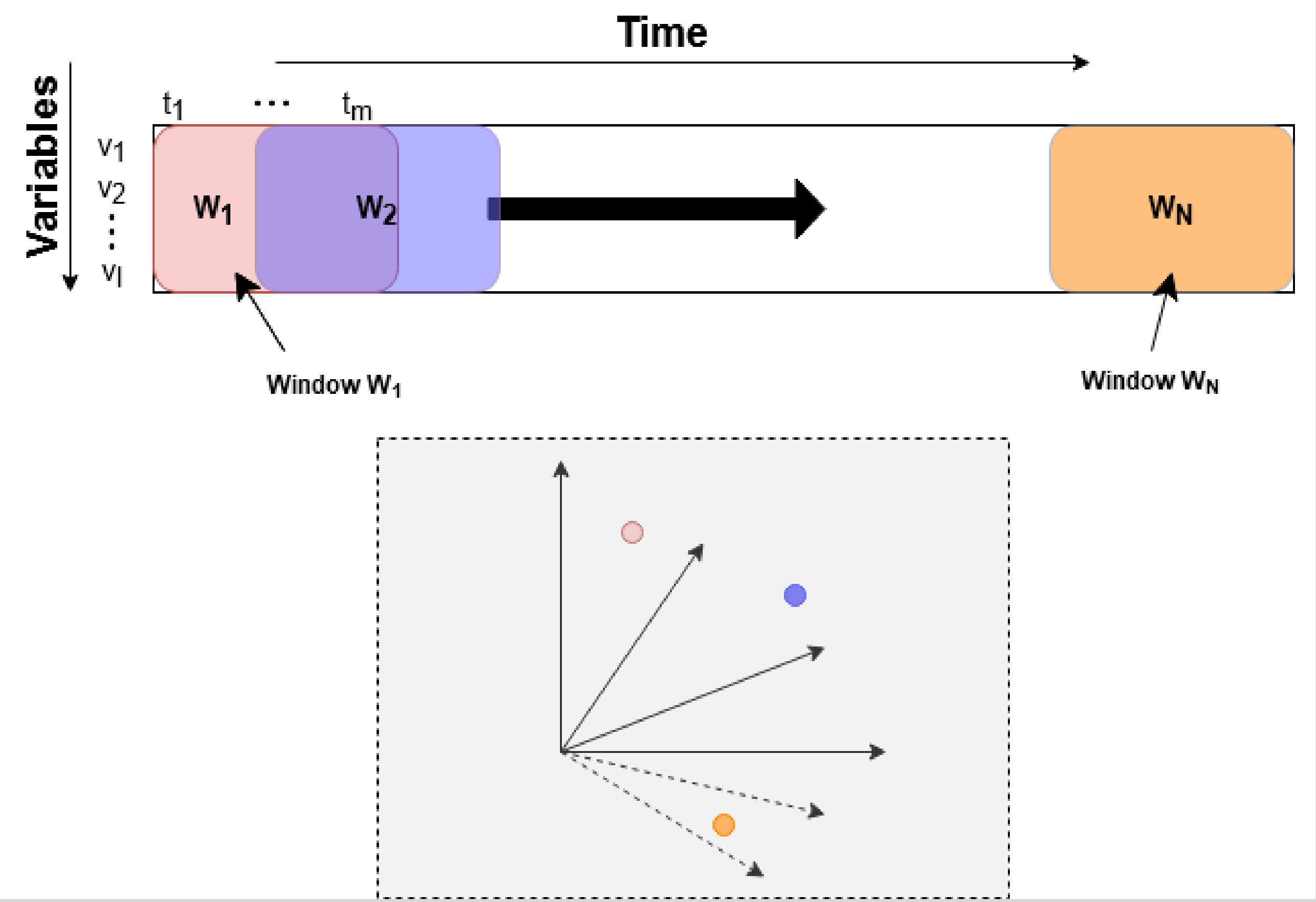


Cumulative variance explained as a function of the number of principal components



K-means : requires prior knowledge about number of clusters : not suitable for this case

Temporal analysis



Representation of the windows in a vector space E : dim E = m * l

- The data are divided into windows containing the values of the variables studied for 1 hour.
- The windows overlap and the distance between two consecutive windows is 5 min; limitation of the RAM memory for the calculation of the distance matrix (O(n²)).

➢ Window size k = number of variables l x number of rows m describing a window (see slide 4 on dataframe). This is proportional to the window duration

➢ A metric such as **Euclidean distance** can be calculated to **determine the similarity** between windows.

➢ Provides information on the **variations along the time of the variables**

Distances matrix

- In this way, we can build representations to visualize the similarity between windows.

We have for each window :

- **Distance to nearest neighbor**
- **Number of neighbors in an arbitrary neighborhood**

- The distance matrix is also calculated in order to apply it as input in a clustering algorithm.

Here, large distances mean that the windows are unique.

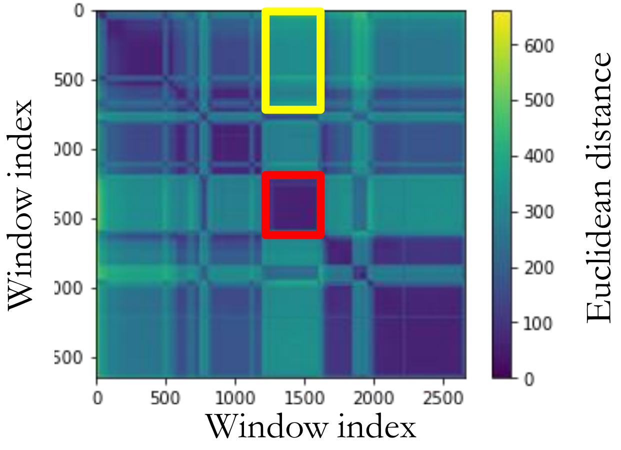
Robustness ?

No, because it depends on the radius describing the neighborhood

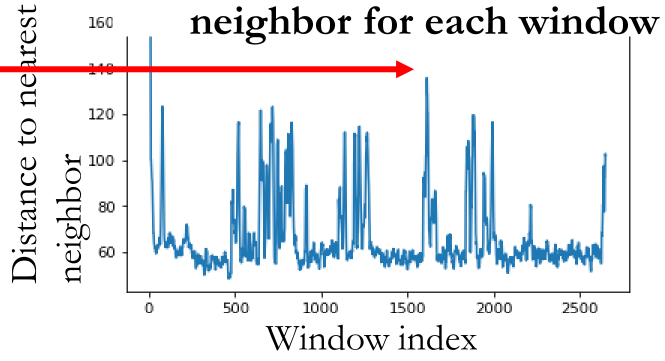
Many windows are close to each other

Some windows are far away

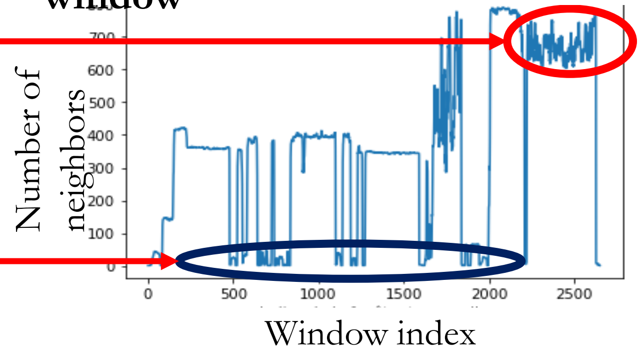
November 2018 data



Distance to nearest neighbor for each window



Number of neighbors for each window



Clustering

- Aggregate data into homogeneous, similar groups.

- Example of algorithm : DBSCAN (Density-based spatial clustering of applications with noise)
  - One of the most commonly used algorithms
  - Special features :
    - Based on data density
    - Data can have complex forms (vs k-means)
    - Allows to process **data with noise**
    - 2 hyperparameters:
      - ➢ **Distance from the neighborhood ε**
      - ➢ **Minimum number of points in a cluster**

➢ Disadvantage

- ➢ Requires a good knowledge of the data since the choice of hyperparameters has a significant impact on clustering.

Here, a point = a window located in a space with p (number of variables) x l (number of lines of the dataframe on which the windows extend) dimensions

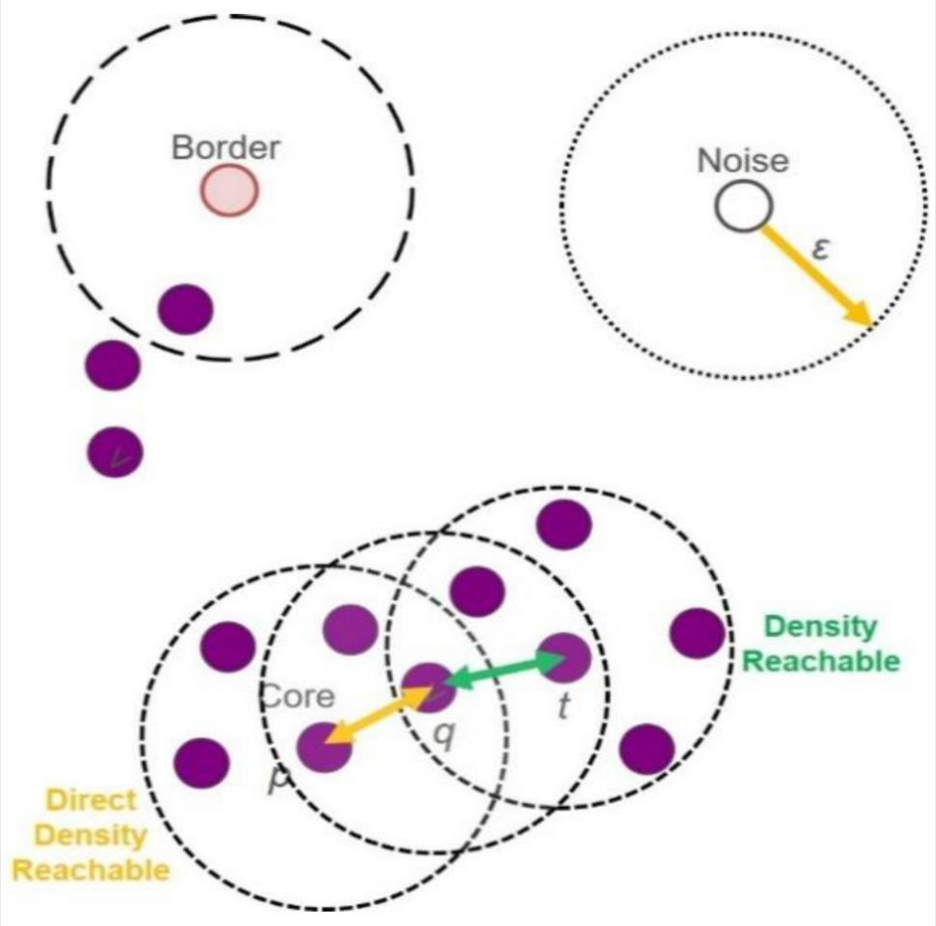
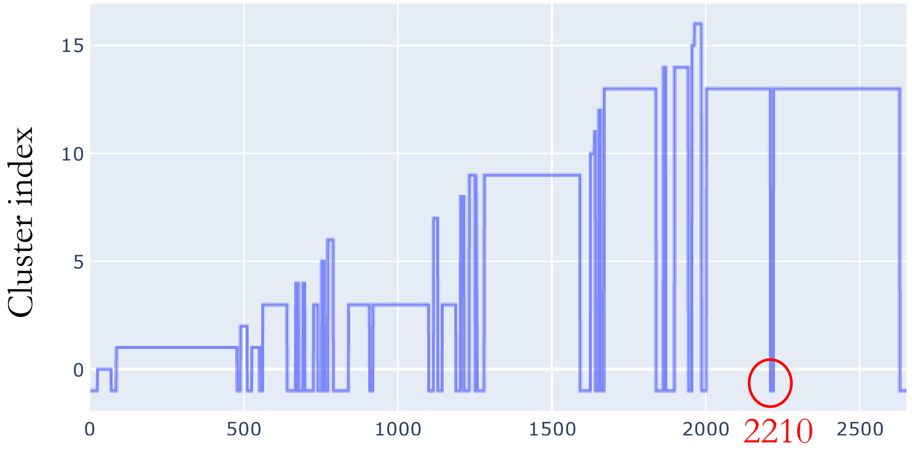
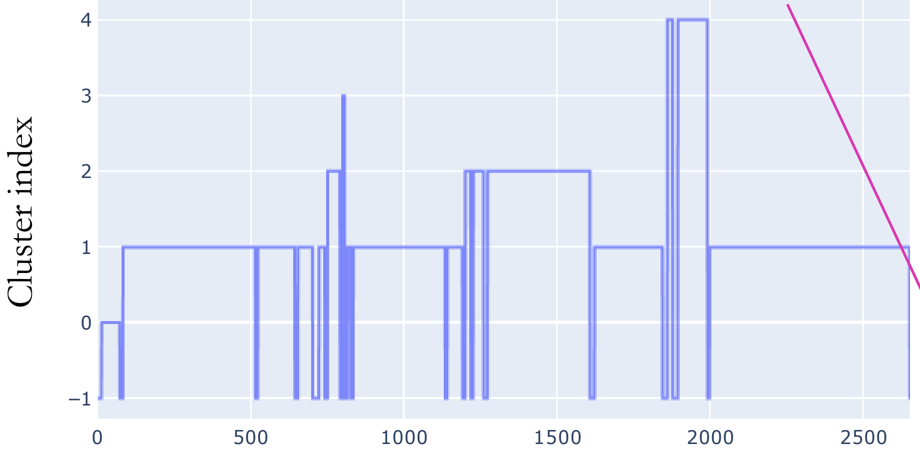


Illustration of DBSCAN

$MinPts = 3$
$\varepsilon = 75$

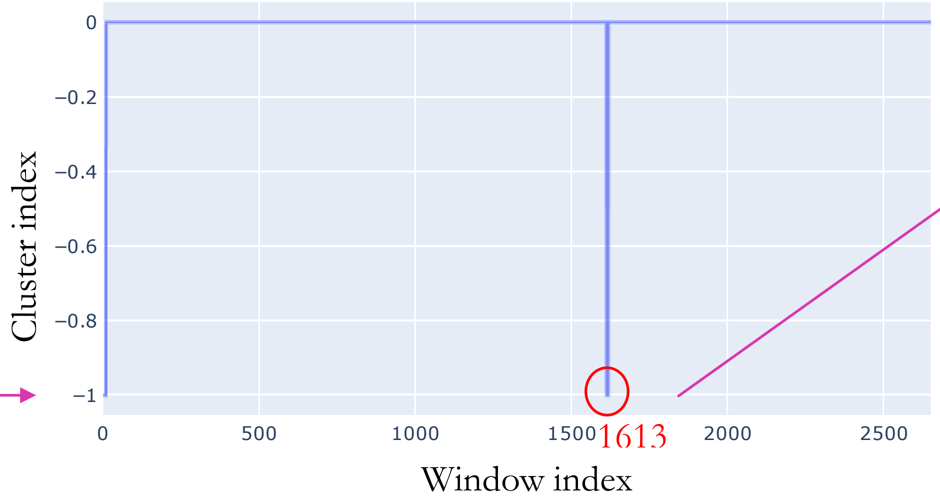


$MinPts = 3$
$\varepsilon = 100$



$MinPts = 3$
$\varepsilon = 125$

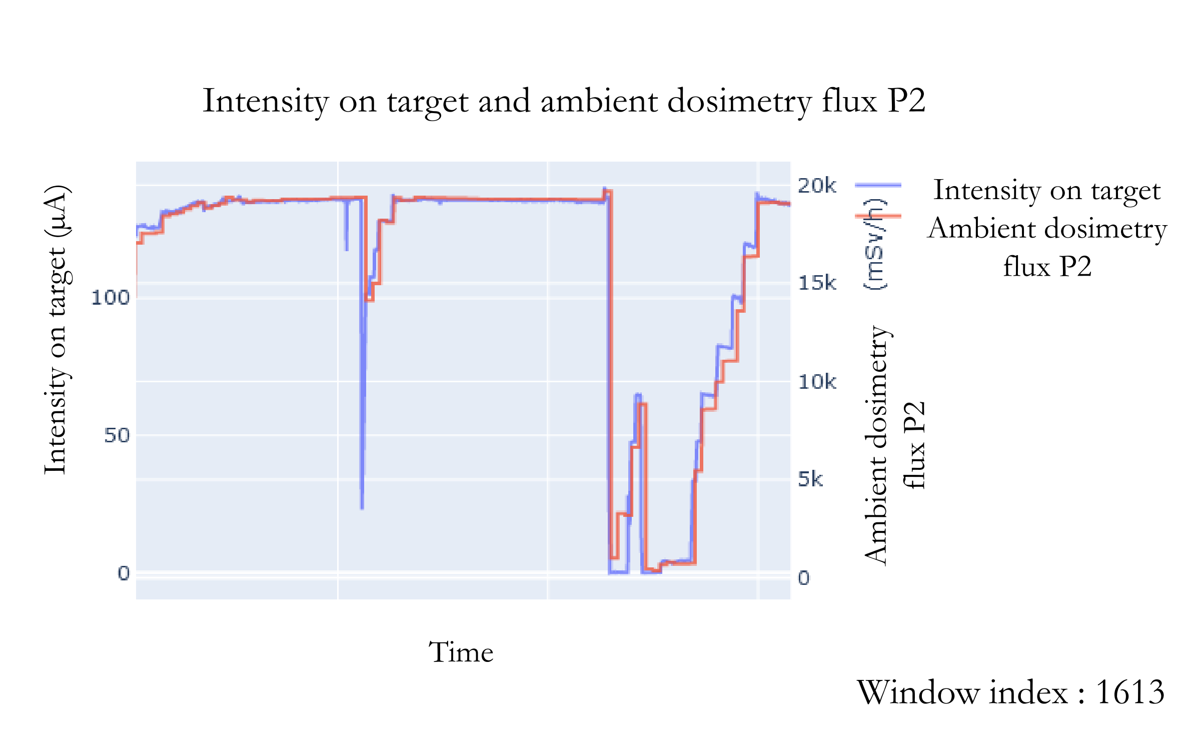
No clustering →

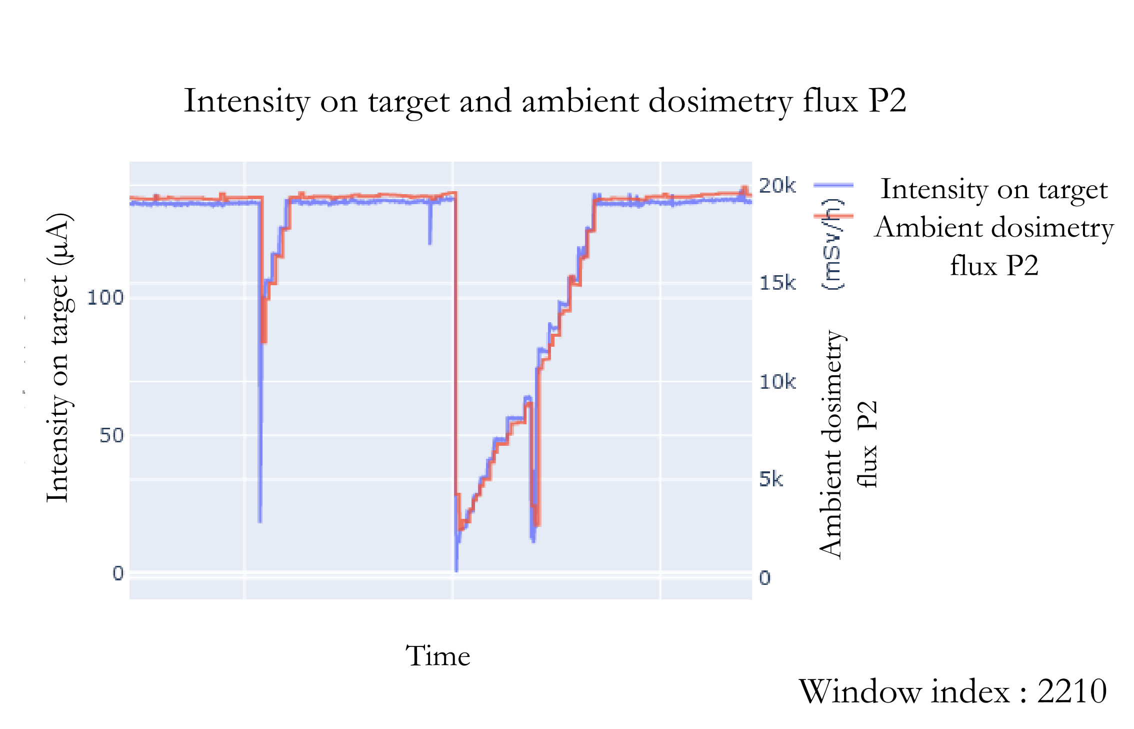


Application of DBSCAN clustering and study of the impact of hyperparameters

- Warning: cluster -1 means that the point is not assigned to any cluster.
- Results strongly dependent on the values of the hyperparameters!
- This can nevertheless provide us with information by looking at the temporal evolution of the variables.
- Some windows are very different with their consecutive windows

We can then check what causes these **significant events (anomalies).**





We notice in these time windows an abrupt variation concerning the intensity on the target and the ambient dosimetry flux.

- To date, manual verification is still required !

➢ Automation of the verification: computation of integrals for each window and comparison ?

Conclusion

- Can selection methods be applied to the data ? ✓
- Can data from different sources be assembled ? ✓ Cyclotron and SPR data
- What is the most suitable methodology ? ✓ Time study
- Choice of algorithm ? ✓ Clustering with DBSCAN
- Do these analytical methods bring out ensembles within the samples? ✓ Identification of variations
- Which parameterization to use for the algorithm ? ✗ Still to be determined
- Data retrieval and assembly / Analysis / Data information ✓

Further research & objectives

Adjusting the hyperparameters of the DBSCAN algorithm

Grid search: testing of several combinations of hyperparameters

Selection of a clustering quality criterion :

: Calinski-Harabasz, Davies-Bouldin, silhouette …

Evaluate the number of anomalies for different values of ε

→ Reject the anomalies for highest values and recognize them for lowest values (selection tool)→ plot the evolution of the number of anomalies as a function of ε

Keep the anomalies which are not too far → the choice of epsilon is very important with this approach

Moving from unsupervised to supervised learning

Recognition of sample damage

Classification of new samples

ARRONAX

Clustering quality criteria

Calinkski-Harabasz index (to be maximized)

- This is the ratio of inter-cluster variance to intra-cluster variance.

- Clustering performs K clusters among N individuals $x^i = (x_1^i, \cdots, x_p^i)$ characterized by p coordinates. We note I_k all the points belonging to a cluster k.
- Let $\mu_k = \frac{1}{|I_k|}\sum_{i \in I_k} x^i$ be the middle point of a cluster k and $\mu = \frac{1}{N}\sum_{i=1}^{N} x^i$ be the middle point of the whole cloud.

- Inter-cluster variances B and intra-cluster variances W_k are defined as follows:

$$B = \sum_{k=1}^{K} |I_k| \|\mu_k - \mu\| \qquad W_k = \frac{1}{|I_k|}\sum_{i \in I_k} \|x^i - \mu_k\|$$

- One can thus calculate the Calinski-Harabasz index :

$$S_{CH} = \frac{(N-K)B}{(K-1)\sum_{k=1}^{K} W_k}$$

Davies-Bouldin index (to be minimized)

- It is the average of the maximum ratio between the distance from a point to the center of its cluster and the distance between two cluster centers.

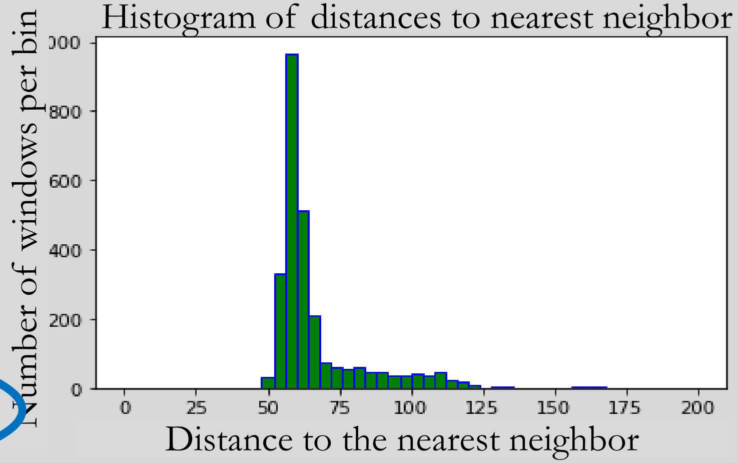
- Let $\mu_k = \frac{1}{|I_k|}\sum_{i \in I_k} x^i$ be the average point of a cluster k and $\delta_k = \frac{1}{|I_k|}\sum_{i \in I_k} d(x^i, \mu_k)$ be the average distance between a point and the center of its cluster.
- The expression of the Davies-Bouldin index is then :

$$S_{DB} = \frac{1}{K}\sum_{k=1}^{K} \max_{k' \neq k}\left(\frac{\delta_k + \delta_{k'}}{d(\mu_k, \mu_{k'})}\right)$$

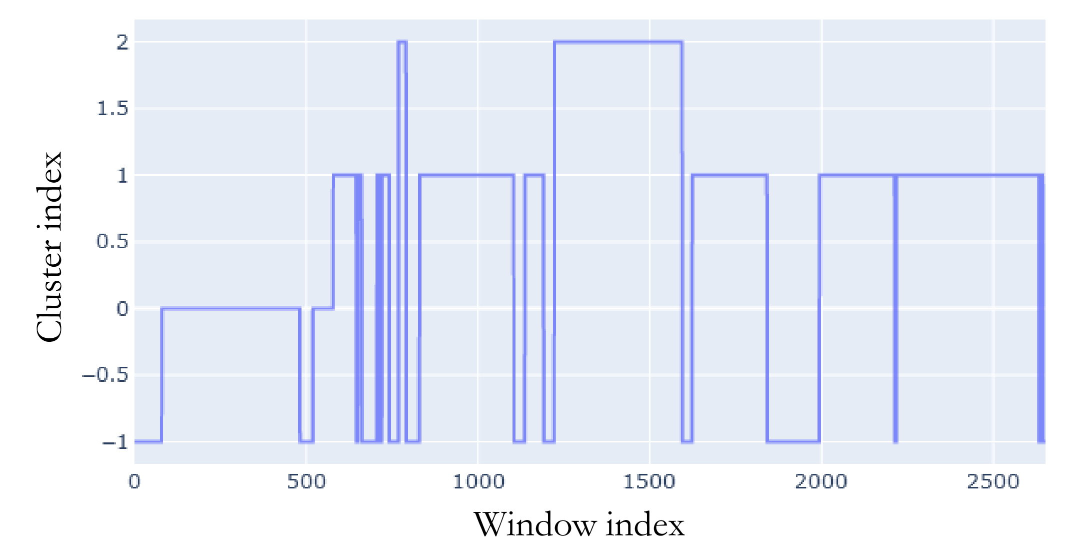
Grid search

- $\varepsilon\ varies\ 40\ between\ 120$
- $MinPts\ varies\ between\ 2\ and\ 1000$

Histogram formed from the vector of distances to the nearest neighbor of each window (see slide 10)

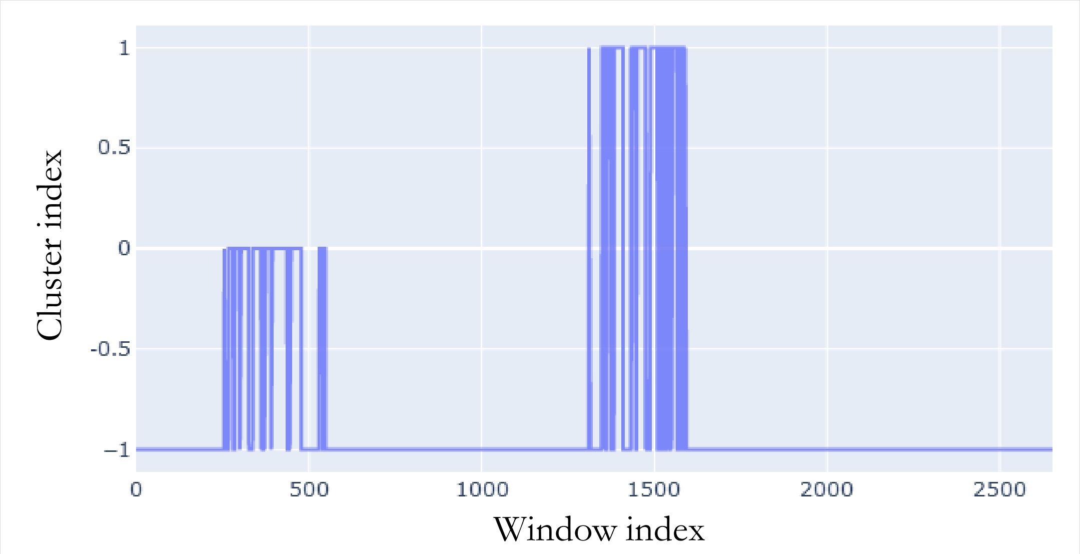


Calinkski-Harabasz index



➡ The results of clustering seem encouraging.

Davies-Bouldin index



➡ The results of clustering are not satisfactory.