This seminar covered an **introduction to machine learning for histopathology**, presented by *Paul Tourniaire* from the 3iA Côte d'Azur Institute.

Histopathology is a gold standard of cancer treatment diagnosis procedures. As part of the French multi-disciplinary diagnosis framework "RCP", pathologists play a key role in charting treatment protocols for patients, using key info such as tumor sizes, metastases, nodes, etc. With the rapid growth of AI in fields such as healthcare, a question arose in histopathology:

Can AI help pathologists diagnose cancer and assign the best treatments?

1 The Research's Goal

The research is undergone by P. Tourniaire, H. Delingette, M. Ilié, P. Hofman, and N. Ayache/ It aims at developing an **AI-based analyzer of imaging and biological data** (i.e. markers) with a predictive capability so as to help better select therapy avenues for Non Small Cell Lung Cancers (abbr. NSCLC).
The current state of research has already resulted in a multi-modal model that is successful in predicting immunotherapy response. The next step is to extend the model to predict for the chance of success of a treatment based on bio-markers.

2 Data Availability and Use

Biopsy and resections are the main ways to produce histopathological data. Tissues are sliced and stained using color agents (e.g. hematoxylin) that helps researchers create digitized slides: **high resolution multi-level pictures in a single digitized file format called a HES Whole Slice Image (WSI), or pyramidal file** (see Fig. 1).
This format holds multiple magnifications for varying levels of analysis. To process the data, researchers have access to both generic (e.g. scikit-learn, PIL) and dedicated (e.g. openslide, pyvips) Python libraries for computer vision.
As part of the project, two other types of data are used: IHC slides, and bionomics.

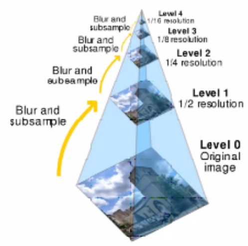

Figure 1: Pyramidal Image Structure

3 Issues and Challenges of Using WSI

WSI are large files that cannot be processed at once. A **tiling approach is almost always necessary**. Thanks to the different magnifications and the tiling, a so-called multiple-instance learning (MIL) is possible. WSI are not normalized, and show multiple colour discrepancies and many types of artefacts (e.g. folds, tears). A strong pre-processing phase is thus necessary.

4 Data Pipeline and Models

Slides are pre-processed using multiple, successive methods (see Fig. 2). More complex combinations of image filters and classifiers can be applied (e.g. HistoQC). Once pre-processed, images are fed into a model.

First task		Second Task		Third Task		Fourth Task
Grayscaling Thresholding	→	filter out noisy elements	→	Tiling	→	Staining Normalization (macenko method)

Figure 2: Pre-processing Pipeline

Three models were covered: **CLAM** (ResNet50 performing feature extraction followed by Multi-Class Attention Branches that yields an attention score), **MIL+RNN** (classifier RNN outputing slide targets), **CHOWDER** (uses local descriptors and a CNN to output classifier predictions). Performance is checked via a Area-under-the-Curve (AUC) metric using mean-pooling as a baseline.

5 Conclusion

Computational histopathology has now reached human-comparable results on cancer related tissues in many different tasks (e.g. tumour classification, survival prediction). The future is bright for RCPs.

The seminar covered **an Introduction to Reinforcement Learning and Deep RL** (RL), presented by *Lucile Sassatelli* (UCA, CNRS, I3S, IUF).

1 Main Concepts

Reinforcement Learning is a a **branch of machine learning aimed at teaching an agent (or several) to react to a dynamic environment in order to to maximize some return**. A Reinforcement Learning model has the following elements:

- An **environment** that can be in a many states ($s \in \mathcal{S}$), which evolution is either random or partly action-dependent
- An **agent** that can perform many actions ($a \in \mathcal{A}$) depending on the state. The agent's action is described by a **policy function** $\pi(.)$ that tells which action to take at a state s. π can be deterministic or stochastic.
- The agent's action leads to an environment output: a **reward** ($r \in \mathcal{R}$)

The resulting evolving system can be described in episodes: $\{(S_0, A_0, R_0)$, ..., $(S_t, A_t, R_t)\}$, and transitions between states follow a **model** described by transition probabilities such as $P(S_{t+1}|S_1, \dots, S_t) = P[S_{t+1}|S_t]$.

$\Rightarrow$ The formal objective of a Reinforcement Learning process is to **maximize the reward associated to the action output by the policy function, which was decided in a dynamic environment**:

$$\pi(.) = \underset{\pi}{argmax}\, \mathbb{E}_{s \in \mathcal{S}, a \sim \pi(.)}[\sum_{t=0}^{+\infty} \gamma^t . r_t]$$

With γ a discount factor, r a reward. $\pi(.)$ impacts the system's episodes.

Two other functions are involved: i) the **action value function** ($Q_\pi(s,a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a]$), which helps choose the next action, ii) the **state value function** ($V_\pi(s') = \mathbb{E}_\pi[G_{t+1}|S_{t+1} = s']$), which tries to predict the cumulated discounted sum of future rewards.

2 Tabular Methods For Reinforcement Learning

Tabular methods refer to problems in which the state and action spaces (the area of possibilities for each) are small enough for the function outputs to be approximated in array and table data structures. If so, many methods are available:

- **Multi-Armed Bandits**: A specific action q has an expected reward: $a_*(a) \doteq \mathbb{E}[R_t|A_t = a]$ allowing to generate trajectories
- **Monte-Carlo Methods**: The predicted trajectory of episodes (S_i, A_i, R_i) is based on an input policy π. There are two possible ways to evaluate the model: i) on-policies (evaluation and improvement on a data-generating policy function), ii) off-policies (different policy than the one used to generate data)
- **Time-Difference Learning**: Involves bootstrapping that does not require a final outcome to compute
- **Tabular Q-Learning**: All $\{S, A\}$ pairs are tracked in a dictionary (thus tabular) in a discrete way

3 Approximating Value Functions

Value functions can be approximated with a weight factor w: $\hat{v}(s,w)_\pi(s)$. It is suited to partially observable problems and can be achieved via different methods (e.g. semi-gradient methods using on and off policy controls, policy gradient methods, deadly triad).

4 Perspectives on Deep Reinforcement Learning

Massive improvement have occured over the last five years such as with Alpha Go. However, deep Reinforcement Learning is not plug and play: it can be sample inefficient and fair competitors can be hard to find. It requires a strongly designed reward function to design and overcome hard-to-escape local optima a. Meanwhile overfitting risk is high and reproducibility is hard to achieve. New models are being developped such as Imitation Learning.