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 multidisciplinary diagnosis framework "RCP", pathologists play a key role in charting treatment protocols for patients, using key data and information 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 imaging and biological data analyzer**. I.e. A neural network that can process and infer from biomarkers so as to 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 responses. The next step is to expand the model to predict a treatment's chance of success based on a defined set of biomarkers.

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.

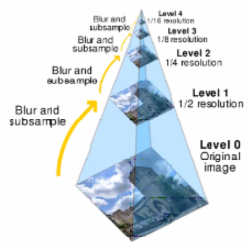

Figure 1: Pyramidal Image Structure

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 outputting slide targets), **CHOWDER** (uses local descriptors and a CNN to output classifier predictions). Performance is checked via an 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 [Deep] Reinforcement Learning** (RL & DRL), 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) how to react to a dynamic environment while maximizing some kind of return via a reward mechanism**. Reinforcement Learning models have the following elements:

- An **environment** that can be in a many states ($s \in \mathcal{S}$) and which evolution can either be random or partly action-dependent. i.e. the environment can react to the agent's *actions*.
- An **agent** that can perform actions ($a \in \mathcal{A}$) dependent on the current and/or past states *s*. An action is the resulting output of a **policy function** $\pi(.)$ given an input state (π can be deterministic or stochastic).
- The agent's actions lead to an environment output called a **reward** ($r \in \mathcal{R}$)

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

$\Rightarrow$ The formal objective of a Reinforcement Learning process is to **maximize the reward associated to the agent's actions, which were produced by the policy function within a dynamic environment**:

$$\pi(.) = \underset{\pi}{argmax}\, \mathbb{E}_{s\in\mathcal{S}, a\sim\pi(.)}[\sum_{t=0}^{+\infty} \gamma^t.r_t] \text{ with } \gamma \text{ a discount factor and } r \text{ the reward}$$

$\pi(.)$ outputs actions by processing the expectation of future episodes discounted by a time factor.

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 where state and action spaces (their area of possibilities) are small enough that possible outputs can be approximated into an array data structure. Many methods are available, e.g.:

- **Multi-Armed Bandits**: A specific action q has an expected reward $a_*(a) \doteq \mathbb{E}[R_t|A_t = a]$ that helps generate trajectories.
- **Monte-Carlo methods**: The predicted trajectory of episodes (S_i, A_i, R_i) is based on the input policy π with two ways to evaluate the model: i) **on-policies** (evaluation and improvement is done on $\pi(.)$, the policy that makes decisions), ii) **off-policies** (evaluation and improvement is done on another kind of function).
- **Time-Difference Learning**: It involves bootstrapping and does not require waiting for a final outcome.
- **Tabular Q-Learning**: All $\{S, A\}$ pairs are tracked in a dictionary (thus tabular) in a discrete way.

3 Approximating Value Functions

Action and state value functions can be approximated via a weight vector w such that $\hat{v}(s, w) \approx v_\pi(s)$ instead of using a tabular method. Approximating is suited to partially observable problems and can be done via many methods (e.g. semi-gradient methods with on/off-policies, gradient methods, deadly triad, etc.).

4 Perspectives on Deep Reinforcement Learning

Massive improvement have occurred over the last five years, such as with Alpha Go. However, Deep Reinforcement Learning is not yet a plug and play tool. Deep RL can be sample-inefficient and fair competitors to assess the quality of a RL model can be hard to source. It requires a well-designed reward function to overcome hard-to-escape local optima. Meanwhile overfitting risk remains high and reproducibility is hard to achieve. Finaly, new models are being developed, such as Imitation Learning models.