Statistical methods for the diagnosis, prognosis and treatment strategies of cancer patients

Méthodes statistiques pour le diagnostic, le pronostic et les stratégies de traitement des patients atteints de cancer

Mémoire d'Habilitation à Diriger des Recherches

Alice Cleynen

soutenance le 19 Novembre 2024

Jury :

Julien Chiquet	DR INRAE	Examinateur
Benoîte De Saporta	Professeure Université de Montpellier	Examinatrice
Anne Gégout-Petit	Professeure Université de Loraine	Examinatrice
Odalric-Ambrym Maillard	CR INRIA	Rapporteur
Nacho Molina	DR CNRS	Examinateur
Nikolay Shirokikh	Group leader Australian National University	Examinateur
Nathalie Vialaneix	DR INRAE	Rapportrice

Rapporteur absent : Gavin Huttley, Professeur Australian National University

Université de Montpellier

IMAG
Institut Montpelliérain Alexander Grothendieck

Charte d'intégrité scientifique

Je déclare avoir respecté, dans la conception et la rédaction de ce mémoire d'HDR, les valeurs et principes d'intégrité scientifique destinés à garantir le caractère honnête et scientifiquement rigoureux de tout travail de recherche, visés à l'article L.211-2 du Code de la recherche et énoncés par la Charte nationale de déontologie des métiers de la recherche et la Charte d'intégrité scientifique de l'Université de Montpellier. Je m'engage à les promouvoir dans le cadre de mes activités futures d'encadrement de recherche.

Alice Cleynen

Adresse : Institut Montpelliérain Alexander Grothendieck UMR 5149, Université de Montpellier, Case courrier 051, Place Eugène Bataillon, 34090 Montpellier, France
E-mail : alice.cleynen@umontpellier.fr
Page internet : https://alice.cleynen.fr/

Remerciements

Je tiens à remercier sincèrement Gavin Huttley, Odalric-Ambrym Maillard et Nathalie Vialaneix d'avoir donné votre temps pour relire mon manuscrit, apprécier son contenu, initier des discussions scientifiques, et encourager ma soutenance. Un merci bien particulier à Nathalie, pour m'avoir soutenue et encouragée dans mes plus grands moments de stress, et surtout pour l'exemple que tu offres au quotidien, donnant le plus de toi-même sur chaque aspect de ton travail, en n'oubliant jamais que les interactions humaines sont les plus importantes.
Un grand merci également à Julien Chiquet, Benoîte de Saporta, Anne Gégout-Petit, Nacho Molina et Nikolay Shirokikh d'avoir accepté de faire partie de mon jury, d'avoir bravé Etamine pour vous déplacer. C'est un honneur pour moi de vous avoir dans mon jury.

Il n'y aurait pas cette soutenance d'HDR sans Benoîte, qui très peu après mon arrivée à Montpellier m'a abordée pour me parler de détection de rupture. Modèles en temps continu pour Benoîte, en temps discret pour moi; application en fiabilité pour Benoîte, en cancérologie pour moi, quelles étaient les chances que nous trouvions un point d'intérêt commun ? Ce sont finalement des modèles en temps continu avec des applications médicales qui nous auront occupées pendant plusieurs années, nous permettant par la suite de rencontrer Régis Sabbadin et d'étendre nos modèles et nos approches dans le contexte de l'intelligence artificielle. Je suis certaine que nous avons encore de quoi nous occuper pendant quelques années (et que nous allons finir par réparer notre preuve) ! Je te remercie donc chaleureusement de nous avoir permis de nous embarquer dans cette belle aventure.
Régis, même si mes propres oreilles ne m'auront pas permis de savoir si tu as des oreilles de dieu ou de vieux, c'est un véritable plaisir de travailler avec toi. J'admire ta patience et ta bonne humeur, ton humour, ta détermination et ton enthousiasme constant. J'aime la façon dont nos projets progressent sûrement, et chacune de tes visites est une garantie de travail et de rire, et de nouvelles discussions. Je remercie Sophie Lèbre aussi, de ne pas s'être laissée découragée par le temps qu'il nous aura fallu pour nous mettre à travailler ensemble. Je suis vraiment heureuse d'avoir pu discuter avec toi en toutes occasions, partager nos questions, et que nous soyons enfin sur le point d'avancer nos projets. Merci surtout pour ton soutien !

Il serait trop long de citer tous les membres du labo, de Montpellier ou d'Australie, qui m'auront accompagnée ces dernières années et sans qui je ne me serais pas autant épanouie dans mon travail. Je n'oublie pas non plus l'équipe administrative, et notamment Nathalie, sans qui le labo ne tournerait pas. Je suis tout particulièrement reconnaissante à Vianney et Louis pour m'avoir appris à co-encadrer, à Agin, Shafi et Stefan, pour me permettre de vous voir évoluer dans vos thèses, et à Orlane Rossini et Amélie Vernay pour ne prendre que du plaisir à vous suivre, et grandir scientifiquement et humainement avec vous au quotidien.

Merci à Nikolay, de m'avoir ouvert plus grande la porte vers la biologie, vers l'ARN et même parfois vers l'origine de la vie. Si j'ai du mal à fermer la porte des mathématiques, je me sens souvent aspirée par nos discussions vers un monde plus vaste, plus fascinant, plus utile. J'adore nos échanges sans limites, même quand ils nous entraînent au delà du raisonnable, nous dévient de nos objectifs. Avec toi le monde est plus grand, et je t'en suis extrêmement reconnaissante.

Enfin, le plus grand des mercis à ma famille. Olivier, de savoir que tu es là est en soi un soutien infini, de rire de tes aventures et de me reconnaître dedans, merci... À mes deux affreux, mes terreurs des 5 océans, mes petites lunes, mon loup et ma chouette... La plus belle chose de ce monde est de vous entendre rire. À Hoel enfin, toujours par ce que tu t'appelles Hoel et qu'il n'y en a qu'un comme toi. Il n'y aurait vraiment pas d'HDR sans toi. Il n'y aurait pas le rire de nos enfants. Il n'y aurait pas l'ambition de continuer et l'envie d'avancer. Merci d'être là, toujours, immanquablement, pour nous.

Contents

Préambule

Au delà de l'acquisition d'une expertise en segmentation statistique, les travaux que j'ai réalisés pendant ma thèse m'ont familiarisée avec les challenges des statistiques au service de l'application. Là où les statistiques théoriques partent d'un modèle fixé et s'intéressent à ses propriétés, par exemple aux comportements d'estimateurs en contexte asymptotique, le choix de modèle pour représenter les données n'est clairement pas anodin. Ainsi une mauvaise paramétrisation peut donner lieux à des résultats ininterprétables, une modélisation trop complexe peut empêcher l'utilisation de certaines techniques d'inférence, une modélisation trop parcimonieuse peut empêcher la détection des événements d'intérêts, etc. De même, là où les statistiques théoriques vont proposer des estimateurs robustes, sans biais, optimaux en vitesse de convergence, etc, l'implémentation algorithmique d'un développement méthodologique est un facteur crucial dans son utilité, en particulier à l'heure où les données génomiques sont de plus en plus volumineuses.

Ces deux aspects particuliers ont été déterminants dans mon expérience scientifique, et ne pourront que l'être dans mes travaux ultérieurs. Tout au long de mon parcours j'ai mené, souvent de front, des travaux très théoriques et des travaux au plus proche des données. Parmi les travaux dont je suis la plus fière, il y a le travail avec Emilie Lebarbier (EJS 2017) sur le choix de modèle en segmentation dans la famille exponentielle. C'est le premier travail vraiment abstrait que j'ai proposé, dont le but était de mettre fin à cette longue série d'articles s'intéressant au choix de modèle dès qu'une nouvelle loi venait au goût du jour (nous avions nous-même peu de temps avant participé à cette série en nous intéressant aux distributions de Poisson et binomiale négative). Il y a aussi bien sûr les nombreux travaux sur le myélome multiple, dont celui sur l'impact des gènes de fusions produits par les translocations chromosomiques (Nature Communications, 2017), qui m'a demandé de me plonger au plus près des données, apprendre des mécanismes biologiques et des techniques de séquençage, pour là encore proposer une approche la plus reproductible possible dans l'identification et l'analyse de ces gènes de fusions.

Ce sont cependant des travaux plus récents, moins achevés, que j'ai choisi de présenter dans ce manuscrit, car je pense qu'ils illustrent parfaitement les challenges applicatifs décrits plus haut. Dans la première partie, la question de la modélisation est centrale. Ce sont les processus de Markov déterministes par morceaux (PDMP) qui sont à l'honneur. C'est d'un premier travail avec Benoîte de Saporta (Automatica, 2018, présenté dans les sections 1.2 et 1.3) qu'est née la série de contributions que je présente. Ce premier travail sur la détection de rupture dans les PDMP, combinant mon expertise sur la question biologique ainsi qu'en segmentation avec celle de Benoîte sur les PDMP, est à mon sens une des contributions majeures de ma recherche. La modélisation par PDMP apparaît très naturellement dès lors que le phénomène biologique donnant lieu aux observations est mis à jour. Or finalement, quoi de plus simple qu'un PDMP, dont les trajectoires sont déterministes modulo quelques sauts aléatoires contrôlés par une intensité et un noyau de saut ? Cette simplicité est essentielle pour les médecins, puisqu'elle permet de décrire l'évolution tumorale avec des concepts intuitifs : la durée de vie sans rechute (progression-free survival, PFS, notion extrêmement naturelle en analyse de survie), et l'agressivité de la rechute (qui contrôle la vitesse de croissance de la tumeur). Et pourtant, combien d'articles scientifiques avons-nous lu sur la résolution des contrôles impulsionnels (à instant discret) qui se contentent d'affirmer que "si votre processus est continu, il n'y a qu'à le discrétiser". Sauf qu'en pratique, discrétiser quand la résolution numérique est essentielle à l'application, ce n'est pas si trivial. Les sections 1.3 et 1.4 sont des contributions sur cet aspect, dans des configurations de complexité différentes. De nombreuses questions se posent alors, en particulier quand le processus devient plus complexe (plusieurs variables d'intérêt) ou mal connu (agressivité de la rechute spécifique au patient, risque de rechute inconnu, etc) et ce sont des travaux en cours et avec deux étudiantes en thèses que je présente ici sur ces questions (section 1.5).

Dans la seconde partie, je retourne au plus près des données où l'aspect numérique est vraiment crucial, chaque jeu de données contenant plus d'un million de lectures, chaque lecture consistant en l'enregistrement d'un courant électrique à très haute fréquence. Ce volume de données dicte la simplicité des modèles statistiques considérés. Ce sont deux projets que j'ai menés tout récemment lors de mon séjour en Australie que je présente. Immergée dans un institut de biologie, j'y ai appris, en plus des nombreux mécanismes biologiques qui m'étaient inconnus, l'art de la recherche interdisciplinaire en immersion. Le travail sur la dégradation de l'ARN (section 2.2) m'a permis d'encadrer deux étudiants de biologie (une en Master, l'autre en thèse), et de monter les protocoles expérimentaux pour générer les données permettant de calibrer et valider les modèles que j'ai proposés. Ce travail me tient particulièrement à cœur car c'est la première fois que je contrôle tous les aspects d'un projet, tant méthodologiques qu'expérimentaux. De plus, si le projet était originellement pensé pour corriger les biais de dégradation dans l'étude de l'abondance différentielle des transcrits, c'est une communauté inattendue, celle de l'étude du cycle de l'ARN, depuis sa production jusqu'à sa dégradation, qui s'est emparée de l'outil, et a motivé de nombreuses améliorations et initié de nouvelles collaborations. Le travail sur les modifications de l'ARN (section 2.3) m'a quant à lui introduit à la manipulation des algorithmes de machine-learning, dont je ne connaissais jusque là que les fondements théoriques. Là encore, j'ai pu travailler avec deux étudiants (dont une thèse que je co-encadre) sur l'implémentation et la calibration de réseaux de neurones, tout en apportant mon expertise statistique, par exemple dans les tests entre conditions. Si la réalisation de cet outil reste une contribution très technique, je pense qu'il sera central dans l'étude des modifications de l'ARN, qui sont une des voies les plus prometteuses dans le développement de nouvelles thérapies contre le cancer.

Preamble

Beyond acquiring expertise in statistical segmentation, the work I carried out during my PhD familiarized me with the challenges of applied statistics. While theoretical statistics start from a fixed model and focus on its properties, such as the behavior of estimators in asymptotic contexts, the choice of a model to represent data is far from trivial. A poor parameterization can lead to uninterpretable results, overly complex modeling can prevent the use of certain inference techniques, and overly sparse modeling may hinder the detection of relevant events, etc. Similarly, while theoretical statistics propose robust, unbiased estimators that are optimal in terms of convergence rates, the algorithmic implementation of a methodological development is a crucial factor in its utility, particularly in the current era of increasingly large genomic datasets.

These two specific aspects have been decisive in my scientific experience and will undoubtedly remain so in my future work. Throughout my career, I have simultaneously pursued both highly theoretical work and projects closely tied to data. Among the works I am most proud of is the one conducted with Emilie Lebarbier (EJS 2017) on model selection in segmentation within the exponential family. It was the first truly abstract work I proposed, aimed at ending the long series of papers focusing on model selection whenever a new law became fashionable (we ourselves had previously contributed to this series by studying Poisson and negative binomial distributions). There is also the numerous contributions on multiple myeloma, particularly the study on the impact of fusion genes produced by chromosomal translocations (Nature Communications, 2017), which required me to dive deep into the data, learn about biological mechanisms and sequencing techniques, and propose a reproducible approach for identifying and analyzing these fusion genes.

However, it is more recent, less finalized work that I chose to present in this manuscript because I believe it perfectly illustrates the applied challenges described above. In the first part, the question of modeling is central. Piecewise Deterministic Markov Processes (PDMPs) take the spotlight. This series of contributions, presented in sections 1.2 and 1.3, originated from an initial collaboration with Benoîte de Saporta (Automatica, 2018). This first work on change-point detection in PDMPs, combining my expertise in biology and segmentation with Benoîte's expertise in PDMPs, is, in my view, one of the major contributions of my research. PDMP modeling arises very naturally when the biological phenomenon giving rise to the observations is explicited. And in the end, what could be simpler than a PDMP, whose trajectories are deterministic modulo a few random jumps controlled by an intensity and a jump kernel? This simplicity is essential for clinicians, as it allows for the description of tumor evolution using intuitive concepts: progression-free survival (PFS, a very natural concept in survival analysis), and the aggressiveness of the relapse (which controls the tumor's growth rate). And yet, how many scientific papers have we read about the resolution of impulse control problems (at discrete moments) that simply claim, "if your process is continuous, just discretize it." But in practice, discretization when numerical resolution is essential for the application is not that straightforward. Sections 1.3 and 1.4 contribute to this aspect, in different configurations of complexity. Many questions then arise, particularly when the process becomes more complex (several variables of interest) or not fully known (patient-specific relapse aggressiveness, unknown risk of relapse, etc.). These are ongoing works with two PhD students that are briefly presented here (section 1.5).

In the second part, I return to data-driven work, where the numerical aspect is truly crucial, with each dataset containing more than a million reads, each read consisting of a very high-frequency electric current recording. This data volume dictates the simplicity of the statistical models considered. I present two projects that I recently worked on during my visit in Australia. Immersed in a biology institute, I learned, in addition to the many biological mechanisms I was previously unfamiliar with, the art of interdisciplinary research through immersion. The work on RNA degradation (section 2.2) allowed me to supervise two biology students (one Masters and one PhD student) and set up experimental protocols to generate the data needed to calibrate and validate the models. This work is particularly dear to me because it was the first time I controlled all aspects of a project, both methodological and experimental. Additionally, while the project was initially designed to correct degradation biases in the study of differential transcript abundance, it unexpectedly caught the attention of the research community of RNA cycle, from production to degradation, which led to numerous improvements and new collaborations.

The work on RNA modifications (section 2.3) introduced me to the actual implementation of machine learning algorithms, of which I previously only knew the theoretical foundations. Here again, I worked with two students (one of whom I co-supervise for their PhD) on the implementation and calibration of neural networks, while contributing my statistical expertise, for example, in testing for modification differences between conditions. Although the development of this tool remains a highly technical contribution, I believe it will be central to the study of RNA modifications, one of the most promising pathways in the development of new cancer therapies.

Introduction

There is nothing more fascinating than cancer. In his best-seller, Pulitzer prize, Siddhartha Mukherjee defines it as *The Emperor of All Maladies* (Mukherjee 2010), the disease of the body against the body, the exploitation of the extraordinary cell machinery to infinity. How four letters, *A, C, G* and *T* can command a life within a life can have no matching amazingness.
And yet, there is nothing more exciting than mathematics. Both the absolute truth and a world entirely constructed by humans, mathematics are an infinite resource of tricks and tools to reach the ultimate description of the world. There is nothing like being stuck on an equation, writing it under all it's possible forms, making it more complex, simplifying it, until finally it reveals the hidden solution.
I consider myself incredibly lucky to be able to combine the best of both worlds, using games to understand what we are made of. This thesis is the exposition of some works at the interface of two great scientific fields. Driven by the ultimate goal of designing methodological tools to improve patient care, I have organised this manuscript in two main chapters. The first has a more theoretical flavor, the contribution going beyond cancer research, bringing together tools from the Artificial Intelligence (AI) field, a series of algorithms for the resolution of complex Markov Decision Processes (MDPs), and models from the Probabilistic field, variants of Piecewise Deterministic Markov Processes (PDMPs). The second has a more applied flavor, exploiting models and algorithms to enhance the use of Direct RNA Sequencing (DRS) in patient samples. Those works represent my main contributions over the last 6 years, they were carried out in parallel despite their orthogonal nature.

In the first chapter, I start with the exposition of Piecewise Deterministic Markov Processes (PDMPs), present the context of impulse control, and show how it can be equivalently expressed in the framework of Markov Decision Processes (MDPs), allowing the use of a wide range of resolution algorithms. I then present three contributions designed for the optimal treatment of cancer patients.

In the first work (with Benoîte de Saporta), we propose a simple partially observable PDMP model for the free evolution of cancer marker data, and an algorithm to detect relapse and relapse types as early as possible from noisy and partial observations of the process. From the mathematical point of view, this work represents a major contribution as the first theoretical framework for change-point detection when the process is both time and state-continuous, and the change-points do not occur on observation times. We provide convergence rates and error bounds to our resolution strategy, and show its enhanced performance compared to state-of-the-art approaches.

In the second work (again with Benoîte de Saporta), we increase the complexity of the model by allowing the practitioner to act on the evolution of cancer (typically through introduction of chemotherapies), and to select next visit dates for the patient (which corresponds to (noisy) observation times of the process). Once again, we provide an algorithm to optimally select treatment and visit dates in order to control the cancer load of the patient. From a mathematical point of view, this works provide a significant contribution in the resolution of impulse control problems for partially observable PDMPs where once again both the time and state-space are continuous and change-points do not occur on observation dates. We provide convergence and error bounds for our algorithm, and discuss the numerical difficulties associated to this resolution.

The third work (with Benoîte de Saporta, Aymar Thierry d'Argenlieu and Régis Sabbadin) is the first main bridge between PDMPs's probabilistic framework and the IA approach to MDP resolutions. The biological problem and model remain unchanged, but here we exploit the simulation properties of PDMPs to make use of simulation-based resolution algorithms such as Monte-Carlo tree search approaches. Theoretical convergence properties are in this case particularly hard to derive, however we show that the performance is almost as good as that of the theoretical approach from above, with the possibility to easily extend to more complex models (for instance coupling of PDMPs, which could reveal instrumental to take into account several cancer markers) or to include patient-specific preferences (for instance allowing the cost function to include individual tolerance to treatment side-effects).
Importantly, this work has also opened the gate to the resolution of even more complex problems, for instance cases where the model is only partially known and needs to be learnt while optimising. In the ongoing thesis of Orlane Rossini, we consider that the behaviour of the cancer marker is known up to a (patient-specific) cancer-aggressiveness parameter, and frame the question in a Reinforcement learning paradigm where algorithms from the AI community, such as BAMDP or DDQN will prove extremely efficient.

In the second chapter, I start with a (not so) short biological introduction to the most fascinating processes of (evolved) living organisms: the central dogma of biology, the role of RNA in cell functions, the diversity of RNA modifications and their impact on cell proliferation and cancer. I also briefly introduce Direct RNA Sequencing, a fairly recent technology giving access to the native RNA composition. I then present two main methodological contributions designed to enhance the usage of DRS on cancer patient samples.

In the first work with Nikolay Shirokikh and several students from the ANU, I propose a strategy to estimate RNA degradation from the distribution of long-read sequencing. Focusing on *random fragmentation* and considering the technical specificities of DRS, I propose to model degradation through a Bernoulli process, and propose an estimation framework that combines techniques from segmentation (to capture only the last fragment of a transcript) and survival analysis (to take into account the read-censoring from transcripts rejected from the sequencer). From this estimator, I then propose a *Direct Transcriptome Integrity* number meant to mimic the reference *RNA Integrity Number.* I then provide a goodness of fit test to challenge the assumption of random fragmentation in the sample, normalisation approaches to correct degradation bias from differential transcript expression, and finally a Bayesian test framework to deconvolve *biological* and *technical* degradation effects from the data and perform differential biological degradation analyses in samples with un-eaven technical degradation.

In the second work (methodological contributions with Stefan Prodic and Akanksha Srivastava, and in collaboration with Eduardo Eyras and Nikolay Shirokikh), I propose an algorithm to detect RNA modifications from the raw output of DRS. Specifically, we combine two neural networks to predict modification probability and stoichiometry at the site level. The first neural network takes a set of 7 vectorial features including raw signal, distances to expected signals, sequencing time and base-calling quality scores, and classifies a nucleotide from a single read as modified or not. The second neural network takes a histogram distribution of all outputs of the first network on nucleotide mapping to the same transcriptomic sites, and predicts general modification class as well as stoichiometry. We then propose a generalised linear model to compare the stoichiometries of different replicates from different conditions, and we widely benchmark our method on an extensive sets of cell-line data.

I end this manuscript with a description of how we are currently applying those two works to a large study of Myeloma patients aiming at identifying novel drug-able targets from the standpoint of RNA modifications.

Contributions discussed in this manuscript

[1] Cleynen A. and de Saporta B. *Change-point detection for piecewise deterministic Markov processes*, (2018) *Automatica* 97,234-247: 10.1016/j.automatica.2018.08.011

[2] Cleynen A. and de Saporta B. *Sequential decision making for a class of hidden Markov processes, application to medical treatment optimisation* (2023) arxiv:2112.09408

[3] de Saporta B. and d'Argenlieu A. T. and Sabbadin R. and Cleynen A. *Medical follow-up optimization: A Monte- Carlo planning strategy* (2024) arxiv:2401.03972

[4] Cleynen A. and Ravindran A. and Kumar B. and Sethi A.J. and Wondward k. and Eyras E. and Shirokikh N. *INtegrity and DEGradation of RNA Analysis.* (2024) Manuscript in preparation github.com/Arnaroo/INDEGRA

[5] Cleynen A. and Prodic S. and Srivastava A. and Mahmud S. and Kanchi M. and Ravindran A. and Shirokikh N. and Eyras E. *Single-molecule Workflow for Analysing RNA Modifications* (2024) Manuscript in preparation . github.com/comprna/SWARM

Chapter 1

A probabilist framework for patient relapse detection, and treatment allocation

This chapter relates the work exposed in three papers: the first, with Benoîte de Saporta, defines an online strategy to identify the regime changes in a simple piece-wise deterministic Markov process (PDMP) when the jumps occur between observation dates. The second, also with Benoîte de Saporta, extends this approach to controlled PDMP: the strategies taken by the actionner now influence the dynamics of the process, which must be controlled over the finite horizon. Finally, the last section, in collaboration with Benoîte de Saporta, Aymar Thierry D'Argenlieu and Régis Sabbadin, describes an IA strategy to control PDMPs via a simulation-based algorithm .

1.1 Monitoring patients throughout their journey with the disease

Cancer is a group of genetic diseases involving abnormal cell growth with the potential to invade or spread to other parts of the body. Several treatment types exist, such as chemotherapy, radiation therapy, immune therapy, etc, that will in most cases bring the cancer patient into a (possibly partial) remission state. Unfortunately, many cancer types are susceptible to relapse, or recurrence, meaning that the cancer might come back after treatment, either in the first place it originally started or somewhere else in the body. It is not possible to say how likely a cancer is to recur, but cancer is harder to treat and more likely to come back if it is fast growing and more advanced (National Cancer Institute 2024). In order to detect those relapses as early as possible, cancer patients are proposed a follow-up care plan, a check-up agenda that will typically include bloodwork, as well as other tests and procedures.

Multiple Myeloma (MM) is the second most common haematological malignancy in the world and is characterised by the accumulation of malignant plasma cells in the bone marrow. Classical treatments are based on chemotherapies, which, if appropriate, act fast and efficiently bring MM patients to remission in a few weeks. However almost all patients eventually relapse more than once and the five-year survival rate is around 50%. Immunoglobulins are produced by plasma cells, and clonal plasma cell proliferative diseases such as MM usually secrete a monoclonal immunoglobulin (M-protein) that can be used as a serologic "tumour" marker. The secreted proteins can be used as a diagnostic tool for the identification of the clone of plasma cells as well as a quantitative marker to follow the course of the disease and response to therapy. A typical follow-up routine in Multiple Myeloma consists in frequent blood-test screening to perform Serum protein electrophoresis. Electrophoresis separates proteins based on their physical properties, and the subsets of these proteins are used in interpreting the results. In particular, a homogeneous spike-like peak in a focal region of the gamma-globulin zone indicates a monoclonal gammopathy, and the peak may be used to quantify the level of clonal plasma cells in the blood (O'connell, Horita, and Kasravi 2005; Willrich and Katzmann 2016).

Throughout my long-going collaboration with Pr Hervé Avet-Loiseau, PUPH at the *Centre de Recherche en Cancérologie de Toulouse (CRCT)*, and oncologist in MM, I have obtained data from the Intergroupe Francophone du Myélome 2009 clinical trial (Attal et al. 2017) which has followed 748 French MM patients from diagnosis to their first relapse on a standardised protocol for up to six years. At each visit a blood sample was obtained to evaluate the amount of monoclonal immunoglobulin protein in the blood. An example of patient data-set is given in Figure 1.1.

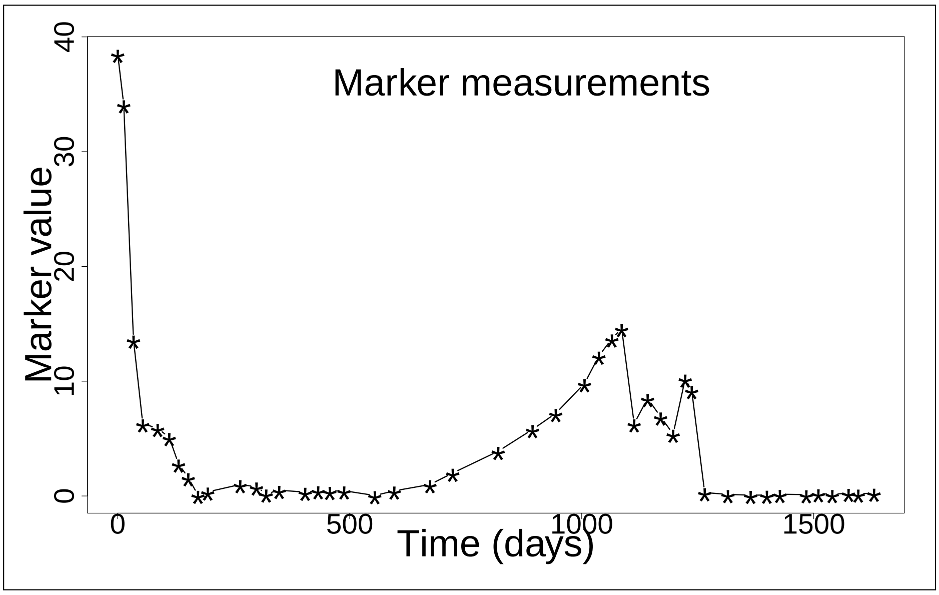

Figure 1.1: Example of patient follow-up data. Marker values are measured at each patient visits over a certain period of time. Data from the *Intergroupe Francophone du Myélome* 2009 clinical trial, courtesy of the *Centre de Recherche en Cancérologie de Toulouse.*

These data are used by practitioners as the first indication of a potential relapse, which are later confirmed by more expensive and invasive tests such as MRIs. The general idea is that when the marker measurements are low, the patient is in remission, whereas when the marker level increases, the patient may be relapsing, with some more or less aggressive clone.

The general goal of this first chapter is to deliver an automated tool to help practitioners decide on the relapse status of a patient, guide the treatment choice, and optimise the follow-up plan, that is the visit dates at the hospital. One of the major specifications is to allow the practitioner to make their own decision without impacting the ability of the tool to suggest new decisions at ulterior decision times.

One of the main requirements for such task is the definition of a universal model adapted to patient-specific parameters that could describe in an exhaustive manner the possible consequences of the practitioner's decisions. Mathematical models have been developed to link the tumour markers to tumour sizes (Lutz et al. 2008), or to predict evolution of tumour growth from initial measurements (Nicolò et al. 2020; J. Xu, Vilanova, and Gomez 2016), but online adaptive models predicting relapses and automating treatment strategies are still lacking. In particular, such a model should be able to reconcile the continuous time evolution of the disease, continuous values for the markers leading to any possible values within a given range, and the noisy observations at discrete visit dates. A good candidate is the class of PDMP (Davis 1984; Cloez et al. 2017; Rudnicki and Tyran-Kamińska 2017). Indeed, PDMPs are non diffusive hybrid stochastic processes that can handle both continuous and discrete variables and their interactions in continuous time. The only source of stochasticity comes from the jumps of the process. They are thus simple to simulate and easy to interpret. Controlled PDMPs allow continuous time dynamics on continuous (or hybrid discrete and continuous) state spaces with decisions taken in continuous or discrete time (Davis 1993; de Saporta, Dufour, and Zhang 2015).

The second requirement is the ability to define an optimisation function associated to the mathematical model that can encode the diverse impacts of treatment on the patient's quality of life. Defining this impact is a difficult task as it will typically depend on the treatment's side effects, the number of visits, the burden of living with a disease and the remaining life expectancy. Finally, the last requirement is the ability to actually numerically optimise this function and provide an optimal strategy for the follow-up of the patient.

The contributions discussed in this chapter aim at addressing those challenges. I first present PDMPs and their controlled version in a generic context, and describe how they can be formulated as Markov Decision Processes when the control is designed to occur in discrete time. I then discuss several contributions to provide an automated tool in two different contexts, the first when the observer is not allowed to influence the dynamics of the marker, and simply wants to detect relapse, the second when they aim at controlling the process so that the marker values remain as close as possible to a given reference value. In both situations, I discuss the choice of the optimisation function and describe the strategies to provide an explicit optimal strategy. I end the chapter by briefly discussing future projects to relax some of the assumptions of the model, allowing for some patient-specific parameters that can be learned while controlling the trajectory, and to include additional data such as measurements of additional markers, or clinical data sampled at rare and discrete times.

1.2 Piece-wise deterministic Markov Processes, and their controlled version

This section is a shortened and adapted version of an ongoing exposition paper written with my two PhD students, Orlane Rossini and Amélie Vernay, and their co-supervisors, Benoîte de Saporta and Régis Sabbadin, in which we describe the probabilistic framework of PDMPs (and their controlled version) and their counterpart in the IA community, POMDPs, BAMDPs, etc.

1.2.1 Piece-Wise Deterministic Markov Processes

Generic definition in $\mathbb{R}^d$. Let E be a finite union of Borel subsets of $\mathbb{R}^d$, for some $d \geq 1$ and $\mathcal{E}$ its Borel σ-field. A Piecewise Deterministic Markov Process (PDMP) $X = (X_t)_{t\geq 0}$ on $(E, \mathcal{E})$ can be described by means of three local characteristics, namely a flow $\Phi(x,t)$, a jump rate $\lambda(x)$, and a Markov kernel $Q(x, dx')$. The PDMP dynamic can be described informally as follows: starting from some initial point $x \in E$, the motion of the process follows the flow $\Phi(x,t)$ until a first jump time T_1, which may be deterministically triggered by the process reaching the boundary ∂E of E, or may occur randomly with hazard rate $\lambda(\Phi(x,t))$. At T_1 the process jumps to a new point $x' = X_{T_1}$ selected with probability $Q(\Phi(x,T_1), dx')$ (conditional on T_1), and the motion restarts from this new point as before.

It is particularly convenient, when dealing with PDMP, to separate the state space E and states x into discrete and Euclidean components. In this case the state space is called hybrid. Let M be a finite set, and for all $m \in M$, let E_m be some Borel subset of $\mathbb{R}^{d_m}$, which dimension is allowed to change with m. The state space is thus

$$E = \bigcup_{m\in M} \{m\} \times E_m.$$

Any state $x \in E$ can be written as $x = (m, \zeta)$ with $m \in M$ and $\zeta \in E_m$. The discrete component m is called the mode or regime and ζ is called the Euclidean component. In between jumps, mode m remains constant and only ζ evolves through the flow. At jump times, both the mode and the Euclidean component may be allowed to change through the Markov kernel Q.

Let ∂E be the boundary of E: $\partial E = \cup_{m\in M}\{m\} \times \partial E_m$, $\bar{E} = E \cup \partial E$ its closure, and $\mathcal{B}(\bar{E})$ the Borel σ-field on $\bar{E}$. The local characteristics verify the following properties:

- The flow Φ is a continuous function from $E \times \mathbb{R}_+$ onto $\bar{E}$ satisfying a semi-group property, *i.e.* $\Phi(\cdot, t+s) = \Phi(\Phi(\cdot,t), s)$ for all $s, t \in \mathbb{R}_+$. In the hybrid setting, we set $\Phi(x,t) = (m, \Phi_m(\zeta, t))$ for all $x = (m,\zeta) \in E$ and $t \in \mathbb{R}_+$. The flow prescribes the deterministic trajectory between jumps. Note that the mode is always constant between jumps, only the Euclidean variable is allowed to change.
- The jump intensity, or hazard rate, or risk function λ is a measurable function from $\bar{E}$ onto $\mathbb{R}_+$ such that for any x in E, there exists $\epsilon > 0$ such that

$$\int_0^\epsilon \lambda(\Phi(x,t))dt < +\infty,$$

forbidding instantaneous jumps. Let $t^*(x)$ be the deterministic time the flow takes to reach the boundary ∂E of E when it starts from x:

$$t^*(x) = \inf\{t > 0 : \Phi(x,t) \in \partial E\}.$$

Starting from x at time 0, the first jump time T_1 has the following distribution

$$\mathbb{P}_x(T_1 > t) = \mathbb{P}_x(T_1 > t | X_0 = x) = \exp\left(-\int_0^t \lambda(\Phi(x,s))ds\right) \mathbb{1}_{t<t^*(x)}, t \geq 0.$$

For convenience, we also define

$$\Lambda(x,t) = \int_0^t \lambda(\Phi(x,s))ds.$$

This implies that jumps may occur via two means: *random jumps* occur from the realisation of the random clock with intensity λ (e.g. time T_1 on Figure 1.2), while *deterministic jumps* occur when the process reaches the boundary of the state space (e.g. time T_2 on Figure 1.2). In both cases, the value of the process at the jump-time is selected from kernel Q.

• Finally, the Markov kernel Q on $(\bar{E}, \mathcal{B}(\bar{E}))$ selects the new location after each jump. It satisfies $Q(x, \{x\}) = 0$ so that a jump has to change the location of the process (either the mode, or the Euclidean variable or both). After the first jump time T_1, the process restarts from a new state selected by Q and follows the flow until the next jump time T_2. The sojourn time $T_2 - T_1$ until the next jump is drawn from the jump intensity and time to reach the boundary as before. At the jump time T_2, the post-jump location is selected by the Markov kernel Q and so on iteratively.

A process X defined by a flow Φ, an intensity function λ and a kernel Q verifying the above properties is well defined and is strong Markov, as is for instance shown in Jacobsen 2006.

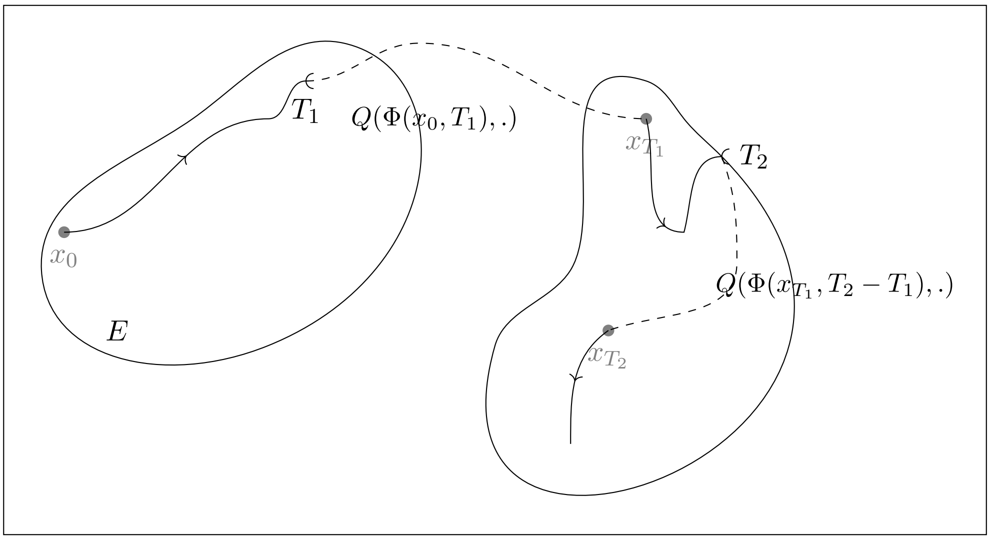

Figure 1.2: Generic representation of a PDMP: starting from an initial value x_0, the process follows a deterministic trajectory until a jump-point occurs, either at a random time (as at T_1) or because the process reaches the state boundary (as at T_2). At jump-times, the process jumps to a new location drawn from kernel Q.

A very interesting property of PDMPs is that they are particularly easy to simulate due to their intrinsic iterative construction, as shown in Algorithm 1. This will come particularly handy for all numerical approaches relying on simulations, such as discretisation by quantisation (see for instance Section 1.3), tree exploration by simulations (see for instance Section 1.4), or reinforcement learning via neural networks (see for instance Section 1.5).
Computing the trajectory along the flow (Alg. 1, line 5) is straightforward if the flow is explicit. Computing the post-jump location from the kernel Q (Alg. 1, line 9) is also usually straightforward. In most applications, the kernel only changes the mode and leaves the Euclidean variable unchanged, or changes the Euclidean variable through some known mapping depending only on its value at S^-, the instant before the jump-time. The only challenging step is then to simulate the jump dates from the non constant intensity (Alg. 1, line 3). We present here an exact simulation method called the Stochastic Simulation Algorithm (SSA), or Gillespie algorithm (Gillespie 1977)

that is especially suitable for PDMPs. The interested reader may consult Cocozza-Thivent 1997; Saporta, Dufour, and H. Zhang 2016 for an alternative algorithm to simulate only sojourn-times and post-jump locations instead of full trajectories, and Riedler 2013; Veltz 2015 and references therein for a broader discussion of alternative exact and approximate simulation methods.

Algorithm 1 Simulation of a trajectory of a PDMP between times t and $t+r$ from state $x = (m, \zeta)$

```
$s \leftarrow 0$
while $s < r$ do
    $S \sim \lambda(x)$
    $S \leftarrow \min\{S, t^*(x)\}$
    $X_{t+u} \leftarrow \Phi(x, u)$ for $0 \leq u \leq \min\{(s+S), r\}$
    $\zeta \leftarrow \Phi(\zeta, s)$
    if $s + S \leq r$ then
        $s \leftarrow s + S$
        $(m', \zeta') \sim Q(\cdot|x)$
        $m \leftarrow m', \zeta \leftarrow \zeta', x \leftarrow (m, \zeta)$
    end if
end while
```

The SSA is based on the Markov property and the following property of the minimum of independent exponential distributions.

Lemma: Let S_1 and S_2 be two independent random variables with respective (non-constant) intensity functions λ_1 and λ_2. Set $S = \min\{S_1, S_2\}$ and $I = \arg\min\{S_1, S_2\}$. Then the intensity of S is $\lambda_1 + \lambda_2$ and conditionally to $S = t$, the distribution of I is $(\frac{\lambda_1(t)}{\lambda_1(t)+\lambda_1(t)}, \frac{\lambda_2(t)}{\lambda_1(t)+\lambda_1(t)})$.

Proof. First, notice that if S_i has intensity λ_i, then its density function is $\lambda_i(t)e^{-\int_0^t \lambda_i(u)du}$. Using the independence between S_1 and S_2, one obtains the joint distribution of (S, I) as follows

$$\begin{aligned}
\mathbb{P}(S > t, I = 1) &= \mathbb{P}(S_1 > t, S_2 > S_1) \\
&= \int_t^\infty \lambda_1(x)e^{-\int_0^x \lambda_1(u)du} \int_x^\infty \lambda_2(y)e^{-\int_0^y \lambda_2(u)du} dy dx \\
&= \int_t^\infty \lambda_1(x)e^{-\int_0^x \lambda_1(u)du} e^{-\int_0^x \lambda_2(u)du} dx \\
&= \int_t^\infty \frac{\lambda_1(x)}{\lambda_1(x)+\lambda_2(x)}(\lambda_1(x)+\lambda_2(x))e^{-\int_0^x(\lambda_1(u)+\lambda_2(u))du} dx.
\end{aligned}$$

Hence the density of S is $(\lambda_1(x)+\lambda_2(x))e^{-\int_0^x(\lambda_1(u)+\lambda_2(u))du}$, and

$$\mathbb{P}(I = 1|S = t) = \frac{\lambda_1(x)}{\lambda_1(x)+\lambda_2(x)},$$

and the result follows. □

The principle underlying the SSA algorithm is as follows. The aim is to sample from a random variable S_1 with (non-constant) intensity λ_1. Suppose that λ_1 is bounded from above by some constant $\overline{\lambda}$ and let S be random variable with (regular) exponential distribution with parameter $\overline{\lambda}$, so that is is very easy to sample from S. In view of the Lemma, S can be interpreted as the minimum of S_1 and some extra random variable S_2 with intensity $\lambda_2 = \overline{\lambda} - \lambda_1$. Let t be a sample value from S, then t can be considered as a value sampled from S_1 with probability

Algorithm 2 Stochastic Simulation Algorithm for a PDMP starting from state $x_0 = (m_0, \zeta_0)$ at time 0 and up to time horizon H

```
 1: $x \leftarrow x_0$
 2: $t \leftarrow 0$
 3: while $t < H$ do
 4:    $S \sim \text{Exp}(\overline{\lambda})$
 5:    $X_{t+s} \leftarrow \Phi(x, s)$ for $0 \leq s < \min\{S, t^*(x)\}$
 6:    if $S > t^*(x)$ then
 7:       $x' \sim Q(\Phi(x, t^*(x)), \cdot)$
 8:    else
 9:       $U \sim U[0, 1]$
10:       if $U \leq \frac{\lambda(S)}{\overline{\lambda}}$ then
11:          $x' \sim Q(\Phi(x, S), \cdot)$
12:       else
13:          $x' \leftarrow \Phi(x, S)$
14:       end if
15:    end if
16:    $x \leftarrow x'$
17:    $t \leftarrow t + \min\{S, t^*(x)\}$
18: end while
```

$\frac{\lambda_1}{\overline{\lambda}}$. This simple fact leads to a rejection algorithm where one repeatedly samples from S until the result is accepted as a sample from S_1. In a Markov context, it is even more efficient as by construction $S_1 \geq S$. Hence all samples are used, some are accepted as jump times (boundary jump: Alg. 2 line 6 or random jump: Alg. 2 line 10), the others just move the process forward without creating a jump (Alg. 2 line 12). The procedure then restarts from the new location, as detailed in Algorithm 2.

The choice of the upper bound $\overline{\lambda}$ can have a strong impact on the speed of the algorithm. If it is too high, it will keep proposing very short times and move forward extremely slowly. If at all possible, one should choose a tight upper bound for the intensity.

PDMP example in the medical context. The marker dynamics of a patient can be modelled by a PDMP $X_t = (m_t, x_t, u_t)$ defined on a state space $E = \mathcal{M} \times \mathbb{K} \times \mathbb{R}_+$, where $\mathcal{M} = \{0, 1, \ldots, d\}$ is the finite set of modes, 0 corresponding to a remission status, and 1 to d to different possible diseases states, $\mathbb{K}$ is a compact subset of $\mathbb{R}$ representing the position of the process, *i.e.* the set of possible values of the blood marker, and the third coordinate is the running time since the last jump, needed to ensure the process is Markovian.
Starting from point $(0, x, 0)$ in E, which means starting from remission status (mode 0) and position x at time 0, we assume that there can be only one jump time T (relapse time of the patient) of the process which has distribution

$$\mathbb{P}(T > t | X_0 = (0, x, 0)) = e^{-\int_0^t \lambda(s) ds}.$$

For $t \in [0, T)$, we will assume that $X_t = (0, \zeta_0, t)$, which is equivalent to assuming that the flow in mode 0 is constant and equal to $\zeta_0 \in \mathbb{K}$. At time T, the process selects a new mode $i \in \{1, \ldots, d\}$ with positive probability π_i.
Finally, for $t \geq T$, $\boldsymbol{X}_t = (i, \zeta_t, t - T)$, where $\zeta_t = \Phi_i(x_T, t - T)$ for some exponential flow $\Phi_i : \mathbb{K} \times \mathbb{R}_+ \to \mathbb{K}$ such that $\Phi_i(x, t) = xe^{v_i t}$ for some positive constant v_i. An example of such trajectory is given in Figure 1.3

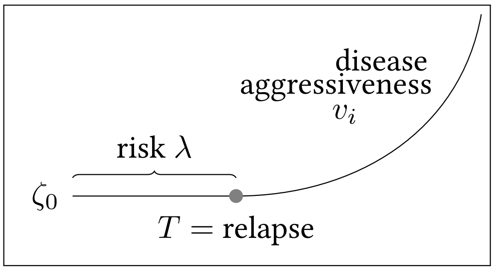

Figure 1.3: example of PDMP model, representation of the marker level of a patient. The risk function λ controls the time to relapse, while parameter v_i controls the aggressiveness of the disease.

1.2.2 Impulse control for PDMPs

Generic definition. The notion of controlled PDMP is quite tricky to define, as the process does not exist without a control strategy, and the control strategies themselves are typically defined from the (partial) observation of the controlled PDMP. In this section I will try to present a class of controlled PDMPs that I have extensively studied in my works with Benoîte de Saporta, and later with Régis Sabbadin, Aymar Thierry d'Argenlieu, and Orlane Rossini. To do so, I will further decompose the discrete state space $\mathcal{M}$ of a PDMP into a two-dimensional finite set $L \times M$, where modes in L can be controller chosen while modes in M cannot. For all regimes (or modes) (ℓ, m) in $\mathcal{M}$, let E^ℓ_m be an open subset of E so that $E = \{(\ell, m, \mathrm{x}), \ell \in L, m \in M, \mathrm{x} \in E^\ell_m\}$, and $E^\ell = \{(m, \mathrm{x}), m \in M, \mathrm{x} \in E^\ell_m\}$. Once again, a controlled PDMP on the state space E is determined by three local characteristics:

- the flow $\mathbf{\Phi}(x, t) = (\ell, m, \Phi^\ell_m(\mathrm{x}, t))$ for all $x = (\ell, m, \mathrm{x})$ in E and $t \geq 0$, where $\Phi^\ell_m : E^\ell_m \times \mathbb{R}_+ \to \mathbb{R}^d$ is continuous and satisfies a semi-group property $\Phi^\ell_m(\cdot, t+s) = \Phi^\ell_m(\Phi^\ell_m(\cdot, t), s)$, for all $t, s \in \mathbb{R}_+$. It describes the deterministic trajectory between jumps. Let $\boldsymbol{t}^*(x) = t^{*\ell}_m(\mathrm{x})$ be the deterministic time the flow takes to reach the boundary of E when it starts from $x = (\ell, m, \mathrm{x})$:
$$\boldsymbol{t}^*(x) = \inf\{t > 0 : \Phi^\ell_m(\mathrm{x}, t) \in \partial E^\ell_m\}.$$
- the jump intensity $\boldsymbol{\lambda}(x) = \lambda^\ell_m(\mathrm{x})$ for all $x = (\ell, m, \mathrm{x})$ in E, where $\lambda^\ell_m : \bar{E}^\ell_m \to \mathbb{R}^+$ is a measurable function such that for any $x = (\ell, m, \mathrm{x})$ in E, there exists $\epsilon > 0$ such that
$$\int_0^\epsilon \lambda^\ell_m(\Phi^\ell_m(\mathrm{x}, t))dt < +\infty.$$
For all $x = (\ell, m, \mathrm{x})$ in E and $t \in [0; \boldsymbol{t}^*(x)]$, set
$$\mathbf{\Lambda}(x, t) = \Lambda^\ell_m(\mathrm{x}, t) = \int_0^t \lambda^\ell_m(\Phi^\ell_m(\mathrm{x}, s))ds.$$
- the Markov kernel $\boldsymbol{Q}$ on $(\mathcal{B}(\bar{E}), \bar{E})$ represents the transition measure of the process and allows to select the new location after each jump. It satisfies for all $x \in \bar{E}$, $\boldsymbol{Q}(\{x\} \cup \partial E | x) = 0$. We also write $Q^\ell_m(\cdot|\mathrm{x}) = \boldsymbol{Q}(\cdot|x)$ for all $x = (\ell, m, \mathrm{x}) \in E$ and add the additional constraint that Q cannot change the value of ℓ, as ℓ is intended to be controller-chosen, i.e. Q^ℓ_m sends E^ℓ onto itself.

A general impulse strategy $\mathcal{S} = (\ell_n, r_n)_{0 \leq n \leq N-1}$ is a sequence of non-anticipative E-valued random variables on a measurable space $(\Omega, \mathcal{F})$ and of non-anticipative intervention lapses. This means that on the one hand, the controller may act on the process by changing both its regime (ℓ, m) and value ζ via intervention ℓ_n, and on the other hand may select the lapse between consecutive interventions belongs via the time decision r_n. In what follows, we will only consider a slightly easier case where ℓ_n takes values in L, hence the controller can only act on the process by changing its regime, i.e. by selecting the local characteristics to be applied until the next intervention, and r_n is a multiple of fixed minimal lapse δ (and δ is not supposed to be small): $r_n \in \mathbb{T}$ where $\mathbb{T}$ is a subset of $\delta^{1:N} = \{\delta, 2\delta, \dots, N\delta\}$ that contains δ, *i.e.* the lapse between consecutive interventions belongs to the finite set $\mathbb{T}$.

The trajectories of the PDMP controlled by strategy $\mathcal{S}$ are constructed recursively between intervention dates τ_n (defined recursively by $\tau_0 = 0$ and $\tau_{n+1} = \tau_n + r_n$) as described in Algorithm 3.

Algorithm 3 Simulation of a trajectory of a controlled PDMP between interventions n and $n+1$ from state $x_n = (\ell_n, m_n, \mathtt{x}_n)$

```
1:  $s \leftarrow 0$
2:  $\ell \leftarrow \ell_n, m \leftarrow m_n, \mathtt{x} \leftarrow \mathtt{x}_n, x \leftarrow (\ell, m, \mathtt{x})$
3:  while $s < r_n$ do
4:      $S \sim \lambda_m^\ell(\mathtt{x})$
5:      $S \leftarrow \min\{S, \boldsymbol{t}^*(x_n)\}$
6:      $X_{\tau_n + t} \leftarrow \boldsymbol{\Phi}(x, t)$ for $0 \leq t \leq \min\{(s+S), r_n\}$
7:      $\mathtt{x} \leftarrow \Phi_m^\ell(\mathtt{x}, s)$
8:      if $s + S \leq r_n$ then
9:          $s \leftarrow s + S$
10:         $(\ell, m', \mathtt{x}') \sim Q_m^\ell(\cdot|\mathtt{x})$
11:         $m \leftarrow m', \mathtt{x} \leftarrow \mathtt{x}', x \leftarrow (\ell, m, \mathtt{x})$
12:     end if
13: end while
```

In Line 5 of Algorithm 3, $S \sim \lambda_m^\ell(\mathtt{x})$ means that S has the survival function

$$\mathbb{P}_x(S > t) = e^{-\int_0^t \lambda_m^\ell(\Phi_m^\ell(\mathtt{x},s))ds} = e^{-\boldsymbol{\Lambda}(x,t)}.$$

As a boundary jump can also occur, the distribution of the next jump time T, starting from x is given by the survival function

$$\mathbb{P}_x(T > t) = e^{-\int_0^t \lambda_m^\ell(\Phi_m^\ell(\mathtt{x},s))ds}\mathbb{1}_{t<\boldsymbol{t}^*(x)} = e^{-\boldsymbol{\Lambda}(x,t)}\mathbb{1}_{t<\boldsymbol{t}^*(x)}.$$

Controlled PDMP example in the medical context. We consider again a modelisation of the the marker dynamics of a patient by a PDMP where this time we allow treatments to act on the process. The process $X_t = (\ell_t, m_t, \zeta_t, u_t)$ has four components. The mode m corresponds to the overall state of the patient ($m = 0$: sound, $m = 1$: disease 1, $m = 2$: disease 2, $m = 3$: death of the patient). The control ℓ correspond to the medical treatment ($\ell = \emptyset$: no treatment, $\ell = a$: efficient for disease 1 and slows the progression of disease 2, $\ell = b$: efficient for disease 2 and slows the progression of disease 1). The variable ζ still corresponds to the blood marker value, ζ_0 being the nominal value for a sound patient, and we further assume that if the marker reaches level D, the patient dies. The variable u corresponds to the time since the last change-point occurred. Decision dates correspond to visits to the medical centre when the marker is measured and a new treatment is selected and applied until the next visit, with possible visits every 15, 30 or 60 days ($\delta = 15$).

If treatment $\emptyset$ is applied, the patient may randomly jump from $m = 0$ to any of the two disease states $m \in \{1, 2\}$. In the disease states ($m = 1$ or $m = 2$), the marker level grows exponentially and can reach the death level D in finite time, no other change of state is possible. If treatment a is applied, the patient may only randomly jump from $m = 0$ to the other disease state $m = 2$. In the disease state $m = 1$, the marker level decreases exponentially and can reach the nominal level ζ_0 in finite time or randomly jump to the other disease state $m = 2$. In the disease state $m = 2$, the marker is level grows exponentially and can reach the death level D in finite time, no other change of state is possible. Effects of treatment b is similar: exponential decrease of the marker in disease $m = 2$, exponential increase in disease $m = 1$. An example of trajectory is illustrated in Figure 1.4.

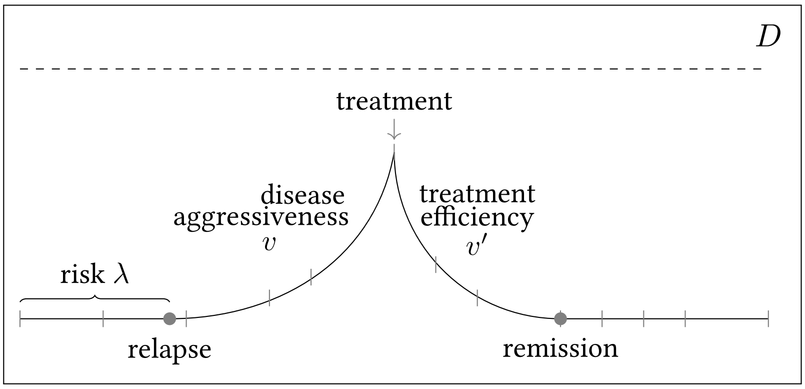

Figure 1.4: Example of controlled PDMP model in the medical framework: the risk function λ controls the time to relapse, while parameters v and v' control the aggressiveness of the disease and the efficiency of the treatment respectively.

1.2.3 Partially observed (controlled) PDMPs

Now that the generic PDMP and its more complex counterpart controlled PDMP are defined, we can define the observed and unobserved variables in the applications. In all our work, we have assumed that the (controlled) PDMP X_t is completely unobserved, but its Euclidean variable ζ is observed through noise: at time τ_n, if the state of the process is $x = (\ell, m, \zeta, u)$, the observer or controller receives observation $y_n = F(\zeta) + \epsilon_n$ where (ϵ_n) are real-valued independent and identically distributed random variables with density f independent from the controlled PDMP. We further assume that the random variables Y_n take values in a compact interval I of the real line.

Those observations are not achievable in continuous time, but either on a fixed agenda (typically a regular grid of size δ in the simple PDMP scenario), or on observation times defined by the controller (via decision r_n at time τ_n, the possible values of r_n typically being limited to a few values multiple of δ) in the controlled PDMP case. Moreover, the controller may have access to two additional observations at each collection time: the observation z_n of a subset of the modes, $\bar{m} \in M$ (typically, in our medical context, the practitioner will know if the patient is still alive), and the elapsed time since the beginning of the process (the follow-up time in the medical context) w_n. We will denote O_n the set of observations available at time τ_n, *i.e.* $O_n = (Y_n, Z_n, W_n)$.

By denoting $t_0, \ldots, t_n$ the observation dates, $X_0, \ldots, X_n$ the process values and $O_0, \ldots, O_n$ the observation values at those observation dates, we argue (and prove in the papers) that X_n is a Markov chain with kernel P, and (X_n, O_n) is a Markov chain with kernel R.

1.2.4 Optimisation framework

In applications, PDMP models are mostly studied in two different contexts: inference of the parameters, and optimisation of trajectories. Inference in PDMP models is not an easy task, in particular in the case of partially observed PDMPs. One may refer to Azais and Bouguet 2018 for the estimation of the flow parameters, or to Krell and Schmisser 2021 for the estimation of the jump rates.

Here on the other hand, we will assume that all parameters of the models are perfectly known to the user (a discussion about relaxing this assumption is proposed in Section 1.5) and set our contributions in the optimisation context. We present the framework and our work in the case of optimisation / control over a finite horizon, which we denote H in the sequel, but most of our contributions would remain valid in the infinite horizon case.

The strategy $\mathcal{S}$ induces a family of probability measures $\mathbb{P}_x^{\mathcal{S}}$, $x \in E$, on a suitable probability space $(\Omega, \mathcal{F})$. Associated to strategy $\mathcal{S}$, we can define the following expected total cost for a process starting at $x \in E$

$$\mathcal{J}(\mathcal{S}, x) = \mathbb{E}_x^{\mathcal{S}}\Big[\int_0^H c_r(X_s)ds + \sum_{n=0}^{N-1} c_i(X_{t_n}, \ell_n, r_n) + c_t(X_H)\Big],$$

where $\mathbb{E}_x^{\mathcal{S}}$ is the expectation with respect to $\mathbb{P}_x^{\mathcal{S}}$, c_r is some running cost, c_i some intervention cost and c_t some terminal cost.

The last ingredient needed to state the optimisation problem is to define admissible strategies. Informally, decisions can only be taken in view of some discrete-time noisy observations of the process, instead of the exact value of the process at all times. The rigorous definition will be given in the next section in the framework of POMDPs. Denoting by $\mathbb{S}$ the set of all admissible strategies, the aim is to compute the value function

$$\mathcal{V}(x) = \inf_{\mathcal{S} \in \mathbb{S}} \mathcal{J}(\mathcal{S}, x),$$

and explicitly construct an optimal strategy.

One may note that in the AI community, the literature most often defines a *reward* function instead of a cost function, leading to a value function to be maximised. Both approaches are of course perfectly equivalent, and I will only focus here on the cost minimisation framework.

1.2.5 Partially observed Markov decision process

In this section, we show how impulse control problems for PDMPs with discrete decision dates may be framed in the context of Partially observed Markov decision processes (POMDPs), and argue that the reasoning holds in the more simple case of generic PDMPs, even though the approach may be more straightforward. We will add some constraint on the decision space and cost functions to best fit our medical example, but they cause no loss of generality.

Finding a suitable rigorous way to state an impulse control problem for PDMPs with hidden jumps and under noisy observation is by no means straightforward, especially as regards defining admissible strategies, see e.g. Almudevar 2001 or Costa and Dufour 2013, sec. 1.1. In the sequel, we drop the regime $\ell \in L$ from the state x and include it in the action instead to better fit the standard POMDP notation. Denote $E_m = \cup_{\ell \in L} E_m^\ell$, $E_{<\bar{m}} = \cup_{m \neq \bar{m}} E_m$ and $E_M = \{(m, \mathrm{x}), m \in M, \mathrm{x} \in E_m\}$. (We remind the reader that $\bar{m}$ is the subset of modes that are observed by the controller through random variable Z.) Define on E_M the following hybrid distance, for $x = (m, \zeta)$ and $x' = (m', \zeta') \in E_M$,

$$\|x - x'\| = (\|\zeta - \zeta'\|)\mathbb{1}_{m=m'} + \infty \mathbb{1}_{m \neq m'}.$$

Let $(\mathbb{X}, \mathbb{A}, \mathbb{K}, R, c, C)$ be the POMPD with the following characteristics.

• The state space is $\mathbb{X} = \{\xi = (x, y, z, w) \in (E_M \times \mathbb{O}); z = \mathbb{1}_{\{x_1 = \bar{m}\}}\} \cup \{\Delta\}$, with $\mathbb{O} = \{\gamma \in I \times \{0,1\} \times \delta^{1:N}\}$ the observation space. It gathers the values X_n of the hidden PDMP at dates τ_n as well as the observation processes Y_n, and the two additional observed variables Z_n and W_n. State Δ is a cemetery state where the process is sent after the horizon H or mode $\bar{m}$ is reached.

• The action space is $\mathbb{A} = (L \times \mathbb{T}) \cup \{\check{d}\}$, where $\check{d}$ is an empty decision that is taken when the horizon H or mode $\bar{m}$ is reached. This purely technical decision sends the process to the cemetery state Δ.

• The constraints set $\mathbb{K} \subset \mathbb{X} \times \mathbb{A}$ is such that its sections $\mathbb{K}(\xi) = \{d \in \mathbb{A}; (\xi, d) \in \mathbb{K}\}$ satisfy $\mathbb{K}(\Delta) = \{\check{d}\}$ and $\mathbb{K}(x, y, z, w) = \mathbb{K}(z, w)$ for $(x, y, z, w) \in \mathbb{X} - \{\Delta\}$ as decisions are taken in view of the information from the observations only. In addition, we impose

- $\mathbb{K}(0, w) = L \times (\mathbb{T} \cap \{\delta, 2\delta, \dots, N\delta - w\})$ if $w \leq \delta(N-1)$, to force the last intervention to occur exactly at the horizon time H, unless mode $\bar{m}$ ($z = 1$) has been reached,
- $\mathbb{K}(0, N\delta) = \mathbb{K}(1, w) = \check{d}$: no intervention is possible after the horizon or the top event has been reached.

Note that as $\delta \in \mathbb{T}$, the set $\mathbb{T} \cap \{\delta, 2\delta, \dots, N\delta - w\}$ is never empty.

• The controlled transition kernels R are defined as follows: for any bounded measurable function g on $\mathbb{X}$, any $\xi \in \mathbb{X}$ and $d \in \mathbb{K}(\xi)$, one has

$$\begin{aligned}
Rg(\xi, d) &= g(\Delta), \quad \text{if } d = \check{d}, \\
Rg(\xi, d) &= \int_I \int_{E_{<\bar{m}}} g(x', y', 0, w + r) f(y' - F(x')) P(dx'|x, d) dy' \\
&\quad + \int_{E_{\bar{m}}} g(x', 0, 1, w + r) P(dx'|x, d), \quad \text{if } \xi = (x, y, 0, w),\ d = (\ell, r) \neq \check{d}, \\
Rg(\xi, d) &= 0, \quad \text{otherwise},
\end{aligned}$$

where $P(\cdot|x, d)$ is the distribution of X_r conditionally to $X_0 = x$ under regime ℓ, if $d = (\ell, r)$. While it is possible to derive the explicit analytical form of P, we choose not to expose the computations here. The interested reader may find it in Cleynen and de Saporta 2023.

• The terminal cost function $C : \mathbb{X} \to \mathbb{R}_+$ satisfies $C(\Delta) = 0$ and $C(x, y, z, w) = c_t(x)$ for $(x, y, z, w) \in \mathbb{X} - \{\Delta\}$ with $c_t(x) = c_{\bar{m}}$ if $x \in E_{\bar{m}}$,where $c_{\bar{m}}$ is a penalty for reaching the top value, *i.e.* in our context we suppose that only the underlying process (and not the observations) has an effect on the cost. This constraint is not necessary and results will hold in the general case.

• The non-negative cost-per-stage function $c : \mathbb{K} \times \mathbb{X} \to \mathbb{R}_+$ satisfies $c(\Delta, \check{d}, \cdot) = 0$, $c(\xi, \check{d}, \cdot) = C(\xi)$ and $c(\xi, d, \xi') = c_{\bar{m}}$ if $x' \in E_{\bar{m}}$. Ideally, to match the definition of the expected total cost of a strategy $\mathcal{S}$, for $d = (\ell, r)$ one should choose c as

$$c(x, d) = \mathbb{E}_x^\ell \Big[\int_0^r c_r(X_s) ds \Big] + c_i(x, d).$$

However, the integral part has no simple analytical expression and we chose some simpler proxy instead, see next Sections.

• The optimisation horizon is finite and equals N corresponding to date $N\delta = H$.

Classically, the sets of observable histories are defined recursively by $H_0 = \mathbb{O} \cup \{\Delta\}$ and $H_n = H_{n-1} \times \mathbb{A} \times (\mathbb{O} \cup \{\Delta\})$. A decision rule at time n is a measurable mapping $g_n : H_n \to \mathbb{A}$ such that $g_n(h_n) \in \mathbb{K}(\gamma_n)$ for all histories $h_n = (\gamma_0, d_0, \gamma_1, d_1, \dots, \gamma_n)$. A sequence $\pi = (g_n)_{0:N-1} =$

$(g_0, \dots, g_n, \dots, g_{N-1})$ where g_k is a decision rule at time k is called an admissible policy. Let Π_N denote the set of all admissible policies. The controlled trajectory of the POMDP following policy $\pi = (g_n)_{0:N-1} \in \Pi_N$ is defined recursively by $\Xi_0 \in \mathbb{X}$ and for $0 \leq n \leq N-1$,

- $A_n = g_n(\Xi_0, A_0, \dots, \Xi_n)$,
- $\Xi_{n+1} \sim R(\cdot|\Xi_n, A_n)$.

Note that the cemetery state Δ ensures that all trajectories have the same length N, even if they do not have the same number of actual decisions ($d \neq \check{d}$). Then one can define the total expected cost of policy $\pi \in \Pi_N$ starting at $\xi_0 \in \mathbb{X}$ as

$$J_\pi(\xi_0) = \mathbb{E}^\pi_{\xi_0}\left[\sum_{n=0}^{N-1} c(\Xi_n, A_n, \Xi_{n+1}) + C(\Xi_N)\right],$$

and our control problem corresponds to the optimisation problem

$$V(\xi_0) = \inf_{\pi \in \Pi_N} J_\pi(\xi_0).$$

1.2.6 Fully observed Markov decision process

A first problem with value function V is that it can never be computed as the admissible strategies have to be independent from the unobservable states, and they are in general not Markovian. The classical approach is therefore to convert the POMDP into an equivalent fully observed MDP. In Cleynen and de Saporta 2018, we have provided details to derive such equivalence in the simple PDMP framework. More generic computations can also be found in Bäuerle and Rieder 2011; Brandejsky, Saporta, and Dufour 2013. Here, I briefly expose the steps in the controlled PDMP framework.

For $n \leq N$, set $\mathcal{F}^O_n = \sigma(Y_k, Z_k, W_k, 0 \leq k \leq n)$ the σ-field generated by the observations up to n. Let

$$\Theta_n(A) = \mathbb{P}(X_n \in A|\mathcal{F}^O_n) = \mathbb{P}(\Xi_n \in A \times \{(Y_n, Z_n, W_n)\}|\mathcal{F}^O_n),$$

denote the filter or belief process for the unobserved part of the process. The standard prediction-correction approach yields a recursive construction for the filter.

For any $n \geq 0$, one has:

1. Conditionally on $(Y_{n+1}, Z_{n+1}, W_{n+1}) = (y', 0, w')$, $d = (\ell, r) \in L \times \mathbb{T}$ and $\Theta_n = \theta$, one has $\Theta_{n+1}(E_{<\bar{m}}) = 1$ and $\Theta_{n+1} = \Psi(\theta, y', 0, w', d)$ with

$$\Psi(\theta, y', 0, w', d)(A) = \frac{\int_{E_{<\bar{m}}}\int_{E_{<\bar{m}}} f(y' - F(x'))\mathbb{1}_A(x')P(dx'|x, d)\theta(dx)}{\int_{E_{<\bar{m}}}\int_{E_{<\bar{m}}} f(y' - F(x'))P(dx'|x, d)\theta(dx)},$$

for any Borel subset A of $E_{<\bar{m}}$.

2. Conditionally on $(Y_{n+1}, Z_{n+1}, W_{n+1}) = (y', 1, w')$, $d = (\ell, r) \in L \times \mathbb{T}$ and $\Theta_n = \theta$, one has $\Theta_{n+1}(E_{\bar{m}}) = 1$, and $\Theta_{n+1} = \Psi(\theta, 0, 1, w', d)$ with

$$\Psi(\theta, 0, 1, w', d)(A) = \frac{\int_{E_{<\bar{m}}}\big(\int_{E_{\bar{m}}} \mathbb{1}_A(x')P(dx'|x, d)\big)\theta(dx)}{\int_{E_{<\bar{m}}}\big(\int_{E_{\bar{m}}} P(dx'|x, d)\big)\theta(dx)},$$

for any Borel subset A of $E_{\bar{m}}$.

Let $\mathcal{P}(A)$ denote the set of probability measures on set A, The equivalent fully observed MDP is defined as follows.

• The state space is a subset $\mathbb{X}'$ of $(\mathcal{P}(E_M)\times\mathbb{O})\cup\{\Delta\}$.

• The action space is still $\mathbb{A}=(L\times\mathbb{T})\cup\{\check{d}\}$.

• The constraints set $\mathbb{K}'\subset\mathbb{X}'\times\mathbb{A}$ is such that its sections $\mathbb{K}'(\xi)=\{d\in\mathbb{A};(\xi,d)\in\mathbb{K}'\}$ satisfy $\mathbb{K}'(\Delta)=\check{d}$ and $\mathbb{K}'(\theta,y,z,w)=\mathbb{K}'(z,w)=\mathbb{K}(z,w)$ for $(\theta,y,z,w)\in\mathcal{P}(E_M)\times\mathbb{O}$.

• The controlled transition kernels R' are defined as follows: for any bounded measurable function g on $\mathbb{X}'$, any $\xi\in\mathbb{X}'$ and $d\in\mathbb{K}'(\xi)$, one has $R'g(\xi,d)=g(\Delta)$ if $d=\check{d}$, and

$$\begin{aligned}R'g(\xi,d)=\int_{E_M}\int_I\int_{E_M} & g\big(\Psi(\theta,y',0,w+r,d),y',0,w+r\big)\times\\ & f(y'-F(x'))\mathbb{1}_{m\neq\bar{m},m'\neq\bar{m}}P(dx'|x,d)dy'\theta(dx)\\ & +\int_E\int_E g(\Psi(\theta,0,1,w+r,d),0,1,w+r)\mathbb{1}_{m\neq\bar{m}}\mathbb{1}_{m'=\bar{m}}P(dx'|x,d)\theta(dx),\end{aligned}$$

if $\xi=(\theta,y,0,w)$, $d=(\ell,r)$, $x=(m,\zeta,u)$ and $x'=(m',\zeta',u')$.

• The non-negative cost-per-stage function $c':\mathbb{K}'\to\mathbb{R}_+$ and the terminal cost function $C':\mathbb{X}'\to\mathbb{R}_+$ are defined by $C'(\Delta)=c'(\Delta,\check{d},\cdot)=0$ and for $\xi=(\theta,\gamma)\in\mathcal{P}(E_M)\times\mathbb{O}$ and $d\in\mathbb{K}'(\xi)$,

$$\begin{aligned}c'(\xi,d)=c'(\theta,\gamma,d) &= \int_{E_M^2}c(x,d,x')P(dx'|x,d)\theta(dx),\\ C'(\xi)=C'(\theta,\gamma) &= \int_{E_M}C(x)\theta(dx).\end{aligned}$$

• The optimisation horizon is still N.

Denote (Ξ'_n) a trajectory of the fully observed MDP. The set of admissible strategies Π_N of the partially observed MDP remains admissible for the POMDP, and the cost of a strategy $\pi\in\Pi_N$ is

$$J'(\pi,\xi'_0)=\mathbb{E}^{\pi}_{\xi'_0}\left[\sum_{n=0}^{N-1}c'(\Xi'_n,A_n)+C'(\Xi'_N)\right],$$

and the value function of the fully observed problem is,

$$V'(\xi'_0)=\inf_{\pi\in\Pi_N}J'(\pi,\xi'_0).$$

If $\xi'_0=(\delta_{x_0},y_0,z_0,w_0)$, one has $V'(\xi'_0)=V(x_0,y_0,z_0,w_0)$, so that solving the partially observed MDP is equivalent to solving the fully observed one. Of course the policies in Π_N are still in general not Markovian, but one can show that the set of all Markov policies for the filtered Markov Decision Model can be identified as a subset of Π_N. Moreover, one can show that history-dependent policies do not improve the minimal expected cost over Markovian policies (see for instance Karl Hinderer and Hinderer 1970), so that the value function V' satisfies the well known dynamic programming equations, see e.g. Bäuerle and Lange 2018. For $\xi\in\mathbb{X}'$, set $v'_N(\xi)=C'(\xi)$ and for $0\le n\le N-1$, define by backwards induction

$$v'_n(\xi)\ =\ \min_{d\in\mathbb{K}'(\xi)}\left\{c'(\xi,d)+R'v'_{n+1}(\xi,d)\right\}.$$

Let $\xi'_0=(\delta_{(0,\zeta_0,0)},y,0,0)\in\mathbb{X}'$. Then we have $v'_0(\xi'_0)=V'(\xi'_0)=V((0,\zeta_0,0),y,0,0)$.

Now the problem is rigorously stated. Still in the context of partially observed continuous-time PDMPs, the value function can never be computed exactly because of the complexity of the kernels R'. The next three sections present specific classes of (controlled) PDMPs where our contribution was to provide error-controlled approximations of the value function V.

1.3 Detection of relapse, a simple PDMP

In our first paper (Cleynen and de Saporta 2018), we have considered the simple PDMP model presented in the example of Section 1.2.1, in which the observation dates were considered fixed on a grid of size δ. The goal was to detect the jump-time T (corresponding to the relapse time of the patient) and the mode after the jump (relapse type of the patient) based on the noisy marker observations Y_n. We chose to formulate this problem as an optimal stopping problem for a discrete-time Markov chain. However, as in every work that will follow, it is important to note that the underlying process is time-continuous, and in particular that the jump-time T may occur between observation dates. Moreover, the step-size δ corresponding to patient visit dates to the hospital, there is no reason to assume that δ is small, and even less that it may tend to 0.

In this work, since the controller cannot act on the process, we did not need to use observation Z_n corresponding to the death status of the patient. Moreover, to reduce the dimension of the Markov chain X_n, we chose not to include time t_n in its definition. The consequence is that X_n is now an inhomogeneous Markov chain, with corresponding kernel P_n at time n.

For $0 \leq n \leq N$, set $\mathcal{F}_n^{\mathbb{Y}} = \sigma(Y_k, 0 \leq k \leq n)$ the σ-field generated by the observations up to time n and let $\mathcal{T}^{\mathbb{Y}}$ be the set of $\mathcal{F}^{\mathbb{Y}}$-stopping times. A decision taken at the stopping-time $\tau \in \mathcal{T}^{\mathbb{Y}}$ is a $\mathcal{F}_\tau^{\mathbb{Y}}$-measurable random variable A taking values in $\mathcal{M}_+ = \{1, 2, \dots, d\}$ if $\tau \leq N$, equal to 0 if $\tau > N$. Decision $A = i$ corresponds to deciding X_τ is in mode i. Let $\mathcal{A}_\tau^{\mathbb{Y}}$ be the set of admissible decisions at stopping time τ.

We formulate our change-point detection problem as an optimisation problem where we define, until stopping-time τ, the cost-per-stage c and the terminal cost when taking decision a (at stopping time τ) C, with

$$\begin{aligned} c(i,x,y) &= \beta_i \delta, \\ C(m,x,y,0) &= c(m,x,y), \\ C(m,x,y,a>0) &= \alpha \mathbb{1}_{(m=0)} + \gamma_{ma} \mathbb{1}_{(m \neq a; m>0)}, \end{aligned}$$

with $\beta_0 = 0$ and for positive i, $\beta_i = \beta > 0$. Thus, β represents the penalty for late detection, α the false alarm penalty and γ_{ma} the cost of selecting mode a instead of mode m. The cost of an admissible strategy $(\tau, A_\tau) \in \mathcal{T}^{\mathbb{Y}} \times \mathcal{A}_\tau^{\mathbb{Y}}$ for starting point $\xi \in \mathbb{X} \times \mathbb{Y}$ is

$$J(\tau, A_\tau, \xi) = \mathbb{E}_\xi \left[\sum_{n=0}^{(\tau-1)\wedge N} c(X_n, Y_n) + \mathbb{1}_{(\tau \leq N)} C(X_{\tau\wedge N}, Y_{\tau \wedge N}, A_\tau) \right],$$

and the value function of the problem is

$$V(\xi) = \inf_{(\tau, A_\tau) \in \mathcal{T}^{\mathbb{Y}} \times \mathcal{A}_\tau^{\mathbb{Y}}} J(\tau, A_\tau, \xi).$$

The optimal (possibly not achievable) cost is 0 when the jump is detected at the first observation after its occurrence and the right post-jump mode is selected. Unfortunately, optimising this value function is not possible as the PDMP process X_n is not observed. As presented in Section 1.2.6, the classical approach to deal with partial observations is to introduce the filter process and the corresponding completely observed optimal stopping problem for filtered trajectories. In the case of non-controlled PDMP, the filter and its operator have a more simple form, which is recalled below.

For any starting point $\xi = (0, x, y) \in \mathbb{X} \times \mathbb{Y}$, set $\Theta_0 = \theta_0 = \delta_{(0,x)}$ and for $1 \leq n \leq N$, and any Borel subset A of $\mathbb{X}$ set

$$\Theta_n(A) = \mathbb{P}_\xi(X_n \in A | \mathcal{F}_n^{\mathbb{Y}}),$$

the filter for the unobserved part of the process. Setting Set $\mathcal{P}(\mathbb{X})$ the set of probability measures on $\mathbb{X}$, we show that (Θ_n, Y_n) is a Markov chain on $\mathcal{P}(\mathbb{X}) \times \mathbb{Y}$, and that the filter is recursively obtained as follows: for any $n \geq 0$, conditionally on $(\Theta_n = \theta, Y_{n+1} = y')$, one has $\Theta_{n+1} = \Psi_n(\theta, y')$ with

$$\Psi_n(\theta, y')(A) = \frac{\int_{\mathbb{X}} f(y' - F(x'))\mathbb{1}_A(x')P_n(dx'|x,d)\theta(dx)}{\int_{\mathbb{X}} f(y' - F(x'))\mathbb{1}_A(x')P_n(dx'|x,d)\theta(dx)},$$

for any Borel subset A of $\mathbb{X}$.

Let us now define the following generic notation: for g from $\mathbb{X} \times \mathbb{Y}$ onto $\mathbb{R}$ g' is the function from $\mathcal{P}(\mathbb{X}) \times \mathbb{Y}$ onto $\mathbb{R}$ such that $g'(\theta, y) = \int_{\mathbb{X}} g(m, x, y)d\theta(m, x)$. Then the partially observed optimal stopping problem defined above is equivalent to a fully observed optimal stopping problem using the filtered trajectories introduced above. In particular, let

$$J'(\tau, A_\tau, \xi') = \mathbb{E}_{\xi'}\left[\sum_{n=0}^{(\tau-1)\wedge N} c'(\Theta_n, Y_n) + \mathbb{1}_{(\tau \leq N)} C'(\Theta_{\tau\wedge N}, Y_{\tau\wedge N}, A_\tau)\right],$$

Then the value function of the problem is

$$V'(\xi') = \inf_{(\tau, A_\tau) \in \mathcal{T}^{\mathbb{Y}} \times \mathcal{A}_\tau^{\mathbb{Y}}} J'(\tau, A_\tau, \xi'),$$

and can be achieved with the same optimal strategy as that optimising $V(\xi)$. Moreover, the value function is then solution to the dynamic programming equations:
set $v'_N(\theta, y) = \min_{a \in \mathcal{M}} C'(\theta, y, a)$ and for $0 \leq n \leq N-1$, define by backwards induction

$$v'_n(\theta, y) \quad = \quad \min\left\{\min_{a \in \mathcal{M}_+} C'(\theta, y, a); c'(\theta, y) + \mathbb{R}'_n v'_{n+1}(\theta, y)\right\}.$$

Let $\xi'_0 = (\delta_{(0,x)}, y) \in \mathcal{P}(\mathbb{X}) \times \mathbb{Y}$. Then we have

$$v'_0(\xi'_0) = V'(\xi'_0) = V(0, x, y).$$

The main contribution of this first paper was to provide a complete and controlled discretisation strategy to solve the value function and identify a candidate optimal strategy. The main difficulties are first that the filter Θ_n is measure-valued and thus infinite-dimensional and second that this filter cannot be simulated as the Bayes operators Ψ_n involve continuous integration. To build our approximation, we started from the dynamic programming equations and proposed a two-step discretisation of operators R'_n. Our global approach and the relationships between the different Markov chains we introduce, together with their state space and kernels are summarised in Figure 1.5. The left column corresponds to the construction presented above from the original continuous-time PDMP to the fully observed dynamic programming equations.

The first step in the middle column corresponds to a time-dependent discretisation of the state space of the Markov chain (X_n). We obtain a finite state space Markov chain $(\bar{X}_n)$ that we plug into the observation equation $Y_n = F(X_n) + \epsilon_n$ and in the filter operator to obtain Markov chains $(\bar{X}_n, \bar{Y}_n)$ with kernels $\bar{R}_n$ and $(\bar{\Theta}_n, \bar{Y}_n)$ with kernels $\bar{R}'_n$. Finally we replace R'_n by $\bar{R}'_n$ in the dynamic programming equations to obtain the first sequence of approximate value functions. Note that by doing so, $\bar{Y}_n$ does not correspond to a discretisation of the observations Y_n and $\bar{\Theta}_n$ is not the filter of $\bar{X}_n$ given the observations Y_n. By this procedure, we start from a finite state space Markov chain $(\bar{X}_n)$ and obtain a simulatable filter $\bar{\Theta}_n$ that is still measure-valued but can be identified to finite-dimensional vectors. One more approximation is still required to obtain a finite state-space Markov chain.

$$\begin{tikzcd}
\begin{array}{c} \boldsymbol{X}_t = (m_t, x_t, t) \\ \mathbb{X} \times \mathbb{R}^+, \boldsymbol{P} \end{array} \arrow[d] & & \\
\begin{array}{c} X_n = (m_{t_n}, x_{t_n}) \\ \mathbb{X}, P_n \end{array} \arrow[r, "\text{quantisation}"'] \arrow[d, "\text{observations}"', "Y_n = F(X_n) + \varepsilon_n"] & \begin{array}{c} (\bar{m}_{t_n}, \bar{x}_{t_n}) = \bar{X}_n \\ \Omega_n, \bar{P}_n \end{array} \arrow[d, dotted] & \\
\begin{array}{c} (X_n, Y_n) \\ \mathbb{X} \times \mathbb{Y}, R_n \end{array} \arrow[r, dotted] \arrow[d, "\text{filtering}"', "\Psi"] & \begin{array}{c} (\bar{X}_n, \bar{Y}_n) \\ \Omega_n \times \mathbb{Y}, \bar{R}_n \end{array} \arrow[d, dotted] & \\
\begin{array}{c} (\Theta_n, Y_n) \\ \mathcal{P}(\mathbb{X}) \times \mathbb{Y}, R'_n \end{array} \arrow[r, dotted] \arrow[d, "\text{dynamic programming}"'] & \begin{array}{c} (\bar{\Theta}_n, \bar{Y}_n) \\ \mathcal{P}(\Omega_n) \times \mathbb{Y}, \bar{R}'_n \end{array} \arrow[r, "\text{quantisation}"'] \arrow[d, dotted] & \begin{array}{c} (\hat{\Theta}_n, \hat{Y}_n) \\ \Gamma_n, \hat{R}'_n \end{array} \arrow[d, dotted] \\
v'_n(\Theta_n, Y_n) \arrow[r, dotted] & \bar{v}'_n(\bar{\Theta}_n, \bar{Y}_n) \arrow[r, dotted] & \hat{v}'_n(\hat{\Theta}_n, \hat{Y}_n)
\end{tikzcd}$$

Figure 1.5: Two-step approximation of the value functions

The second step in the right column consists in the joint discretisation of the Markov chain $(\bar{\Theta}_n, \bar{Y}_n)$. We obtain a finite state space Markov chain $(\hat{\Theta}_n, \hat{Y}_n)$ with kernel $\hat{R}'_n$. Again, we plug this new kernel into the dynamic programming equations. As the Markov chain $(\bar{\Theta}_n, \bar{Y}_n)$ has a finite state space, integrating with respect to $\hat{R}'_n$ simply corresponds to computing weighted sums. Hence the dynamic programming equations are now fully solvable numerically. This leads both to a numerically tractable approximation of the original value function and to a candidate ϵ-optimal strategy. Both steps are based on discretisation by optimal quantisation of the vector-valued Markov chains.

By defining proper projection operators, distances on the state spaces of the non-observed and filtered processes, and resorting to quantisation grids, we show that if all quantisation grids Ω_n (on the non-observed process) have the same number of points N_Ω and all quantisation grids Γ_n (on the filtered process) also have the same number of points N_Γ, then $\mathbb{E}[|X_n - \bar{X}_n|] = O(N_\Omega^{-1})$ and $\mathbb{E}[|\hat{Y}_n - \bar{Y}_n| + \|\hat{\Theta}_n - \bar{\Theta}_n\|_{n,1}] = O(N_\Gamma^{-1/N_\Omega})$. Thus one has

$$|v'_0(\delta_{(0,x_0)}, y_0) - \hat{v}'_0(\delta_{(0,x_0)}, y_0)| = O(N_\Omega^{-1} + N_\Gamma^{-1/N_\Omega}).$$

This rate of convergence is very slow, which is not surprising given that one had to discretise infinite dimension measure-valued random variables. This is the well known curse of dimensionality one is faced with when dealing with partial observations.

We can finally construct a computable stopping strategy using the fully discretised value function. Suppose that the process starts from point $\xi_0 = (0, x_0, y_0)$ and observations $y_0, \ldots, y_n$ are available at time n. One cannot compute the filter $\mathbb{P}_{\xi_0}(X_n \in \cdot|(Y_0, \ldots, Y_n) = (y_0, \ldots, y_n))$ because of the continuous integrals in the iterative computation of the filter . However, one can recursively compute an approximate filter as follows.

$$\bar{\theta}_0 = \delta_{(0,x_0)}, \quad \bar{\theta}_k = \bar{\Psi}_{k-1}(\bar{\theta}_{k-1}, y_k),\ 1 \leq k \leq n.$$

By construction, $\bar{\theta}_k$ belongs to $\mathcal{P}(\Omega_k)$ for all k. Then this approximate filter can be projected onto the quantisation grids $(\Gamma_k)_{0\leq k\leq n}$:

$$(\hat{\theta}_k, \hat{y}_k) = p_{\Gamma_k}(\bar{\theta}_k, y_k),$$

for all $1 \leq k \leq n$. Finally, the values of $\hat{v}'_k(\hat{\theta}_k, \hat{y}_k)$ are available for all $0 \leq k \leq n$.

Now we define two sequences of function $(r_n)_{0\leq n\leq N}$ and $(a_n)_{0\leq n\leq N}$ as
• for $0 \leq n \leq N-1$, $r_n : \mathcal{P}(\Omega_n) \times \mathbb{Y} \to \{0,1\}$ and $a_n : \mathcal{P}(\Omega_n) \times \mathbb{Y} \to \mathcal{M}_+$ are such that

$$\begin{aligned} r_n(\bar{\theta}, \bar{y}) &= \mathbb{1}_{\min\limits_{a\in\mathcal{M}_+} C'(p_{\Gamma_n}(\bar{\theta},\bar{y}),a) < c'(p_{\Gamma_n}(\bar{\theta},\bar{y})) + \hat{R}'_n \hat{v}_{n+1}(p_{\Gamma_n}(\bar{\theta},\bar{y}))}, \\ a_n(\bar{\theta}, \bar{y}) &= \arg\min_{a\in\mathcal{M}_+} C'(p_{\Gamma_n}(\bar{\theta}, \bar{y}), a)\mathbb{1}_{(r_n(\bar{\theta},\bar{y})>0)}. \end{aligned}$$

• $r_N : \mathcal{P}(\Omega_N) \times \mathbb{Y} \to \{0,1\}$ and $a_N : \mathcal{P}(\Omega_N) \times \mathbb{Y} \to \mathcal{M}$ are such that

$$\begin{aligned} r_N(\bar{\theta}, \bar{y}) &= \mathbb{1}_{(\arg\min_{a\in\mathcal{M}} C'(p_{\Gamma_N}(\bar{\theta},\bar{y}),a)>0)}, \\ a_N(\bar{\theta}, \bar{y}) &= \arg\min_{a\in\mathcal{M}} C'(p_{\Gamma_N}(\bar{\theta}, \bar{y}), a). \end{aligned}$$

Thus r_n is a stopping indicator depending on which term won the minimisation in the dynamic programming, and a_n corresponds to the mode to be selected after the jump. Our candidate stopping strategy is then illustrated in Figure 1.6:

We have shown the performance of our approach on several generic PDMP examples, comparing our results to standard techniques such as Kalman filtering and moving average and shown that as soon as the flows are non-exponential, it significantly outperforms the other approaches. In particular, we show a fun example where the dynamic change in the different modes is in the frequency of an oscillating process, and where even given a full trajectory it is almost impossible to distinguish the modes by eye. Our strategy performs remarkably well even in this setting, taking advantage of the knowledge of the model in the different modes.

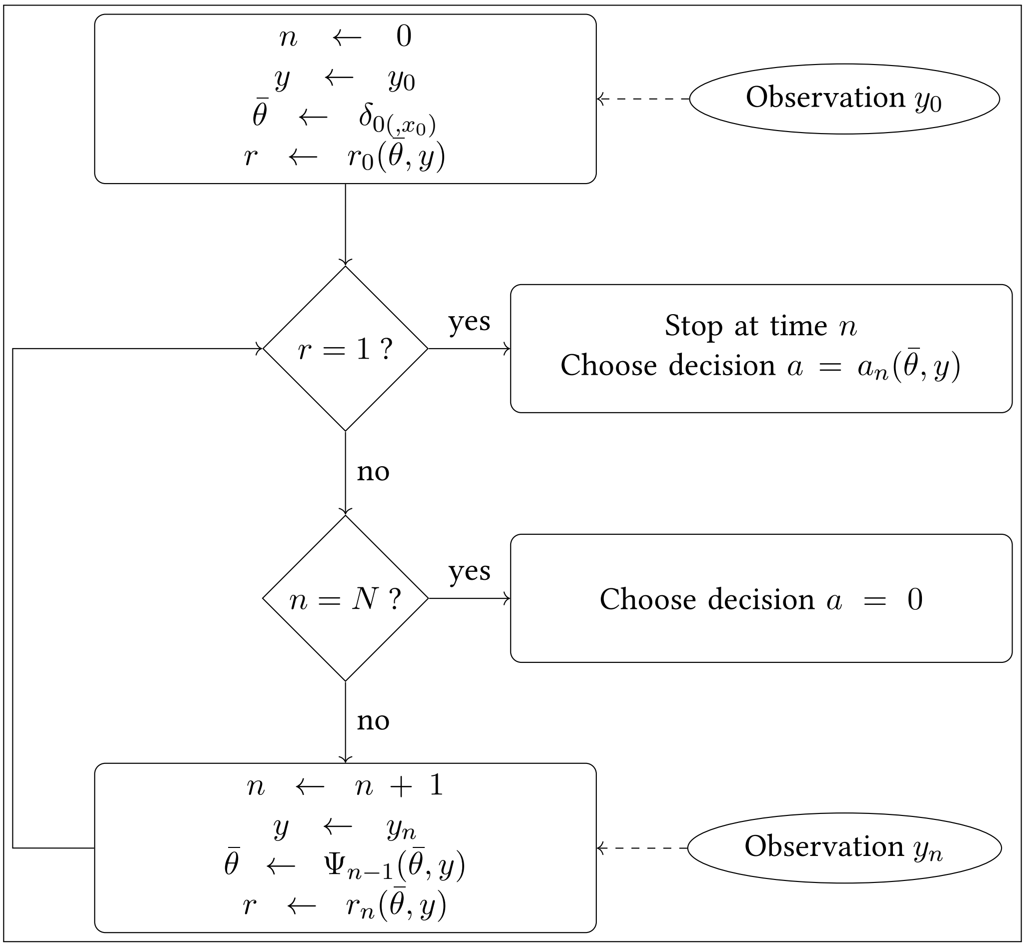
$n \leftarrow 0$
$y \leftarrow y_0$
$\bar{\theta} \leftarrow \delta_{0(,x_0)}$
$r \leftarrow r_0(\bar{\theta}, y)$
Observation y_0
$r = 1$?
yes
Stop at time n
Choose decision $a = a_n(\bar{\theta}, y)$
no
$n = N$?
yes
Choose decision $a = 0$
no
$n \leftarrow n + 1$
$y \leftarrow y_n$
$\bar{\theta} \leftarrow \Psi_{n-1}(\bar{\theta}, y)$
$r \leftarrow r_n(\bar{\theta}, y)$
Observation y_n

Figure 1.6: Computable stopping strategy

1.4 Visit dates, treatment allocation, a controlled PDMP

In the two works presented in this section (see Cleynen and de Saporta 2023 and de Saporta, d'Argenlieu, et al. 2024), we considered the controlled PDMP example presented in Section 1.2.2 in the medical framework of following cancer patients. For the practitioner, controlling the disease is equivalent to choosing the best available treatment as well as the best next visit date in order to minimise its impact on the patient's quality of life along time. Defining the impact of treatment on the quality of life is a difficult task as it will typically depend on the treatment's side effects, the number of visits, the burden of living with a disease and the remaining life expectancy. We proposed a mathematical definition in terms of a cost function that takes into account those different aspects. For a decision $d = (\ell, r)$ comprising a treatment allocation ℓ and a time to next visit r, and for a current marker level ζ at time t_k, and future marker level ζ' at time $t_{k+1} = t_k + r$, we define

$$c(\zeta, d, \zeta') = C_V + \kappa|\zeta' - \zeta_0|r + \beta r \mathbb{1}_{\{\zeta=\zeta_0, \ell\neq\emptyset\}} + M\mathbb{1}_{\{\zeta'=D\}},$$

where C_V is a visit cost, κ is non-negative scale factor penalising high marker values, β is a penalty for applying an unnecessary treatment and M is the death cost.

This cost function thus takes into account a visit cost, to prevent patients from undergoing too many screening tests, a cost depending on the marker value at the next visit, to encourage treatment and calibrate visit dates, a cost for degradation of quality of life due to treatments, in particular if they are not appropriate, and a cost for dying.

Calibrating cost parameters C_V, κ, β, and M is a very difficult task, which is allowed to be patient-dependent (some patients may even express a wish to be sedated rather than undergo very long and painful treatments), and treatment strategies are bound to be parameters-dependent.

When cast as a controlled PDMP with this cost function, the practitioner's problem is mathematically equivalent to solving a POMDP which expected value optimisation can be stated as

$$V = \inf_{\pi\in\Pi} \mathbb{E}^{\pi}_{X_0}\left[\sum_{n=0}^{N_\pi-1} c(X_n, d_n, X_{n+1})\right],$$

where V is called the *optimal policy value* and represents the lowest possible expected total cost, Π is the set of admissible policies (yielding decisions depending only on current and past observations), N_π is the patient-specific number of visits within the time-horizon of the study when using policy π, $d_n = \pi(Y_0, t_0, \dots, Y_n, t_n)$ is the decision (ℓ, r) taken at the n-th visit date t_n according to policy π, and Y represents the marker observation process for the controlled PDMP/POMDP. Solving this problem amounts to computing (a good approximation of) the optimal policy value and identifying an admissible policy π^* that reaches (a value close to) the minimum.

The two papers described in this section propose fundamentally different approaches to approximate the value function. The first, presented in Cleynen and de Saporta 2023, is in the continuation of the first paper with Benoîte de Saporta where we propose a discrete approximation of the PDMP process and its filter, and solve exactly the value function of this approximate model. The second, presented in de Saporta, d'Argenlieu, et al. 2024, is a work in collaboration with Aymar Thierry d'Argenlieu, masters thesis student in 2022, Régis Sabbadin (DR INRAE in Toulouse) and Benoîte de Saporta, where we propose an approximate resolution by simulation of the exact model. The following sections briefly present the main results of both works.

1.4.1 Let's discretise again

The resolution strategy of Cleynen and de Saporta 2023 is identical to that of Cleynen and de Saporta 2018: as in Figure 1.5, we start by discretising the controlled PDMP X_n in a finite state-space Markov chain $\bar{X}_n$ with Markov kernel $\bar{P}$, and then discretise the fully observed process $(\bar{\Theta}_n, O_n)$ into a finite state-space Markov chain Ξ_n with kernel $\hat{R}_n$.

The first difficulty is that the number of possible strategies is exponential in the horizon time (here at each visit date the practitioner has a choice of 3 treatments and 3 next visit dates, *i.e.* 9 possibilities), hence discretisation by quantisation is no longer possible. The second difficulty comes from the boundaries of the state-space E at ζ_0 and D: the transition kernels of our PDMP are not regular on the whole state space, but only locally regular in some sub-areas. This implies that approximation errors will not be controlled in any given discretisation of E, even if the grid sizes increase.

One of our main contribution is therefore to propose a strategy to build the initial discretisation grid so that the state-space is best explored with a minimal number of points, and the regularity of the kernels are preserved over each cell. To do so, we consider the sets $B_m = \{\zeta \in [0, D], \exists d = (\ell, r) \in \mathbb{A} - \{\check{d}\}, t_m^{*\ell}(\zeta) = r\} \cup \{\zeta_0, D\}$ for $m = 1$ or 2 and $b_m^{(1)} < b_m^{(2)} < \cdots < b_m^{(T)}$ its $T = 3|\mathbb{T}| + 2$ ordered elements. We then consider $(F_j)_{1 \leq j \leq 2T}$ the following partition of E_M:

$$\begin{aligned}
&F_1 = E_0, \quad F_{2T} = E_3\\
&F_j = \{1\} \times (b_1^{(j-1)}, b_1^{(j)}) \times [0, H], \text{ for } 2 \leq j \leq T,\\
&F_j = \{2\} \times (b_2^{(j-T)}, b_2^{(j-T+1)}) \times [0, H], \text{ for } T+1 \leq j \leq 2T-1.
\end{aligned}$$

The splitting points satisfy $t_m^{*\ell}(\zeta) = r$ for $m \in \{1, 2\}$ and $d = (\ell, r) \in \mathbb{A} - \{\check{d}\}$. They separate values of ζ for which the probability of reaching the top value until the next epoch is strictly positive from those with null probability. Let $\Omega = \{\omega^1, \ldots, \omega^{n_\Omega}\}$ be a finite grid on E_M containing at least one point in each mode $m \in M$. Let p_Ω denote the nearest-neighbour projection from E_M onto Ω for the distance defined in 1.2.5 with $\|(\zeta, u)\| = |\zeta| + |u|$. In particular, p_Ω preserves the mode. Let $(C_i)_{1 \leq i \leq K}$ be a Voronoi tessellation of E_M associated to Ω. Namely, $(C_i)_{1 \leq i \leq n_\Omega}$ is a partition of E_M such that for all $1 \leq i \leq n_\Omega$, one has

$$C_i \subset \{x \in E_M; \|x - \omega^i\| \leq \|x - \omega^j\| \; \forall j \in \Omega\}.$$

If for all $1 \leq k \leq n_\Omega$, there exists some $1 \leq j \leq 2T$ such that $C_k \subset F_j$ (in practice, this assumption means that the points closest to the hyperplanes in the grid are symmetric with respect to these hyperplanes) and we define the controlled kernels $\bar{P}$ from $E_M \times (L \times \mathbb{T})$ onto Ω as

$$\bar{P}(\omega^j | x, d) = P(C_j | x, d),$$

then $\bar{P}$ is Lipshitz on all cells C_i.

By replacing P by $\bar{P}$ in the definition of all kernels and dynamic programming equations, we show that functions $\bar{v}'_n$ are good approximations of our value functions v'_n:

$$\begin{aligned}
|v'_N(\xi') - \bar{v}'_N(\xi')| &= 0\\
|v'_n(\xi') - \bar{v}'_n(\xi')| &\leq C_{v'_n} \sup_{j \in \{1, \ldots, \ell\}} \mathcal{D}_j, \quad 0 \leq n < N,
\end{aligned}$$

where $C_{v'_n}$ depends only on n, N, δ and the regularity constants of the parameters.

The second discretisation of the filter and observations presents less technical difficulties than the first. In practice, the main challenge is to discretise probability distributions over a space of large dimension (the dimension of the first grid) with sufficiently few points that the approximate kernels may still be numerically computable by simulation. In theory, if $\Gamma = \{\rho^1, \ldots, \rho^{n_\Gamma}\}$ defines a finite grid on $\bar{\mathbb{X}}'$ containing at least Δ, one element $\xi = (\theta, \gamma)$ such that $\theta(\{x = (m, \zeta, u); m \neq 3\}) = 1$ and one element $\xi' = (\theta', \gamma')$ such that $\theta'(\{x = (m, \zeta, u); m = 3\}) = 1$;

and if $(\bar{C}_i)_{1\leq i\leq n_\Gamma}$ is a Voronoi tessellation of $\bar{\mathbb{X}}'$ associated to the grid Γ, we define the controlled kernels $\hat{R}'$ on $\bar{\mathbb{X}}'$ as follows: for any $\rho^j \in \Gamma$, any $\xi \in \bar{\mathbb{X}}'$ and $d \in \mathbb{K}'(\xi)$, one has

$$\hat{R}'(\rho^j|\xi, d) = \bar{R}'(\bar{C}_j|\xi, d).$$

Then functions $\hat{v}'_n$ are good approximations of functions $\bar{v}'_n$ and thus of our value functions v'_n:

$$|\hat{v}'_N(\rho^j) - \bar{v}'_N(\rho^j)| = 0,$$
$$|\hat{v}'_n(\rho^j) - \bar{v}'_n(\rho^j)| \leq C_{\bar{v}'_n} \sup_{j\in\{1,\dots,n_\Gamma\}} \bar{\mathcal{D}}_j,$$

where $C_{\bar{v}'_n}$ depends only on n, N, δ and the regularity constants of the parameters.

To limit the number of points in the grid, we proposed an iterative construction strategy with a fixed number of points which consists in computing the kernels for an initial grids, obtaining the optimal strategy in this context, simulating trajectories with this optimal strategy, and removing points that are least visited while adding points with maximal distance to the current set of points in the grid. This construction may be stopped when the approximation of the value function no longer improves, or when the maximal distances become smaller than a given threshold.

Here again, we showed on simulation studies that our strategy performs better than other approaches, such as strategies based on estimating the mode only, or naive strategies purely based on marker values. We also show that allowing the practitioner to choose the next visit date (instead of using a fixed grid) yields better results (*i.e.* better quality of life for the patients), with a decreased number of visits compared to the frequent grid approach, *i.e.* with fixed visits every 15 days, and an earlier relapse detection compared to the sparse grid approach, with fixed visits every 60 days.

1.4.2 Another strategy: resolution by simulation

In this work, we considered a completely different approach to compute the optimal strategy: instead of exploring the whole decision tree by backward induction as in dynamic programming, we explore the tree in a forward direction by simulation, pruning the branches of highest cost and stopping the exploration before the terminal leaves. This strategy relies on the Partially Observed Monte-Carlo Planning (POMCP) algorithm by Silver and Veness 2010.

POMCP is an efficient simulation-based algorithm that has been designed for real-time planning in large finite state-space POMDPs. In this paper we show that even though it has not been designed to handle continuous state and observation spaces, we can adapt it to solve controlled PDMPs, thanks to their efficient simulation property, without resorting to the computation of complex integrals for computing transition probabilities. The objective of POMCP is to reduce the complexity of dynamic programming, which requires the construction of the entire decision tree (including the probabilities of every possible outcome with every possible decision at every future time-point), by sampling the tree in a principled way so as to compute the current optimal action. POMCP is thus an *online* algorithm, which re-estimates the optimal strategy at each new data acquisition. It relies on two main properties.

- the ability to simulate trajectories, so as to progressively build the decision tree and update filters Θ at every intermediate node h of the tree. Recall that a filter is a probability distribution representing the (approximate) distribution of the current hidden state given the observations. The standard POMCP algorithm uses a specific family of simulation-based filters Θ^p called *particle filters* specified below. Filters are used to sample sets of plausible states.

- the requirement to provide *estimates* of the expected value of the policy in leaves of the current exploration tree, in order to guide exploration and build the decision policy.

One of our contribution was to show that POMCP is particularly well suited for controlled PDMPs. For instance trajectory simulation and policy evaluation are particularly efficient in POMCP, in the case of a controlled PDMP.

1. Simulation is particularly straightforward with PDMPs (see Algorithm 3, and de Saporta, Dufour, and Zhang 2015; Gillespie 1977; Lemaire, Thieullen, and N. Thomas 2020), requiring only to simulate the jump times and exploit the deterministic behavior between jumps. In our medical framework, it is made even more simple since only few jumps are allowed. When little knowledge is available about the underlying process, a classic approach is to resort to *particle filters* Θ^p (see for instance Del Moral 1996). A particle filter Θ^p at step n is a discrete uniform probability distribution with finite support B^p (where B^p may have repeated atoms). It is updated at step $n+1$ though simulations: states s from B^p are updated through a one-step simulation to a new state s', and selected to be added to B^p if the simulated observation is close to the true one. As an alternative filter to compare to, we propose to use a *conditional filter* Θ^c derived from the exact filter (that is the conditional distribution of the hidden state given the observations) from Cleynen and de Saporta 2023. The exact conditional filter is updated through a recurrence formula involving ratios of integrals over the state space. By discretising the state space, one can construct the approximation Θ^c of the exact filter. Unlike the particle filter that has a dynamically changing support with a uniform mass function, this conditional filter has a fixed support (the discretised space) with changing mass functions that are updated through analytical ratios of weighted sums.

2. To estimate the future expected cost at some node of the tree, POMCP requires to simulate many full trajectories from the current node to a leaf of the tree. This requires to apply an arbitrary strategy to pick actions at every future nodes for which a decision has not yet been optimised. This arbitrary strategy is called a *rollout* in the POMCP framework. The most naive rollout strategy consists in uniformly randomly selecting decisions from the decision set $\{\emptyset, a, b\} \times \{15, 30, 60\}$. We consider instead a mode-based rollout strategy, which consists in choosing action $\emptyset$ in mode 0 (no treatment if the simulated patient is in remission), action a in mode 1 and b in mode 2 (most efficient treatment if the simulated patient has relapsed) and a fixed next visit date of 15 days. This rollout strategy, while not being necessarily optimal (depending on the cost function it might be optimal not to treat at the beginning of a relapse, or to treat preventively when in remission), exploits knowledge of the cost function, hence yields better estimates of action costs at time t. Note also that this mode-based policy is not applicable for real patients, since their mode is not observed. It is only applicable to simulated patients. This is fine since POMCP's rollout strategy is only used through simulations to estimate costs.

The previous paragraphs set the grounds for optimising the long-term follow up of patients. In practice, we will assume a patient will enter the follow-up study once they enter the remission phase after an initial round of treatment. The practitioner may hence assume that their current state is known, *i.e.* $s_0 = (0, \zeta_0, 0)$ and the initial value of both the particle and conditional filters is the Dirac mass at s_0. The initial observation is $\omega_0 = (\zeta_0, t_0)$. The adapted POMCP algorithm is run to obtain the optimal decision d_0, which the practitioner can use (if she decides to) to allocate treatment and decide on the next visit date t_1.
At visit n, the patient will come back for some new marker measurement, so that the n-th observation value $\omega_n = (y_n, t_n)$ is obtained. The practitioner will have access to her full history, $h_n = \langle \omega_0 d_0 \omega_1 d_1 \cdots d_{n-1} \omega_n \rangle$ as well as her last belief filter, Θ^c_{n-1} or Θ^p_{n-1}. An initial update of the filter is performed, either using the recursion formula for Θ^c_n from Θ^c_{n-1} and ω_n, or by particle filtering through rejection sampling for Θ^p_n from Θ^p_{n-1} and ω_n. The adapted POMCP algorithm is then ran to obtain the optimal current decision d_n, which the practitioner can use (or

not) to allocate treatment and decide on the next visit date t_{n+1}. The global follow-up strategy is illustrated in Figure 1.7.

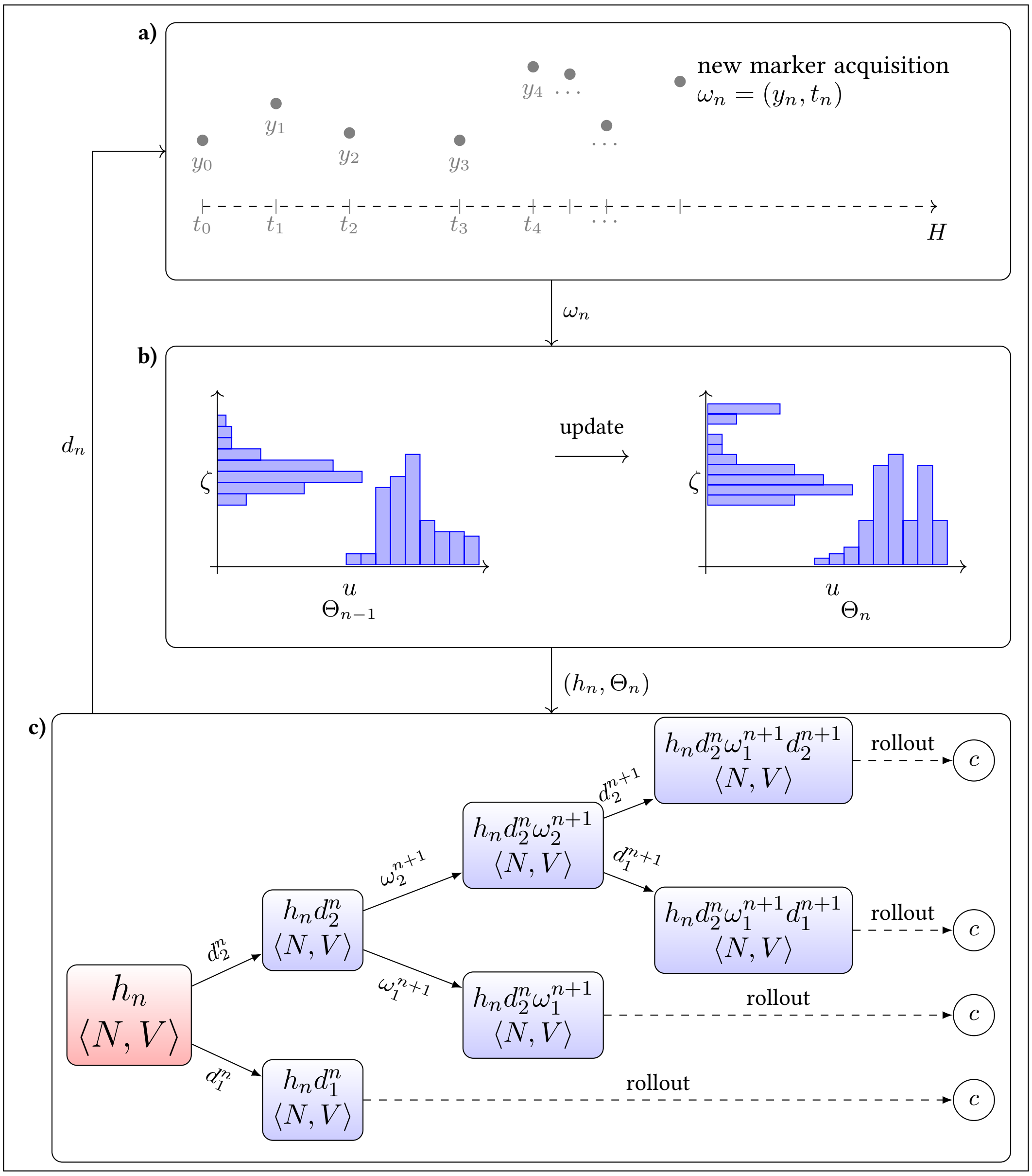
a)
new marker acquisition
$\omega_n = (y_n, t_n)$
y_0 y_1 y_2 y_3 y_4 . . .
t_0 t_1 t_2 t_3 t_4 . . .
H
ω_n
d_n
b)
update
ζ
u
Θ_{n-1}
Θ_n
(h_n, Θ_n)
c)
h_n $\langle N, V\rangle$
d_2^n
d_1^n
$h_n d_2^n$ $\langle N, V\rangle$
$h_n d_1^n$ $\langle N, V\rangle$
ω_2^{n+1}
ω_1^{n+1}
$h_n d_2^n \omega_2^{n+1}$ $\langle N, V\rangle$
$h_n d_2^n \omega_1^{n+1}$ $\langle N, V\rangle$
d_2^{n+1}
d_1^{n+1}
$h_n d_2^n \omega_1^{n+1} d_2^{n+1}$ $\langle N, V\rangle$
$h_n d_2^n \omega_1^{n+1} d_1^{n+1}$ $\langle N, V\rangle$
rollout
c

Figure 1.7: **Practice of patient follow-up. a)** At each new visit the patient has a new marker measurement, and the practitioner receives a new observation $\omega_n = (y_n, t_n)$. **b)** The filter is updated with the new observation, either through particle rejection sampling (*particle filter*) or via a recursion formula (*conditional filter*). **c)** The decision tree is partially explored via simulation through an adapted POMCP algorithm using the updated filter. The algorithm returns the optimal decision d_n, combination of a time to next visit (defining t_{n+1}) and treatment to allocate (influencing y_{n+1}).

1.4.3 Comparison of the two approaches

We have performed a simulation study based on the data presented in Section 1.1. We calibrated the parameters of our PDMP model (see for instance de Saporta, d'Argenlieu, et al. 2024), and we performed simulations to evaluate the performance of the discretisation/DP and POMCP approaches to select the combination of treatment and next visit date at each time point of the trajectories (these time-points being themselves selected by the algorithms). Here I just show an example of study where we compare the results of three resolution strategies (the discretisation/DP approach (DP) of Section 1.4 that relies on exact resolution by dynamic programming of the discretised POMPD, the adapted POMCP with the conditional filter (POMCP-Conditional) and the adapted POMCP with the particle filter (POMCP-Particles) of Section 1.4.2) each calibrated with their optimal parameters. The comparison is performed on biological relevant outcomes: the death rate, the Progression-Free Survival (PFS) time, that is the time from entry in the study to the first relapse, the time spent under treatment, the number of visits to the hospital, and the cost (measured by a Monte-Carlo estimate of the expectation). Those quantities were normalised so that they range between 0 and 1, and such that an optimal result is 0. To do so, death rate was normalised so that a random treatment strategy yields 1 (in our setting, a random strategy leads to the death of 5% of patients); the PFS was transformed as 1-(PFS/H) (where we recall that H is the study horizon), so that a patient who does not relapse has normalised PFS equal to 0; the time spent under treatment was normalised by H, the number of visits was normalised as $\frac{N_{\text{Visit}}-40}{160-40}$ since over the horizon, a visit every 15 days produces 160 visits, whereas a visit every 60 days produces 40 visits; and finally the cost was normalised as $\frac{C-v_0}{C_{\text{random}}-v_0}$ where v_0 is the the best approximation of the optimal value obtained through discretisations in Cleynen and de Saporta 2023, and C_{random} is the average cost of the random strategy.

We simulated 500 trajectories with each strategy under the same cost parameters. The results are summarised in the Radar plot of Figure 1.8, and additional visual information on average trajectory cost are given in the barplots. In the Radar plot representation, a perfect strategy should delimit the inner circle. Interestingly, the combination of POMCP with the conditional filter yields the lowest average trajectory cost and the shortest average time spent under treatment, by slightly increasing the number of visits and reducing PFS compared to the DP approach. However, the particle-based POMCP approach (relying fully on simulations with no other exploitation of the underlying model) yields cost almost as good as the previous two approaches, with increased number of visits but longest progression-free survival time. Importantly, out of 500 simulations, one of the trajectories ended with a patient dying.

The main advantage of the discretisation/DP approach is that solutions are pre-computed for all new patients. This is especially useful under the assumption that all patients have the same dynamics with the same parameters. Its main drawback is that the model presented here is at maximum complexity for such an approach. In particular, it will become intractable if one wants to take into account more disease markers or more modes and treatments.

Conversely, the simulation-based approach can extend to any complexity provided simulations can be performed easily and fast. In addition, one of the main advantages of simulation-based approaches such as that presented here is that cost parameters can be modified with each new patient, yielding a patient-based procedure closer to precision medicine. This remains quite theoretical, as in practice calibrating cost parameters is a very difficult task, but with experience practitioners may be able to encode personal preferences, such as shorter life with better quality, or longer life at the price of more treatments, etc.

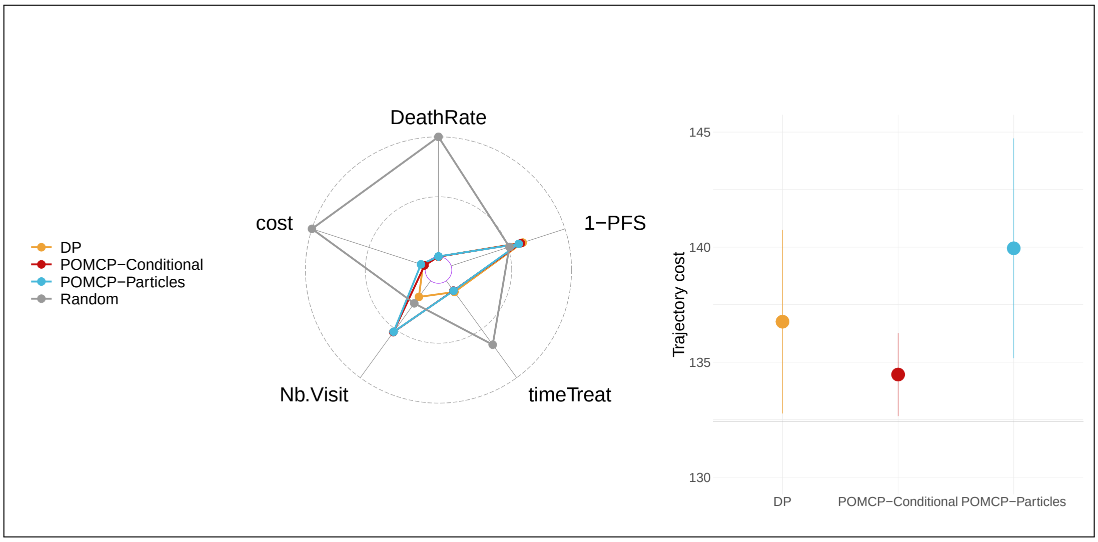

Figure 1.8: **a)** Radar plot comparing performances of 4 solution strategies on death rate, progression-free survival (PFS), time spent under treatment, average number of visits per patient, and average trajectory cost. An optimal strategy would be the inner-circle. **b)** Barplot of trajectory cost for 500 simulations under three main strategies: POMCP with particle filter, POMCP with conditional filter and Dynamic Programming on discretised processes.

1.5 Future research plans

In all the works presented above, we made the assumption that the patient-disease model was known, which is a daring assumption. A fundamental next step of this general follow-up framework is to extend it by considering an unknown model where (at least) some parameters of the model have to be learned while controlling the trajectory, for instance by applying *Reinforcement Learning* methods (Sutton and Barto 2018).

A first step in this direction is to better estimate the survival distributions, here meaning the jump-point distributions, *i.e.* the risk functions λ. This work is one of the main focus of the thesis of Amélie Vernay (co-supervised with Benoîte de Saporta, PhD started in Oct 2023). In her setting, she assumes that the PDMP model is known up to the sojourn times which must be estimated. The goal is to understand if exploiting the (partial) knowledge of this model may provide better estimates of the risk functions compared to model-unaware methods such as Kalman filtering or moving averages plugged to standard survival estimators. In particular, those methods are bound to infer change-points on observation times, regardless of their possible occurrence at any continuous time.

One of the approaches Amélie is currently investigating is the extension of estimators for type 2 interval censored data to its probabilistic counterpart. Indeed, at each new data acquisition, our algorithm provides, among other quantities, probabilities that a jump occurred in between each pair of previous observations. When we condition on the event that a jump actually occurred, we are exactly in the type 2 interval censored data framework and we may apply the non-parametric estimators from the literature (see for instance Bouaziz, Brunel, and Comte 2019). The final estimator may therefore be obtained by integrating over the probability distribution of jumps in an online manner. Among the questions we will investigate, we may cite

- How does the observation noise propagate to change-point estimators?
- Does type 2 censoring estimator provide better results than parametric estimators assuming change-points occur on observation dates?
- Can we provide a (non-parametric) type 2 estimator that is numerically computable?
- Does this estimator have a simulatable distribution?
- Does learning the survival distribution allow to better optimise the treatment strategies?

A second step in the learning while controlling framework is to estimate the parameters of the disease flow (in our case the slope of the exponential disease growth) as we treat the patient. This is the main focus of the thesis of Orlane Rossini (co-supervised with Benoîte de Saporta and Régis Sabbadin, PhD started in Oct 2022). A fundamental problem in Reinforcement Learning is the difficulty of deciding whether to select actions in order to learn a better model of the environment, or to exploit current knowledge about the rewards and effects of actions (Katt, Oliehoek, and Amato 2017). Part of this decision may be encoded in the cost function. For instance, if treatment may only be applied when the marker reaches a minimal level, it will be fundamental to estimate the growth of the marker in order to better decide on the next visit date when to apply the treatment. On the other hand, if the cost function is blind to this parameter, the optimal strategy might never allow to learn the model.

Orlane's PDMP model is very similar to the model described in the previous section, with the following changes: the sojourn time (risk function λ) and efficiency of the treatment (flow parameters in the disease state under treatment) now depend on the number of previous relapses, allowing to model both a tendency to relapse more and more often, and an acquired resistance to treatment. Moreover, instead of having two different diseases types (modes 1 and 2 previously), each patient now has a unique disease state with a patient-specific characteristic (*i.e.* parameter v) that we aim at estimating to better treat. We therefore consider a Bayesian framework with an (unknown) population distribution on parameter v for which we have a non-informative prior. Among the questions we have / will investigate, we may cite

- What are the reinforcement learning algorithms that are compatible (or best adapted) to PDMPs (which are by essence in continuous time, continuous space)?
- How can we update the posterior on v at each new data acquisition of a specific patient to better learn the patient-specific realisation of v without biasing the population posterior v? How then to update the population posterior?
- Can deep-learning methods do better than model-based methods?

Orlane has already started addressing many of those questions, starting from a review of model-free RL approaches (for instance the Q-learning algorithm, see S. E. Li 2023), versus model-based Bayesian RL approaches (for instance BAMDP, see Martin 1967) for which she has assessed the performances and evaluated the limitations through a toy-model study. She is now proposing to use a *Deep* version of an adapted BAMDP algorithm for partially observed models. This approach will allow to exploit both the performances of Deep-learning methods, very greedy in training data, and the knowledge of the model through extensive simulations that can be branched to a deep neural-network learning the optimal strategy. The partial knowledge of the model (through the unknown parameters) is removed through the use of a Bayesian paradigm, where a conjugate prior distribution allows to transform the model into a fully known model-based setting. The posterior distribution is updated at each new observation of a patient. This novel framework allows to overcome the limitations of previous algorithms which were conditioned to work on finite state-spaces.

Finally, a more long-term research project I would like to investigate is how to combine information of other patient measurements (typically targeted sequencing at diagnosis, or more simple covariates such as gender, age, translocations, etc) to better estimate the model and, therefore, provide a better treatment strategy? And maybe even more difficult: MM patients are now being followed both on the heavy chain and light chain of the immunoglobulin gene, *i.e.* two dependent marker variables are acquired at each new collection time. While this is in theory straightforward to model through a PDMP, it dramatically increases the dimension of the state-space, hence preventing any discretisation-based resolution strategy. Can simulation-based algorithm deal with such complexity?

Chapter 2

Statistical tools for establishing Direct RNA Sequencing feasibility in patient samples

This chapter is essentially a collection of works initiated during my visit at the *John Curtin School of Medical Research* of the Australian National University in Canberra. In the first work I contributed most of the methodological aspects, combining methodologies from the segmentation and the survival estimation fields to propose an estimation of RNA degradation in sequencing experiments. In the second work the methodological contributions come from a strong collaboration with two students, Stefan Prodic and Akanksha Srivastava. We propose to use neural networks to predict the modification status of nucleotides at the transcript and site levels. I end this chapter with the brief description of an ongoing project on Multiple Myeloma where both those tools are exploited to describe the modification landscape in subgroups of MM patients.

2.1 Introduction

To anchor this chapter in the global picture of the cell machinery, I'll start with a brief reminder of the central dogma of molecular biology, illustrated in Figure 2.1.

The genetic information we inherit from our parents is contained in our DNA, the famous double-stranded helix of complementary sequences of nucleotides A, C, T and G. This DNA is a succession of *coding regions*, that we call genes, and of *non-coding regions* which have been discovered to play critical roles in the development of a cell but that are off-topic to us. Genes are *transcribed* into RNA, a single-stranded and oriented, from *5' end* to *3' end*, sequence of nucleotides (A, C, U and G) that are complementary to the gene's sequence: an A for a T, a C for a G, a U for an A and a G for a C. There exist 8 types of RNA, classified by function and by length, but we will focus only on two:

- messenger RNAs (mRNA) carry the genetic information necessary to make proteins. They undergo the second part of the Central Dogma (right side of Figure 2.1): translation. mRNAs, produced in the nucleus of the cell, undergo a long preparation phase before exiting to the cytoplasm for translation. Two aspects of this maturation phase (see Figure 2.2) are of particular interest to us. The first is the *splicing* of introns and exons, sub-compartments of the gene that the mRNA may retain or not for translation into different proteins. We call RNA variants of a gene *isoforms*. The second is the addition of a *poly-A tail* (a long sequence of A nucleotides) at the downstream extremity of the mRNA (also called 3' end). Once mRNAs are mature, they are ready to undergo translation, precisely with the help of ribosomal RNAs.

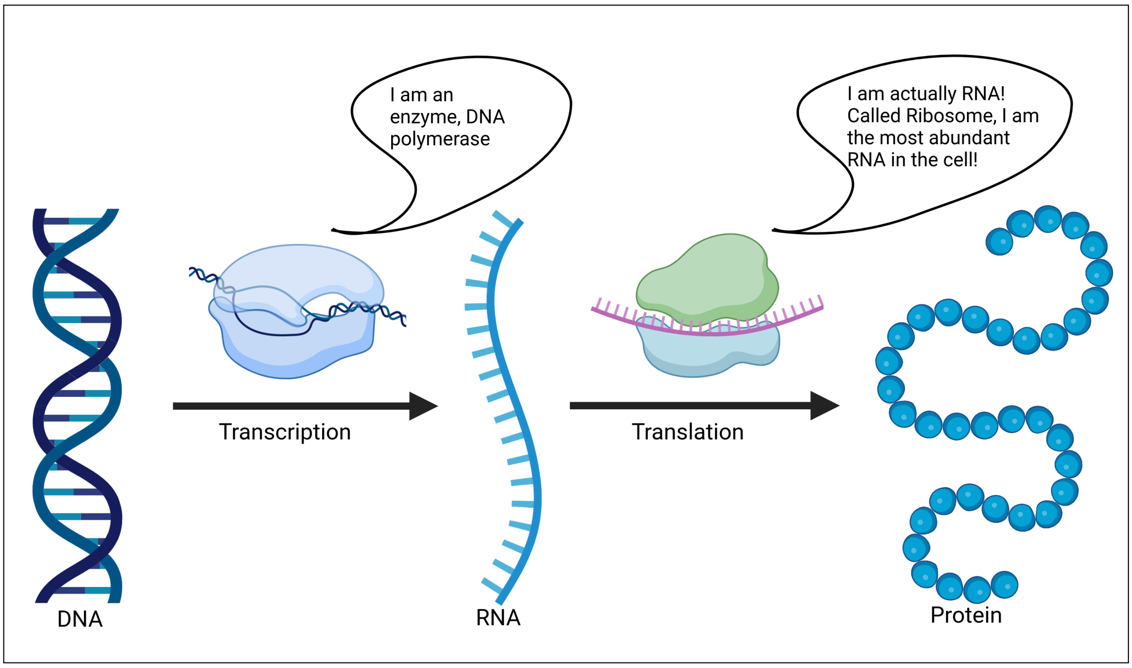
I am an enzyme, DNA polymerase
I am actually RNA! Called Ribosome, I am the most abundant RNA in the cell!
Transcription
Translation
DNA
RNA
Protein

Figure 2.1: **Central dogma of molecular biology.**

- ribosomal RNAs (rRNAs) are the most abundant type of RNA in any given cell (constituting about 80% of the total RNA produced). A long sub-unit (called *28S* in humans) and a small sub-unit (*18S*) assemble around messenger RNAs to decipher the protein code and produce proteins. Ribosomal RNA is not translated, and hence does not undergo the maturation process of mRNAs. It is not poly-adenylated unless it requires degradation by the cell.

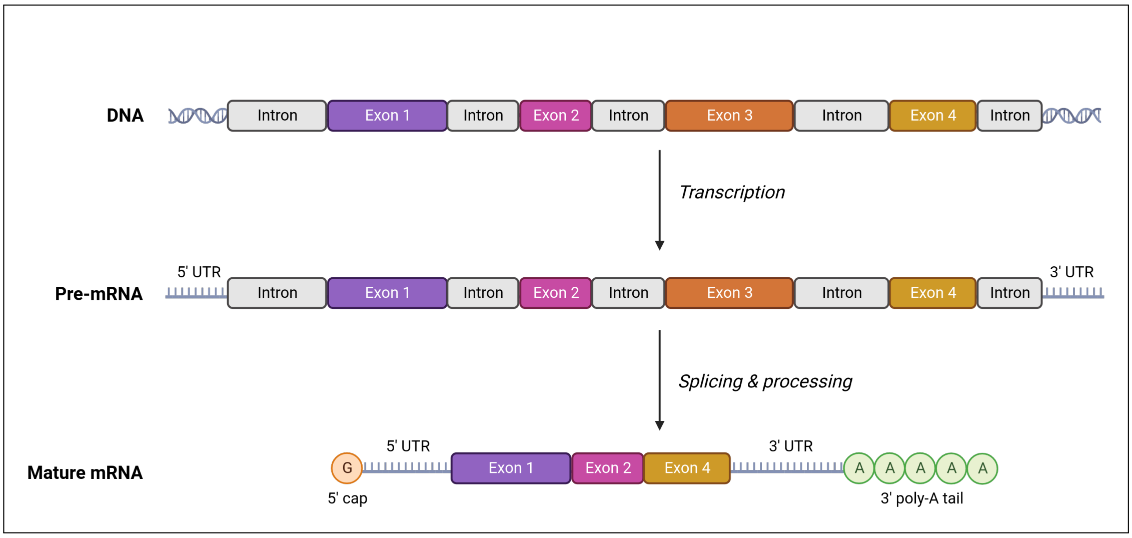

Figure 2.2: **Maturation of mRNAs.**

Why should we study RNA? All our cells, whether from skin, liver, brain, etc, contain the same genetic information, that is the same DNA inherited from our parents. What makes them fundamentally different is which genes they express (*i.e.* transcribe), and how those expressed genes are further translated into proteins. While we know that genetic diseases such as cancer are often characterized by anomalies at the DNA level, understanding how different types of healthy cells differ cannot be approached at the DNA level. The last 3 decades have therefore seen a major shift of attention to RNA, and the development of ultra-high throughput sequencing technologies have allowed major discoveries at the expression level.

Recently, RNA modifications were broadly re-discovered transcriptome-wide and linked with cancer diversity, prognoses and drug resistance. Collectively forming epitranscriptome with over 150 different modification types (Helm and Alfonzo 2014), RNA modifications expand the chemical diversity of RNA and provide specific structure building blocks or points of interactions with the other molecules. This epitranscriptome dynamically re-defines the stability, turnover and transitional performance of the RNA at all levels, including mRNA encoding and rRNA producing the proteins. Cancer cells hijack the epitranscriptome to mold the malignant mRNA and rRNA function towards their increased resistance and proliferation (Nombela, Miguel-Lopez, and Blanco 2021). Here I briefly present 4 of the most abundant RNA modifications that have been at the core of the second contribution of this chapter. Impact of those modifications in cancer are also briefly illustrated in Figure 2.3, which was designed by Nikolay Shirokikh for a common grant application.

- N(6)-methyladenosine (m6A) is the most abundant and a major destabilising mark in mRNA, it is involved in the regulation of alternative splicing (S. Wang et al. 2022), and it can have a direct and selective stimulatory effect on translation (Meyer 2019). Aberrant m6A in mRNA is strongly associated with cancer and can impose oncosuppressive or oncogenic effects depending on malignancy type (T. Sun, Wu, and Ming 2019).

- 5-methylcytosine (m5C) is the second most abundant mRNA modification, it regulates stability (J. Liu et al. 2022; X. Yang et al. 2017) and has been shown to be strongly associated with many cancers and contribute to the malignant heterogeneity (Gu et al. 2023; Nombela, Miguel-Lopez, and Blanco 2021; Song et al. 2022).

- Pseudouridine (pU, or Ψ) is the most frequent modification in all RNA and is also prominent in mRNA. It is involved in the accuracy and efficiency of splicing and translational decoding (Borchardt, Martinez, and Gilbert 2020), and has vastly been used in mRNA vaccines to suppress harmful unspecific immunogenic effects, as well as increase mRNA stability and translatability (Karikó et al. 2008). Importantly, decreased Ψ in rRNA leads to amino acid misincorporation, reduced oncosuppressor expression and aggressive tumour phenotype (Cerneckis et al. 2022).

- N4-acetylcytidine (ac4C), originally found to be conservely deposited into rRNA by NAT10, has recently been shown to broadly feature in mRNA (Jin et al. 2020; W. Zhao et al. 2019). Elevated levels of ac4C are notable for increasing cell resistance and survival (Luo et al. 2023).

Studying RNA has been made incredibly easier with the development of high-thoughput sequencing technologies. Most approaches dedicated to capturing RNA rely on the amplification of fragments to sequence via two steps. First the RNA transcript is reverse-transcribed into complimentary DNA (cDNA) via the same complementary dictionary than transcription, a C for a G, a T for an A, etc, and this cDNA is then amplified via several rounds of Polymerase Chain Reaction (PCR) to increase the amount of starting material (the library size) of the sequencing experiment. This has two major consequences: any perturbation of the initial RNA transcript (such as methylation) is erased during reverse transcription, and PCR induces amplification errors (SNPs) at a non-negligible rate.
Recently, Oxford Nanopore Technology developed a novel sequencing technology bypassing

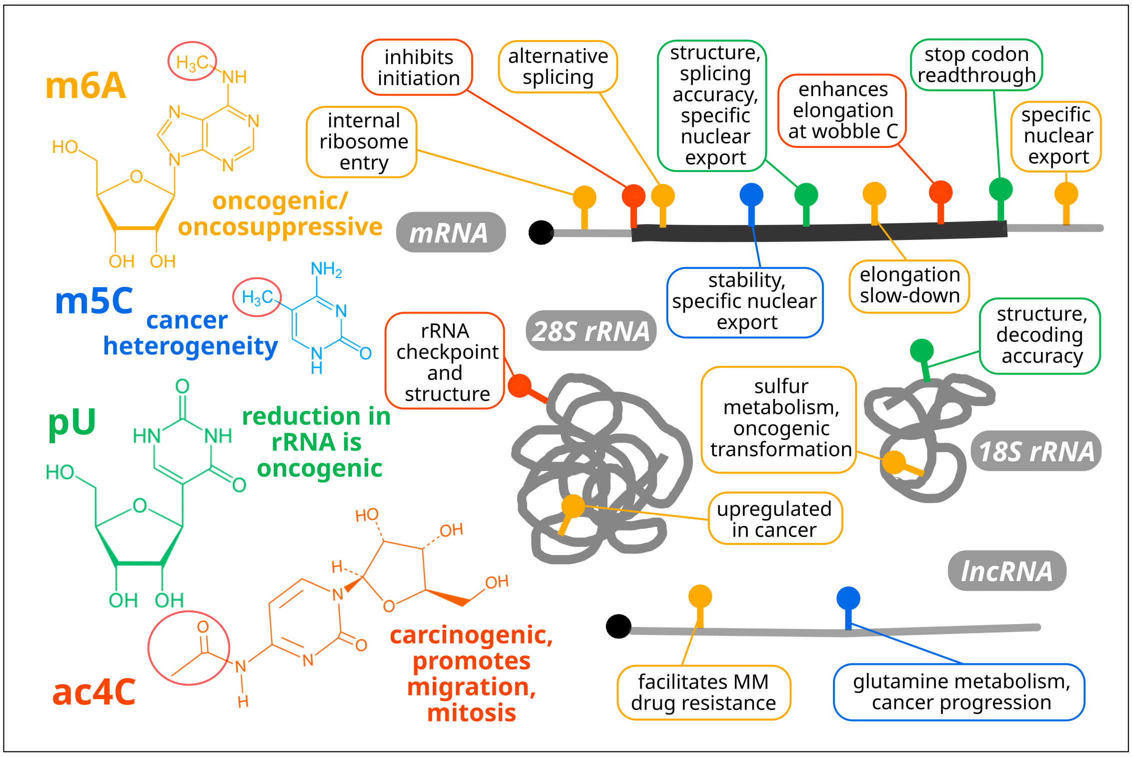
m6A
oncogenic/ oncosuppressive
m5C
cancer heterogeneity
pU
reduction in rRNA is oncogenic
ac4C
carcinogenic, promotes migration, mitosis
mRNA
28S rRNA
18S rRNA
lncRNA
inhibits initiation
alternative splicing
internal ribosome entry
structure, splicing accuracy, specific nuclear export
enhances elongation at wobble C
stop codon readthrough
specific nuclear export
stability, specific nuclear export
elongation slow-down
rRNA checkpoint and structure
sulfur metabolism, oncogenic transformation
structure, decoding accuracy
upregulated in cancer
facilitates MM drug resistance
glutamine metabolism, cancer progression

Figure 2.3: **Impact of main RNA modifications in cancer.**

both steps: Direct RNA Sequencing (DRS). After extraction from the cell, transcripts are navigated through a nanopore by a protein capturing the poly(A) tail of the transcripts. The pore measures an ionic current that is specifically affected by the navigation of nucleotides, and basecalling algorithms such as guppy (Wick, Judd, and Holt 2019) translate the current measures into sequences of nucleotides (see Figure 2.4). DRS can therefore capture the sequence of whole transcripts yielding long reads that can directly be mapped to the appropriate isoform without relying on deconvolution from junction reads.

ONT direct RNA sequencing should therefore present three critical improvements compared to classic short-read sequencing technologies such as Illumina: quantification can be performed at the isoform level, it is not subject to PCR bias (such as higher amplification of transcripts with high GC content, or introduction of sequencing errors), and it potentially has the ability to identify RNA modifications such as methylation of nucleotides.

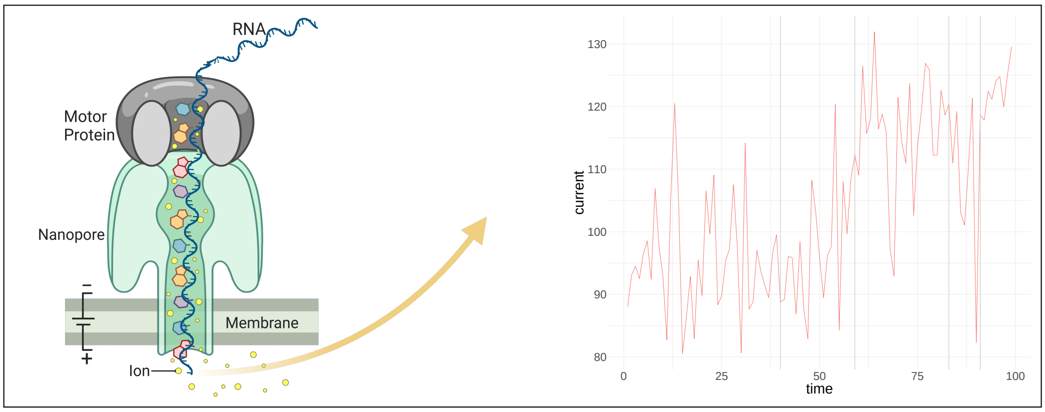

Figure 2.4: **Direct RNA Sequencing with ONT.**

The major aim of the works presented in this chapter is to evaluate the feasibility and interest of using ONT direct RNA sequencing in Multiple Myeloma patient samples. In the first part I will present INDEGRA, a statistical framework to evaluate the degradation of RNA from ONT sequencing data. In the second part I will present SWARM, a deep-learning algorithm to predict the four major RNA modifications presented above from DRS transcripts. Finally I will present the results of those two pipelines on 9 patient samples and discuss the limitations of the approach, as well as future plans to improve the feasibility and outcome of INDEGRA and SWARM in patient samples.

2.2 INDEGRA: Integrity and Degradation of RNA analysis

This work is currently being formatted into a manuscript. It has involved the participation of Agin Ravindran, PhD student at ANU, who has performed all the wet-lab experiments, as well as Bhavika Khumar, Masters student at ANU who has done exploratory analysis of the impact of degradation on typical downstream analysis such as differential gene expression and isoform usage. The project was co-led with Nikolay Shirokikh, associate professor at ANU, with the participation of Eduardo Eyras, professor at ANU.

2.2.1 RNA degradation

Unlike DNA, meant to last the whole cell-life, RNA is subject to rapid decay induced by various mechanisms, including biological, technical, chemical and enzymatic degradation. Messenger RNA's expected half-life is in the order of 2 to 20 minutes (Baudrimont et al. 2017), while DNA is endowed with various repair mechanism to ensure its long-lasting property, a cell-life being on average 7 to 10 years (Cooper and Adams 2023). In-vivo RNA degradation is a physiological and prevalent process in the life cycle of RNA. It occurs in a transcript-specific manner and is largely a consequence of natural mRNA turnover but can also be driven by a sample specific factor such as oxidative, starvation and chemical stresses. On the other hand, in-vitro degradation involves chemical and enzymatic degradation and can occur during sample acquisition, sample storage or preparation, RNA extraction or library preparation or possibly during sequencing itself (for an extensive review of RNA degradation, see Houseley and Tollervey 2009). Regardless of its origin, variability in the extent of RNA degradation creates a source of bias between compared samples, which remains a major challenge in interpreting RNA-based data.

The most popular metric to estimate RNA degradation in samples is the RNA integrity number, RIN (Schroeder et al. 2006) which is an electrophoretic mobility-based technique assessing degradation through ribosomal RNA concentration ratios. Considered as a de-facto standard, this metric, which ranges from 1 (very poor RNA quality) to 10 (no degradation), suffers dramatic limitations: it quantifies mRNA quality based on a very assumptive model of ribosomal RNA decay, and it does not provide transcriptome-wide estimates.

Without getting into the biology of each regulation mechanism, I will distinguish here 3 types of degradation, and explain how their effect is recorded by Direct RNA sequencing.

- **Fragmentation** occurs when an mRNA transcript is chopped into several pieces (an example is endonuclease-mediated mRNA decay, see for instance Schoenberg 2011). In this case, even though all fragments are retained (at least initially) within the cell, only the most 3'-end fragment, which contains the poly(A) tail captured by the motor protein, will be sequenced by DRS (see top panel of Figure 2.5).

- **5' end degradation** occurs when an enzyme progressively eats the RNA transcript from the 5' end, *i.e.* the upstream side. This process, also called 5'→ 3' exoribonuclease degradation, is essential in the natural degradation of RNA for ensuring the fidelity of RNA transcripts by extensive turnover (Houseley and Tollervey 2009). Unless the process is completed and the poly(A) tail affected by exoribonuclease, the whole remaining piece of RNA will be able to transit through the pore, hence leading to sequencing reads shorter than the initial transcript but representative of the true fragment in the cell (see middle panel of Figure 2.5).

- **3' end degradation** occurs when an enzyme progressively eats the RNA transcript from the poly(A) tail. This type of degradation is typically done by the exosome, contributing to regulated RNA decay as well as quality control of gene expression by degrading defective transcripts (Ibrahim, J. Wilusz, and C. J. Wilusz 2008). Because the poly(A) tail is the first part to be disassembled from the transcript, none of the remaining piece of the initial transcript will be able to transit through the pore, hence a complete lack of representation of the RNA in the sequencing experiment (see bottom panel of Figure 2.5).

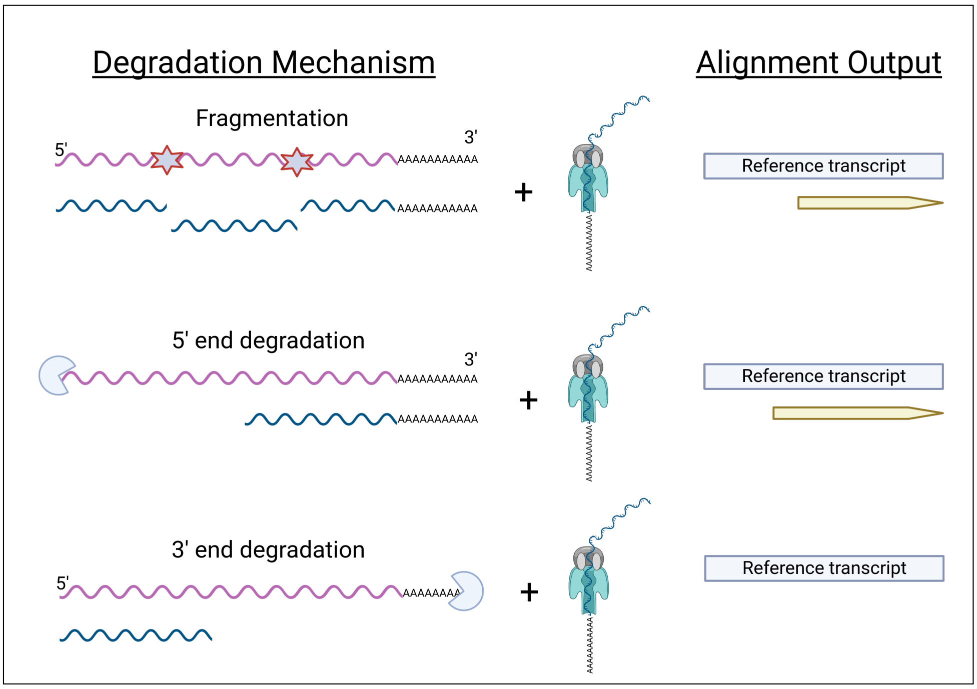

Figure 2.5: **Main degradation mechanisms and effect on DRS.** Purple RNA represents the original mRNA in the cell prior to degradation, while blue RNA fragments represent the remaining fragments after degradation (left panels). After going through the nanopore, sequenced reads are aligned to a reference transcript (which can be built from the data). Only fragments with poly(A) tails transit through the pore, so reads will always align to the 3' end of the reference (right panel). The top row illustrates degradation by fragmentation, resulting in several RNA fragments, the middle raw illustrates 5'end degradation, resulting in a unique fragment with poly(A) tail, and the bottom row illustrates 3'end degradation, resulting in a unique RNA fragment with no poly(A) tail.

One can note that the picture is slightly different in case the RNA is additionally poly-adenylated in vitro (*i.e.* a poly(A) tail is artificially added to every RNA transcript in the cell before sequencing, which we will refer to as poly(A)- data), as in this case any available fragments will transit through the pore. In the natural scenario presented above (referred to as poly(A)+), on which I will focus for the remainder of this section, the only poly(A) sequences present will be those of the naturally-occurring 3' poly(A) stretches. If a standard sequencing protocol is used, in the first case, all fragments generated before in vitro polyadenylation will be able to interact with the sequencing pore as they will all have 3' poly(A), whereas in the second case, only 3'-terminal fragments will be sequenced . In the work presented here I will focus only on degradation by fragmentation. Extension to other types of degradation as well as poly(A)- data will be briefly discussed at the end of this section.

2.2.2 Mathematical model of RNA degradation

There are several pre-processing steps that I will not describe in detail here (even though they represented intensive calibration and evaluation on experimental data). Briefly, they consist in aligning the reads to a reference sample, filtering alignments based on insertion and deletion length, 3'end mapping (since we use poly(A)+ data, and the sequencing is performed from 3' to 5', all reads should align to the reference's 3' end), and soft-clip length. From there we then compute the adjusted length of each read as the difference of mapped 3' and 5' positions, regardless of any insertions, deletions or soft-clip. Of note, even though the poly(A) tail is sequenced by the nanopore, its corresponding part in the read is cleaved by the preprocessing algorithm and its length is not included in the adjusted length.

Random fragmentation, understood as fragmentation occurring uniformly randomly across the transcripts, can be modelled as a Bernoulli process of rate κ. In our context, this is interpreted as fragmentation events occurring in between two nucleotides with probability κ, independently of what happened in between other nucleotides. This implies that for a transcript of length L, the number of fragmentation events Y having occurred follows a binomial distribution with parameters κ and $L-1$. To fit to the traditional statistical segmentation framework, we will introduce the following notation: if p fragmentation events occurred within a transcript of length L, we denote $\tau_1, \ldots, \tau_p$ their location on $\{2, \ldots, L\}$ with the conventions $\tau_0 = 1$ and $\tau_{p+1} = L+1$ so that $\tau_k = \ell$ actually means that the kth fragmentation event occurred between nucleotide $\ell - 1$ and ℓ, and those events define fragments $[\tau_k; \tau_{k+1}[$.

Because DRS only captures the last fragment, we are interested in recovering the length X of the last fragment of each transcript. Simple conditional probabilities lead to

$$\begin{aligned}
P(X=\ell) &= \sum_{p=0}^{+\infty} P(X=\ell, Y=p) \\
&= P(Y=0)\mathbb{1}_{\{\ell=L\}} + \sum_{p=1}^{+\infty} P(\tau_p = L-\ell+1 | Y=p) P(Y=p) \mathbb{1}_{\{\ell<L\}} \\
&= (1-\kappa)^{L-1}\mathbb{1}_{\{\ell=L\}} + \sum_{p=1}^{L-\ell} \frac{\begin{pmatrix} L-\ell-1 \\ p-1 \end{pmatrix}}{\begin{pmatrix} L-1 \\ p \end{pmatrix}} \begin{pmatrix} L-1 \\ p \end{pmatrix} \kappa^p (1-\kappa)^{L-1-p} \mathbb{1}_{\{\ell<L\}} \\
&= (1-\kappa)^{L-1}\mathbb{1}_{\{\ell=L\}} + \kappa(1-\kappa)^{\ell-1} \sum_{p=0}^{L-\ell-1} \begin{pmatrix} L-\ell-1 \\ p \end{pmatrix} \kappa^p (1-\kappa)^{L-\ell-1-p} \mathbb{1}_{\{\ell<L\}} \\
&= (1-\kappa)^{L-1}\mathbb{1}_{\{\ell=L\}} + \kappa(1-\kappa)^{\ell-1}\mathbb{1}_{\{\ell<L\}},
\end{aligned}$$

from which we can define a maximum likelihood estimator $\hat{\kappa} = \dfrac{\sum \mathbb{1}_{\{X_i<L\}}}{\sum X_i - \sum \mathbb{1}_{\{X_i=L\}}}$. Note that the $\sum \mathbb{1}_{\{X_i=L\}}$ in the denominator is typically negligible compared to $\sum X_i$ for any typical transcript. The cumulative density function is then

$$\begin{aligned}
F(x) = P(X \leq x) &= \sum_{\ell=1}^{x} P(X=\ell) \\
&= \kappa \sum_{\ell=1}^{x} (1-\kappa)^{\ell-1} + (1-\kappa)^{L-1}\mathbb{1}_{\{x=L\}} = 1 - (1-\kappa)^x \mathbb{1}_{\{x<L\}}
\end{aligned}$$

2.2.3 Correction for DRS bias

The Nanopore DRS technologies are known to present two kinds of bias. First, they fail to capture efficiently short fragments, typically fragments smaller than 150 or 200 bases. Second, the RNA can get stuck in the pores, which can be resolved by applying a high reversed voltage and ejecting the fragment back into the input solution. This leads to reads shorter than their original fragments, a censoring information that is captured in the sequencing summary file of the experiment.
To account for the first bias, we can compute the distribution of the read length conditionally on seeing only reads longer than a given threshold s.

$$\widetilde{F}_s(x) = P(X \le x | X \ge s) = \frac{F(x) - F(s)}{1 - F(s)} = 1 - (1-\kappa)^{x-s} \mathbb{1}_{\{s \le x < L\}}$$

from which we obtain the new maximum likelihood estimator

$$\widetilde{\kappa} = \frac{\sum \mathbb{1}_{\{s \le X_i < L\}}}{\sum X_i \mathbb{1}_{\{s \le X_i\}} - \sum \mathbb{1}_{\{X_i = L\}}}$$

To account for the second bias, we use censored statistics inspired from the non-parametric survival estimator approach of Kaplan and Meier (Kaplan and Meier 1958), which in essence assigns weights to each observation depending on its censoring value and its closest neighbour's values. The observations are now couples (X_i, C_i) where X_i remains the observed length of read i, and $C_i = 0$ if the read was ejected from the sequencer prematurely, 1 otherwise. From the combination of (X_i, C_i), we order the reads by increasing length to get $(X_{(i)}, C_{(i)})$ and create the weight vector W as in the following example:

$$\begin{array}{ccccc}
(i) & C_{(i)} & & w_{(i)} & \\
1 & 1 & & 1/n & \\
2 & 1 & & 1/n & \\
3 & 0 & & 0 & \\
4 & 1 & \rightarrow & 1/n * \frac{n-2}{n-3} & , \\
5 & 1 & & 1/n * \frac{n-2}{n-3} & \\
6 & 0 & & 0 & \\
7 & 0 & & 0 & \\
8 & 1 & & 1/n * \frac{n-2}{n-3} * \frac{n-5}{n-7} &
\end{array}$$

so that each censored read has a null weight, and other reads (i) have weight $w_{(i-1)}$ if $C_{(i-1)} = 1$, and $w_{(i)} = w_{(i_1-1)} * \frac{n-i_1+1}{n-i_2}$ if $C_{(i-1)} = 0$, where i_1 and i_2 are respectively the first and last indexes of the last block of censored reads before (i), and with convention $w_{(0)} = 1/n$, $C_{(0)} = 1$. Note that we do recover $\sum_i w_{(i)} = 1$, and if no read is censored, then for all i, $w_i = 1/n$.

The final maximum likelihood estimator of κ is then

$$\hat{\kappa} = \frac{\sum w_i \mathbb{1}_{\{s \le X_i < L\}}}{\sum w_i X_i \mathbb{1}_{\{s \le X_i\}} - \sum w_i \mathbb{1}_{\{X_i = L\}}}$$

2.2.4 The DTI metric

Despite its limitation, RIN is an extremely popular RNA quality measures which values are instantly understood by biologists. We therefore proposed to transform our fragmentation estimation measure into a Direct Transcriptome Integrity (DTI) number via the following one-to-one application:

$$f(x) = \begin{cases} 1 + (b-1)\left(\dfrac{x-1}{s-1}\right)^{\frac{1}{2}} & \text{if } x \leq s \\ 10 + (b-10)\left(\dfrac{x-1}{s-1}\right)^{-\frac{b-1}{2(10-b)}} & \text{if } x > s \end{cases}$$

where parameters $s = 1800$ (the inverse fragmentation rate from which transcripts are considered as having an excellent integrity) and $b = 8$ (the DTI value at threshold s) were chosen to provide very slow increases at extreme fragmentation values, and cover a wide range of DTI values in the most frequently observed fragmentation rates, as illustrated in the left panel of Figure 2.6. This function can be applied at the transcript level, and used to provide a sample measure as the median of all transcripts integrity values, *i.e.*

$$DTI = median(f(\kappa_t)) = f(median(\kappa_t))$$

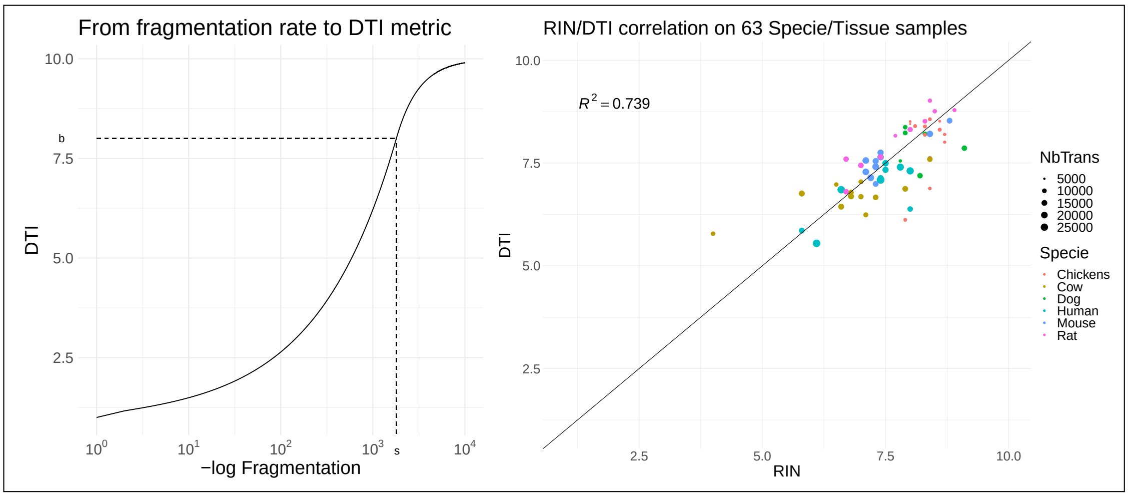

Figure 2.6: **DTI metric and RIN. a)** fragmentation rate to DTI mapping function, in $\log(1/\kappa)$, so that the x-axis can be interpreted as average number of nucleotides in between two fragmentation events. The function is designed to be smooth, with value 1 in $\kappa = 1$ (one cut every nucleotide) and tend to 10 when κ tends to 0, with the greatest range of DTI values in the most frequently observed fragmentation rates (in between one cut per 500 to one cut to 1800 nucleotides). **b).** Correlation between our DTI estimation and RIN numbers on 63 samples from 6 species (Rat, Mouse, Cow, Chickens, Dog and Human) and 6 tissues (liver, hippocampus, Cerebellum, frontal cortex, testis and skeletal muscle).

We applied our method to the analysis of 63 samples from 6 species (Rat, Mouse, Cow, Chickens, Dog and Human) and 6 tissues (liver, hippocampus, Cerebellum, frontal cortex, testis and skeletal muscle) and computed the correlation between our DTI metric and RIN measures (right panel of Figure 2.6). The very high correlation obtained (higher than 0.7) on this heterogeneous dataset proves that our metric is valid in a wide range of scenari.

2.2.5 RNA integrity and Differential Transcript Expression

It is debated whether RNA degradation should induce bias in differential transcript expression. Indeed, expression estimation is purely based on the number of reads assigned to each transcript. In our degradation by fragmentation model, since only the most 3'end fragment is captured by the sequencer, the number of reads obtained with or without degradation should be the same. We have performed several differential transcript expression analysis (based on the standard DESeq2 pipeline, (Love, Anders, and Huber 2014)) comparing biological replicates of same tissues with different degradation levels (evaluated by RIN) and found in all cases a few dozen "differentially expressed" hits, as illustrated in Figure 2.7 showing the correlation between number of hits and degradation estimation differences between compared samples. Figure 2.8 also illustrates an example of comparing 2*2 samples from Rat Cerebellum tissues which we will use to describe our correction approach.

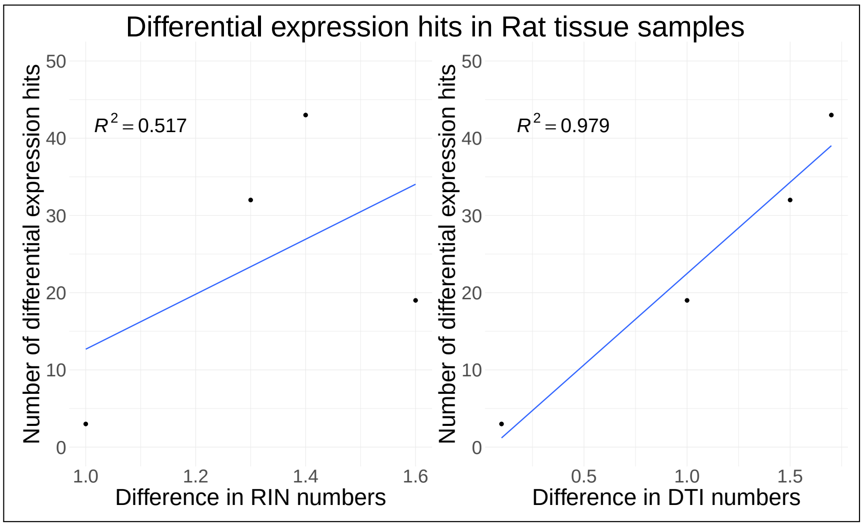

Figure 2.7: **DTE hits in different Rat Samples comparison, and correlation with degradation.** Standard DTE analysis was performed on 4 datasets from 4 different Rat tissues where replicates have different RIN numbers. Interestingly, the number of DTE hits correlates better with the difference in DTI numbers than RIN numbers.

We suggested that those differential hits may be artifacts of different degradation level and proposed to correct transcript read counts for transcript degradation, in a similar way as what was proposed for GC content or transcript length (see Risso et al. 2011). The left panel of Figure 2.9 shows an example of LOWESS regression of the read counts versus their DTI metric in the four Rat Cerebellum samples used to illustrate the volcano plot of Figure 2.8. The normalisation approach is as follows: for each sample we fit a LOWESS regression on the log counts versus DTI estimation and recover fitted counts as the residuals shifted to scale the raw counts. Let Y_i be the raw count of transcript i, and $\hat{\gamma}_i$ the fitted value of transcript i from the lowess regression between $\log Y_i$ and DTI_i. Then the sample-specific normalised expression measure Y_i' is obtained as

$$Y_i' = exp(\log Y_i - \hat{\gamma}_i) + median(Y_1, \dots Y_n).$$

Note that this normalised expression value needs not be a count. The middle panel of Figure 2.9 shows the lowess regression of those normalised expression measures versus the DTI metric in the same four Rat samples.

From there we propose two options to proceed with differential transcript expression.

- The first option consists in further normalising the expression values so that the expression values of different samples are represented in the same signal range (this typically is necessary when samples have not been sequenced with the same depth). Several approaches have been proposed to deal with this *between-lane* normalisation, including Full Quantile, RPKM, upper-quartile, etc (see Bullard et al. 2010 for a review), but we limited our tests

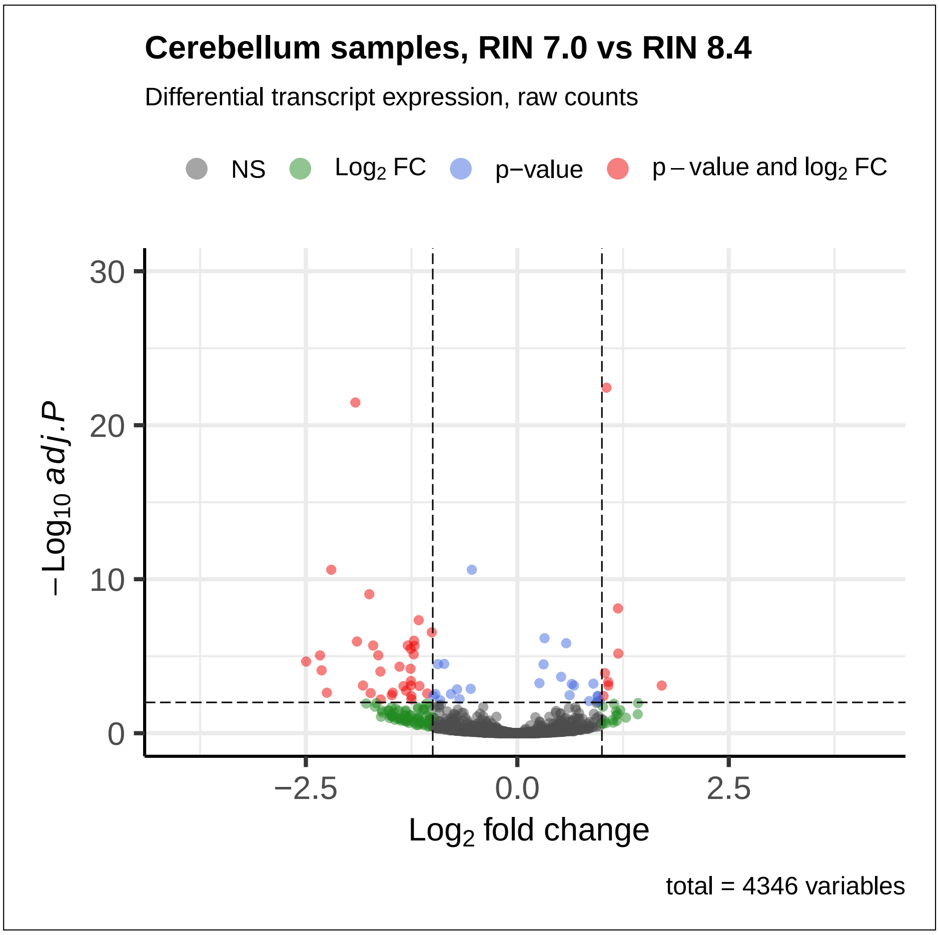

Figure 2.8: **DTE in replicates from same tissue.** Comparison of transcript expression in 2 biological replicates of Rat Cerebellum tissues with different degradation level (RIN 7.0 vs RIN 8.4), each sequenced in two technical replicates. 43 transcripts are found differentially expressed (with log-fold-change above 1) at FDR control of 0.01.

to the full-quantile normalisation approach. This technique forces the empirical distribution of expression values to be identical between all samples by independently ranking the transcript of each sample by expression value (*i.e.* $Y_{(1)j} \leq Y_{(2)j} \leq \ldots$) and setting, for all j, $Y_{(i)j} = \bar{Y_{(i)}}_{.}$ (the right panel of Figure 2.9 illustrates the Lowess regression on the final normalised expressions). The *offsets*, equal to the difference between the normalised and unnormalised expression values, can then be supplied to standard R packages for differential expression analysis, such as DESeq2, as normalisation factors to be used in the negative binomial model.

- The second option consists in rounding the sample-specific normalised expression measure Y_i' to create a matrix of normalised counts, and provide this matrix as input to the differential expression analysis tool, which further performs the *between-lane* normalisation as in standard DE analysis.

The volcano plots of Figure 2.10 illustrate the analysis of differentially expressed transcripts in the Rat Cerebellum samples after the two correction methods (offset on the left, rounded counts on the right), reducing the number of hits from 43 to 12 in both cases. The first method is slightly more elegant from a statistical point of view (in particular, no rounding needs to be done), and extensive tests on the sets of rat samples (including 5 different tissues, each time with 2 biological

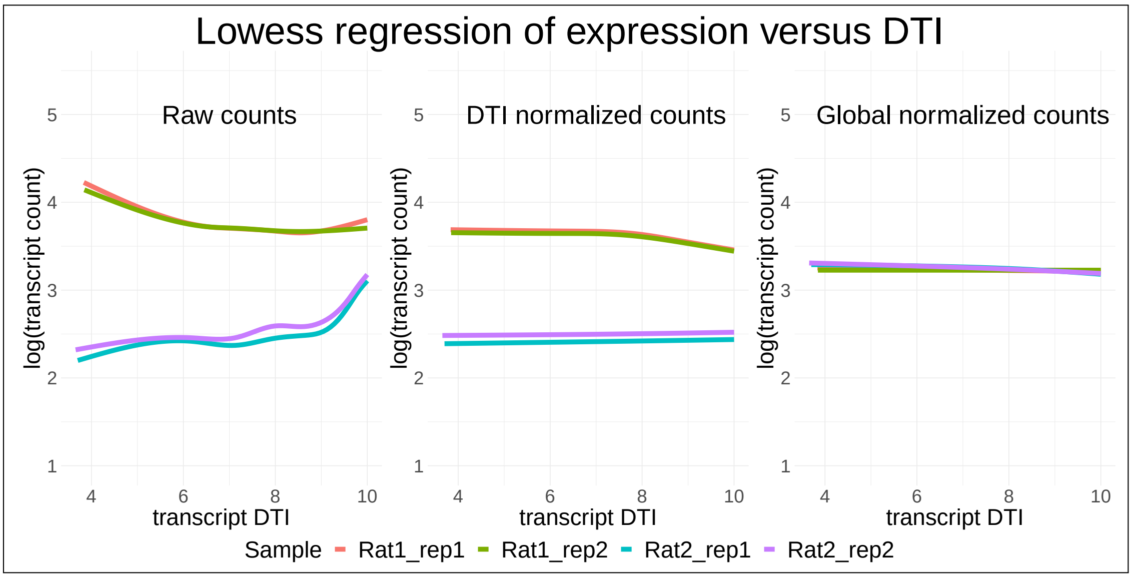

Figure 2.9: **Correcting counts for degradation level. a)** LOWESS regression fits between log raw counts and DTI estimates. **b)** LOWESS regression of the lane-specific normalised expression values and DTI estimates. The normalised expression values are obtained by shifting the residuals of the Lowess regression by the median expression of the sample. **c)** LOWESS regression of the between-lane normalised expression values and DTI estimates. The final normalised expression are quantile-normalised lane-specific normalised expression values.

times 2 technical replicates, with similar or different RIN numbers) and artificial DE analysis (see next paragraph), seems to indicate better power for a similar number of false-positive hits, and we therefore advise to use it in practice. Our algorithm returns both possibilities to allow more versatility with downstream softwares.

Applying our strategy to the four tissue datasets shows that INDEGRA greatly reduces the set of false-positive hits, as illustrated in Figure 2.11 showing the number of hits found on raw data and after using INDEGRA.

To ensure that our normalisation procedure does not correct for true differentially expressed transcripts, we performed a number of simulation studies where we artificially inflated the (raw) counts of a random subset of transcripts in either condition (for each artificially inflated transcript, the condition was sampled with probability $1/2$, and both its technical replicates were inflated) by a 2 to 3 factor uniformly drawn independently for each transcript. We then compared the number of transcripts identified before normalisation and by both normalisation approaches in three scenarios: in the “High” case, selected transcripts are randomly selected from the highest quantile of expression, in the “Medium” case they are randomly selected from the set of transcripts with expression between the first and third quartile, and in the “Low” case they are selected from the lowest quantile of expression. Figure 2.12 shows the number of false positives and true (artificially) differentially expressed transcripts recovered by DESeq2 when no normalisation is performed, and when INDEGRA is used with offset or with rounded counts output.

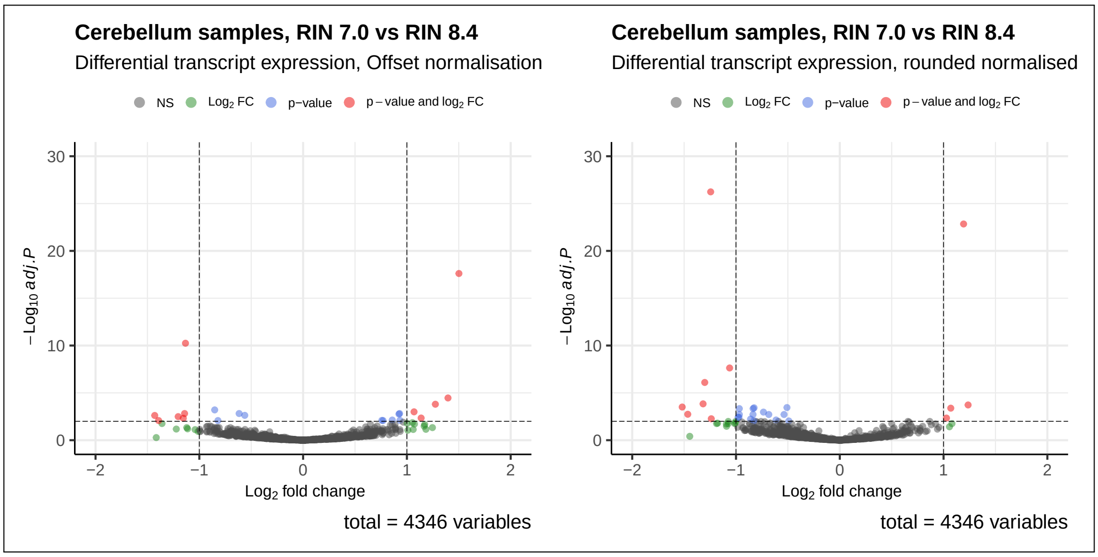

Figure 2.10: **DTE after DTI normalisation.** Comparison of transcript expression in 2 biological replicates of Rat Cerebellum tissues with different degradation level (RIN 7.0 vs RIN 8.4), after correcting for Direct Transcript Integrity. Left panel results from using offset normalisation in DESeq, leading to 12 transcripts found differentially expressed (about 28% of the initial hits). Right panel results from using rounded counts after lowess regression as input in DESeq, also leading to 12 transcripts found differentially expressed.

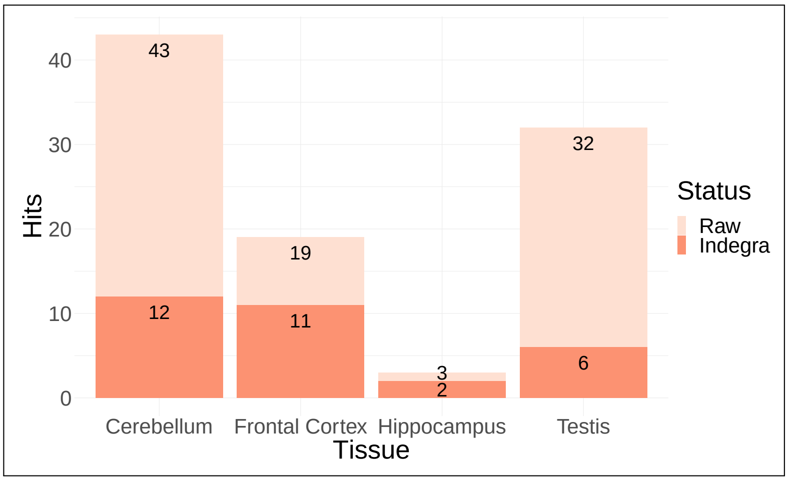

Figure 2.11: **DTE hits on raw data and after DTI normalisation.** INDEGRA significantly reduces the set of false positive hits on samples from same tissues where no differential expression is expected.

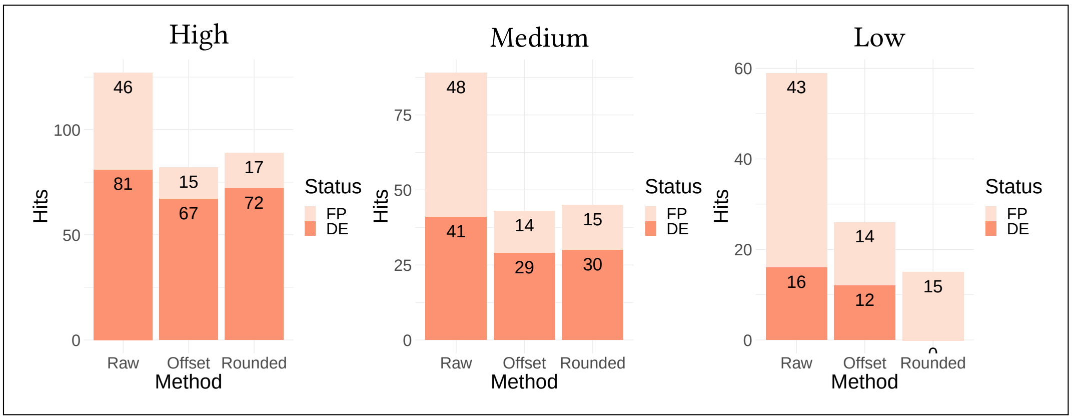

Figure 2.12: **Impact of DTI normalisation on artificially differentially expressed genes.** We simulated 100 differentially expressed transcripts by artificially inflating raw counts by a 2 to 3 factor, regardless of their degradation level, and applied DESeq2 pipeline with and without DTI normalisation in three scenarios where the transcripts are selected based on their expression level. Transcripts were selected from either high, medium, or low expressed sets of transcripts (Left, Middle and Right panel). Our approach reduces the false positive hits by a factor three, while retaining over 70% of the true positive hits.

2.2.6 RNA integrity and biological degradation

While it is important to correct for RNA integrity in the differential expression analyses, actual biological degradation might be the primary interest of DRS experiments. Just as it is necessary to take into account sequencing depth when comparing transcript expression between conditions based on read counts (this can be done via modelisation or normalisation), comparing biological degradation at the transcript level based on fragmentation rate (or DTI) requires to take into account the overall sample quality that can be attributed to technical degradation.
Figure 2.13 shows an example of correlation of transcript fragmentation rates between biological and technical replicates (the same Rat Cerebellum tissues used previously to illustrate differential expression). Samples have an excellent correlation, with a tendency for higher correlation within technical replicates than between biological replicates.

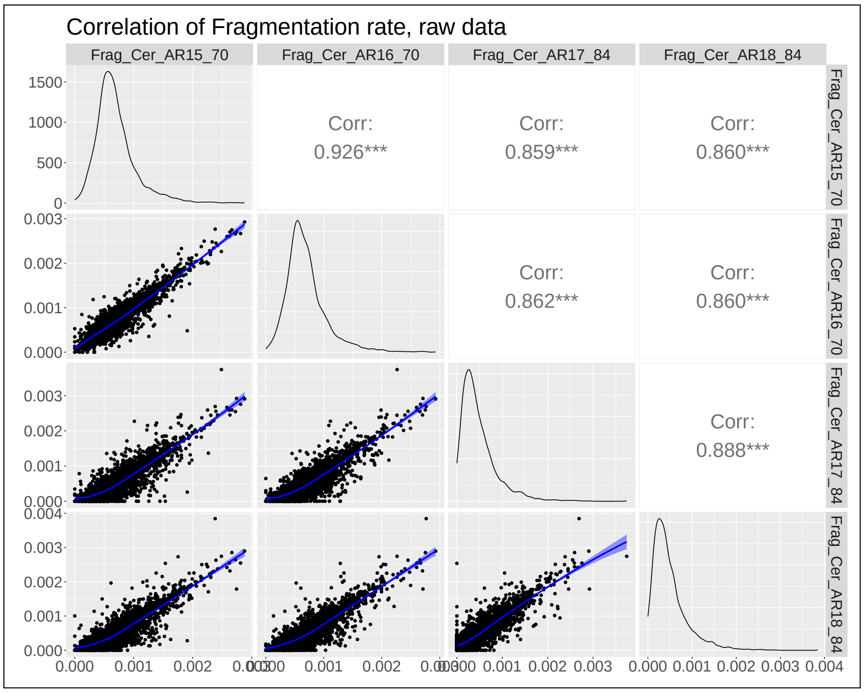

Figure 2.13: **Correlation of fragmentation estimates values between replicates.**

Here, we use a Bayesian model to test the equality of a transcript biological degradation between two conditions. The generative model, illustrated in Figure 2.14, can be described as follows:

- $E_t \sim \mathcal{B}(p)$, denotes the event of equality of biological degradation, specifically $E_t = E_0$ corresponds to same degradation in the 2 conditions, while $E_t = E_1$ corresponds to different degradations.

- $\tau_{\mathbf{t}} = \{\tau_{t1}, \tau_{t2}\}$, are the biological degradation parameters for each condition.
  $\tau_{\mathbf{t}}|E = E_0, \tau_{t1} = \tau_{t_2} = \tau_t, \tau_t \sim \beta(a,b)$.
  $\tau_{\mathbf{t}}|E = E_1, \tau_{\mathbf{t}} \sim \beta(a,b)^{\otimes 2}, \tau_{t1} \neq \tau_{t_2}$.

- X_{tij} length of last fragment of transcript i in condition j after biological degradation,
  $P(X_{tij} = \ell|\tau_{\mathbf{t}}, E_t) = (1-\tau_{tj})^{L-1}\mathbb{1}_{\ell=L} + \tau_{tj}(1-\tau_{tj})^{\ell-1}\mathbb{1}_{\ell<L}$

- α_j technical degradation in condition j. We will suppose that α_j is a known parameter.

- Y_{tij} observed length of last fragment of transcript i in condition j, after technical and biological degradation, with

$$P(Y_{tij} = \ell | X_{tij}, \alpha_j, \tau_{\mathbf{t}}, \alpha, E_t) = (1-\alpha_j)^{X_{tij}-1} \mathbb{1}_{\ell = X_{tij}} + \alpha_j (1-\alpha_j)^{\ell-1} \mathbb{1}_{\ell < X_{tij}}$$

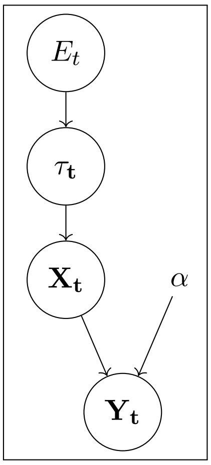

Figure 2.14: **Generative model for read-length.** The biological degradation parameter τ is drawn conditional to the event E describing same or different biological degradation between two conditions. This leads to biologically degraded transcripts of length X, which undergo a second degradation step with parameter α to produce the observed read-length Y.

For sake of simplicity, in the following computations we drop the transcript index t. Then

$$\begin{aligned} P(Y_{ij} = k | \alpha_j, \tau, E) &= \sum_{\ell=1}^{L} P(Y_{ij} = k | X_{ij} = \ell, \alpha_j, \tau, \alpha, E) P(X_{ij} = \ell | \alpha_j, \tau, \alpha, E) \\ &= (1-\tau_j)^{k-1} (1-\alpha_j)^{k-1} \big(1 - (1-\tau_j)(1-\alpha_j)\big)^{\mathbb{1}_{k<L}}. \end{aligned}$$

This is therefore equivalent to a model with a unique degradation step with parameter $1 - (1-\tau_j)(1-\alpha_j)$, as illustrated in Figure 2.15.

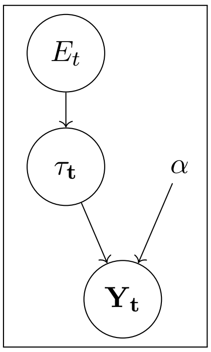

Figure 2.15: **Equivalent generative model for read-length.** Here again, the biological degradation parameter τ is drawn conditional to the event E describing same or different biological degradation between two conditions. The unique degradation step is performed with parameter $1 - (1-\tau_j)(1-\alpha_j)$ to produce the observed read-length Y.

$$\begin{aligned}
\mathcal{L}(\mathbf{Y}_1,\mathbf{Y}_2|\alpha,\tau,E) &= \prod_{i1}(1-\tau_1)^{Y_{i1}-1}(1-\alpha_1)^{Y_{i1}-1}\big(1-(1-\tau_1)(1-\alpha_1)\big)^{\mathbb{1}_{Y_{i1}<L}} \\
&\quad \prod_{i2}(1-\tau_1)^{Y_{i2}-1}(1-\alpha_2)^{Y_{i2}-1}\big(1-(1-\tau_2)(1-\alpha_2)\big)^{\mathbb{1}_{Y_{i2}<L}} \\
&= (1-\tau_1)^{m_1}(1-\alpha_1)^{m_1}\big(1-(1-\tau_1)(1-\alpha_1)\big)^{n_{1<}} \\
&\quad (1-\tau_2)^{m_2}(1-\alpha_2)^{m_2}\big(1-(1-\tau_2)(1-\alpha_2)\big)^{n_{2<}} \\
&= \left[\sum_{k=0}^{n_{1<}}\begin{pmatrix} n_{1<} \\ k \end{pmatrix}\alpha_1^k(1-\alpha_1)^{m_1+n_{1<}-k}\;\tau^{n_{1<}-k}(1-\tau_1)^{m_1}\right] \\
&\quad \left[\sum_{k=0}^{n_{2<}}\begin{pmatrix} n_{2<} \\ k \end{pmatrix}\alpha_2^k(1-\alpha_2)^{m_2+n_{2<}-k}\;\tau^{n_{2<}-k}(1-\tau_2)^{m_2}\right]
\end{aligned}$$

where $n_{j<}$ denotes the number of fragments in condition j with length strictly smaller than L, and $m_j = \sum(Y_{ij}-1)$. Finally, integrating over τ, we get:

$$\begin{aligned}
\mathcal{L}(\mathbf{Y}_1,\mathbf{Y}_2|\alpha,E_0) &= \sum_{k=0}^{n_{1<}}\sum_{k'=0}^{n_{2<}}\begin{pmatrix} n_{1<} \\ k \end{pmatrix}\begin{pmatrix} n_{2<} \\ k' \end{pmatrix}\alpha_1^k(1-\alpha_1)^{m_1+n_{1<}-k}\alpha_2^k(1-\alpha_2)^{m_2+n_{2<}-k} \\
&\quad \frac{\beta(a+n_{1<}-k+n_{2<}-k',b+m_1+m_2)}{\beta(a,b)} \\
\mathcal{L}(\mathbf{Y}_1,\mathbf{Y}_2|\alpha,E_1) &= \sum_{k=0}^{n_{1<}}\begin{pmatrix} n_{1<} \\ k \end{pmatrix}\alpha_1^k(1-\alpha_1)^{m_1+n_{1<}-k}\;\frac{\beta(a+n_{1<}-k,b+m_1)}{\beta(a,b)} \\
&\quad \sum_{k'=0}^{n_{2<}}\begin{pmatrix} n_{2<} \\ k' \end{pmatrix}\alpha_2^k(1-\alpha_2)^{m_2+n_{2<}-k}\;\frac{\beta(a+n_{2<}-k',b+m_2)}{\beta(a,b)}
\end{aligned}$$

so that

$$P(E_0|\mathbf{Y}_1,\mathbf{Y}_2) = \frac{(1-p)\mathcal{L}(\mathbf{Y}_1,\mathbf{Y}_2|E_0)}{(1-p)\mathcal{L}(\mathbf{Y}_1,\mathbf{Y}_2|E_0)+p\mathcal{L}(\mathbf{Y}_1,\mathbf{Y}_2|E_1)}.$$

For a given sample, the posterior distribution of τ can be computed as

$$\pi(\tau|\mathbf{Y}) = \frac{\sum_{k=0}^{n_<}\begin{pmatrix} n_< \\ k \end{pmatrix}\alpha^k(1-\alpha)^{m+n_<-k}\;\tau^{a+n_<-k}(1-\tau)^{b+m}}{\sum_{k=0}^{n_<}\begin{pmatrix} n_< \\ k \end{pmatrix}\alpha^k(1-\alpha)^{m+n_<-k}\;\beta(a+n_<-k,b+m)}$$

and the posterior expectation is

$$\hat{\tau}_{EAP} = \frac{\sum_{k=0}^{n_<}\begin{pmatrix} n_< \\ k \end{pmatrix}\alpha^k(1-\alpha)^{m+n_<-k}\;\beta(a+n_<-k+1,b+m)}{\sum_{k=0}^{n_<}\begin{pmatrix} n_< \\ k \end{pmatrix}\alpha^k(1-\alpha)^{m+n_<-k}\;\beta(a+n_<-k,b+m)}.$$

Choice of the hyperparameters

- Technical degradation parameters

To estimate α_1 and α_2 from the data, we use a heuristic approach based on the frequentist model and degradation estimation. First the relationship linking the different types of degradation $\kappa_j = 1-(1-\alpha_j)(1-\tau_j)$ provides

$$\alpha_j = \frac{1-\bar{\kappa}_j}{1-\bar{\tau}_j},$$

where and the mean is taken over all transcripts, and we assume that in the least overall degraded sample, $\bar{\tau}_j = \bar{\kappa}_j - 10^{-4}$, yielding $\hat{\alpha}_j \simeq 10^{-4}$. Then we assume that for the large majority of transcript, E_0 is verified (*i.e.* the transcript has the same biological degradation in both samples). Then we should have

$$\frac{1-\alpha_1}{1-\alpha_2} = \frac{1-\bar{\kappa}_1}{1-\bar{\kappa}_2}$$

giving an estimate of the second sample technical degradation rate. Even though this approach is largely heuristic, we can show on several examples that the posterior probabilities are very robust to the choice of the least degraded parameter, say α_1, provided this parameter ranges between $\left[10^{-5}, 5.10^{-4}\right]$.

- Prior probability of H_0

As is classical in Bayesian paradigms, the parameter p can be used to control the false positive rate. The user may tune it to fit the expected proportion of transcripts with different biological degradation. If there is no a priori knowledge to guide this choice, $p = 0.05$ or $p = 0.01$ can be used.

- Prior distribution of τ

Degradation rates are expected to be very small, of the ordre of 1 cut per thousand nucleotides, so that we expect τ to live in $[0, 10^{-3}]$. We therefore set $a = 2$ and $b = 1000$ so that the distribution mass essentially covers this range.

Figures 2.16 and 2.17 illustrate the results on the cerebellum dataset, comparing replicates with same technical degradation (Fig 2.16) and replicates with different technical degradation (Fig 2.17). The prior probability p was set to 0.05, yielding 3 false positive hits (out of 11108) in the first scenario, and 54 false positive hits out of 5178 tests in the second case, i.e. a false positive rate of 1%.

Implementation and extensive testing of this approach, including in particular proposing better estimation of the technical degradation parameters, and implementation of the tool to allow for replicates (with different technical degradation) in each condition, is still work in progress. Moreover, it could be interesting to extend also use this Bayesian framework to compute the posterior expectation of the total degradation rate κ (as opposed to using the maximum likelihood estimator from the frequentist framework exposed in the previous sections) in order to unify the different aspects of the work.

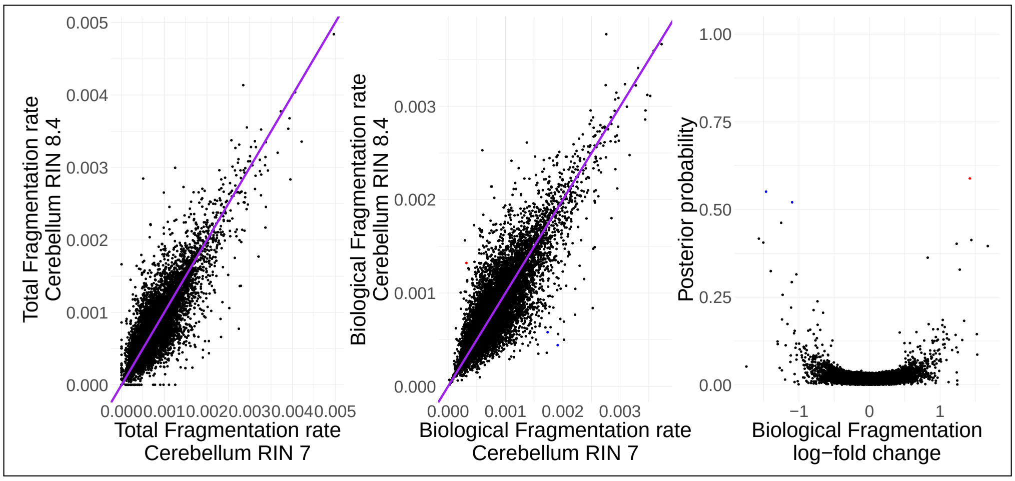

Figure 2.16: **Biological degradation comparison between replicates with same technical degradation.** a) total fragmentation estimate correlation between the replicates. b) scatter plot of the posterior expectation of τ in both replicates. c) pseudo-volcano plot, with posterior probability of E_1 with respect to log-fold change of the posterior expectation of τ in both replicates

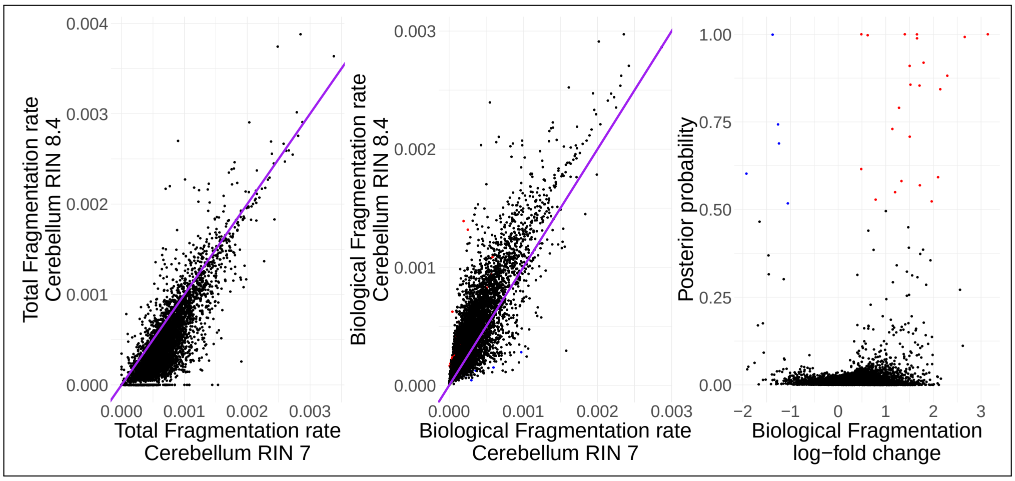

Figure 2.17: **Biological degradation comparison between replicates with different technical degradation.** a) total fragmentation estimate correlation between the replicates. b) scatter plot of the posterior expectation of τ in both replicates. c) pseudo-volcano plot, with posterior probability of E_1 with respect to log-fold change of the posterior expectation of τ in both replicates

2.2.7 Testing for random fragmentation

The mathematical model of degradation is based purely on random fragmentation assumption. It might hence be necessary to test whether degraded transcripts really originate from a random fragmentation context. We propose to use a (per transcript) χ^2 goodness of fit test to compare the empirical distribution of read-length to the theoretical distribution under random fragmentation with rate $\hat{\kappa}$. To make the test more robust, we consider k bins of equal sizes and compute the statistic

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

Where O_j and E_j are respectively the observed and expected number of reads with length within window j. Under the random fragmentation hypothesis, χ^2 follows a chi-square distribution with $k-1$ degrees of freedom. The number k is adaptively chosen depending on coverage as $\max(10, \lfloor \text{Cov}/5 \rfloor)$, and the bin sizes then depend on transcript length. Adjustment for multiple testing is once again performed with Benjamini-Hochberg procedure.

Once more illustrating on the Rat Cerebellum samples, we identified that about 80% of the transcripts did actually show a degradation pattern corresponding to random fragmentation. Interestingly, while the estimated transcript length did not seem to affect the test, the transcripts rejected for degradation by random fragmentation were for the most part transcripts with high coverage (see Figure 2.18). We suspect that those transcripts might be more sensitive to our preprocessing steps, in particular in our definition of transcript length (redefined empirically for each transcript to avoid annotation errors) and full-length transcript (start position within a window of 3'end of the transcript). The left panel of Figure 2.19 illustrate an example were the fragmentation rate is under-estimated because of over-estimation of the proportion of full-length reads. On the other hand, our approach captures examples of transcripts with average coverage which most likely are not subject to random degradation. An example is given in the right panel of Figure 2.19 where there seems to be a preferential cleavage zone around nucleotide 1000 of the transcript.

2.2.8 Conclusion

The INDEGRA pipeline provides a model-based framework to estimate transcriptome-wide degradation in long-read sequencing data, and correct for degradation bias and compare biological degradation in different conditions under the assumption that degradation occurs by random fragmentation. It also provides a statistic to test this hypothesis, flagging transcripts that may exhibit specific patterns of degradation.

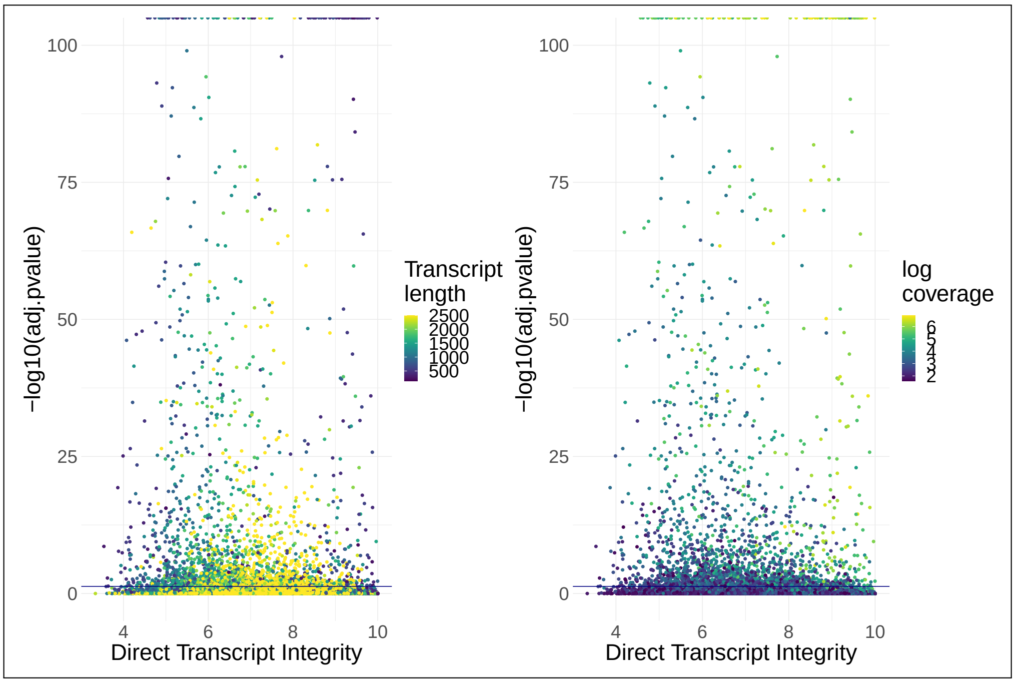

Figure 2.18: **Bias of random fragmentation test towards transcript coverage.** Adjusted p-values do not depend on DTI metrics, but rejected transcripts are biased towards high coverage, regardless of DTI. In each tested sample (here rat cerebellum low-RIN replicate), about 20% of transcripts are rejected for random fragmentation, more than half of them having coverage higher than 800 reads.

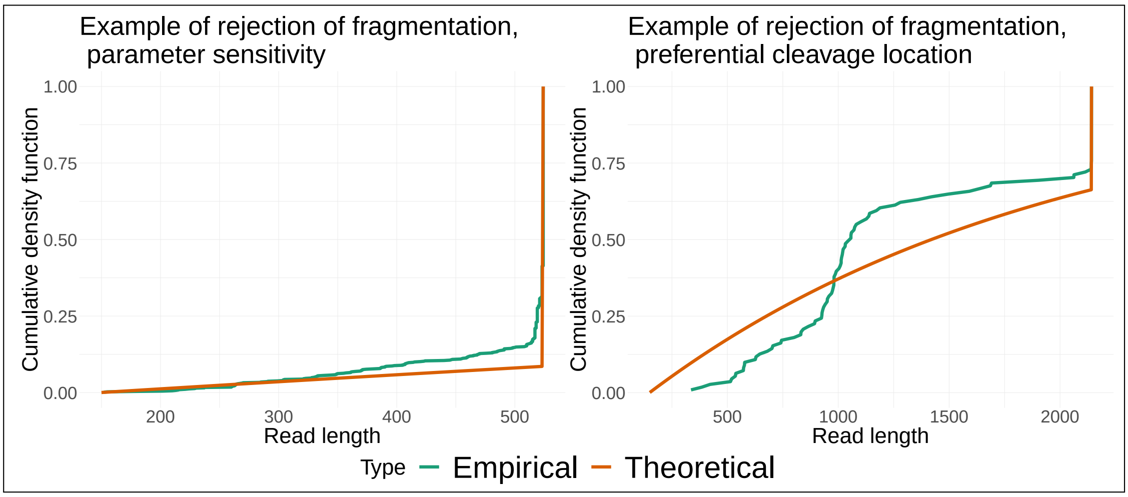

Figure 2.19: **Expected and empirical read-length distributions of two transcripts rejected for random fragmentation.** On the left panel, a transcript with 840 mapped reads for which random fragmentation was rejected due to degradation over-estimation. On the right panel, a transcript with 111 reads rejected for random fragmentation most likely because of preferential cleavage site around nucleotide 1000.

2.2.9 Ongoing work, extension

There are two major limitations to the current pipeline. The first is that it relies heavily on the assumption that only the last fragment of a transcript (containing the polyA tail) will be captured by the sequencing experiment. In data with in-vitro poly-adenylation (*i.e.* where a poly-A tail is added to all RNA fragments in the cell), this assumption does not hold. This issue may easily be resolved by our pre-processing step that retains only reads whose right end maps to the 3' end of the transcript. The second limitation is that the underlying model is not appropriate to study different types of degradation, and in particular 3' end degradation, as it relies on the distribution of the fragment length from the last fragmentation point to the 3'end of the transcript. In the context of a collaboration with two researchers who prefer not to be named (let's call them Alice and Bob), I will briefly present here alternatives that have not yet been fully validated in order to study 3' end degradation.

Spermatogenesis is the process by which haploid spermatozoa (final cells containing 1 version of each chromosome) develop from germ cells (containing 2 versions of each chromosome) in the testicle. This process is usually decomposed into several division steps and morphological changes, from which key stages are Juvenile Round Spermatids (RS_Juv), Round Spermatids (RS), Elongating Spermatids (eES), and finally Residual Bodies (RB), a subproduct of sperm fabrication that contains the remaining materials of the spermatogenesis process not necessary to spermatozoa.

Alice and Bob have collected in-vitro poly-adenylated DRS data from those four stages in a wild-type condition as well as in a knock-out of a specific protein (not specified here as this work is still unpublished). One of the hypothesis is that this protein is responsible for degradation of the residual material, and hence biological degradation is the particular focus of the study. Figure 2.20 for instance illustrates the estimation of sample DTI by Indegra in the 8 conditions.

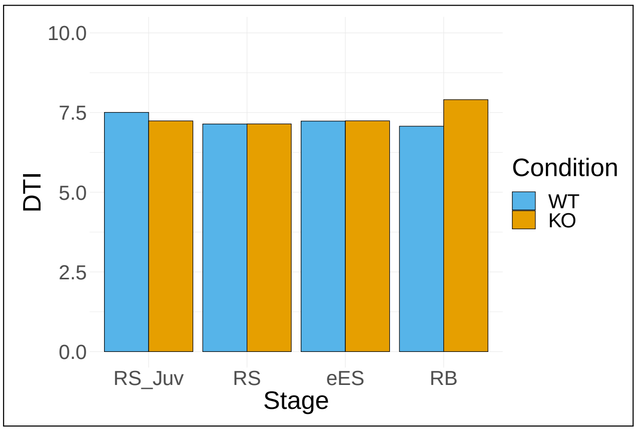

Figure 2.20: **Sample integrity evaluation in the four spermatogenesis stages, Wild-type and Knock-Out.**

In what follows, I have focused on a given transcript to illustrate the methodological developments introduced for this particular project. Once again, I will not disclose the name of the transcript for sake of confidentiality, as, though fascinating from a biological point of view, I believe it does not bring valuable insight to the methodological work presented here. The transcript will be named TR in what follows. Figure 2.21 illustrates the cumulative distribution of read-length mapping to TR, showing a priori no significant difference in degradation by fragmentation in the different stages and conditions.

To assess the presence of 3'end degradation between two given conditions ($j = 1, 2$), we have proposed to compute the empirical distribution of the 3'ends of each transcript t, denoted

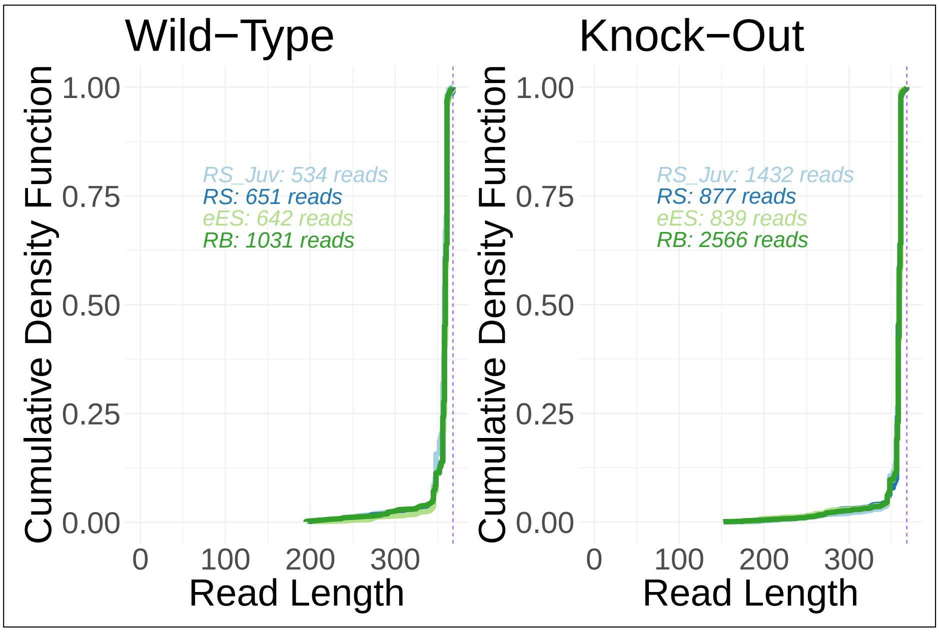

Figure 2.21: **Cumulative distribution of read length in all conditions for transcript TR.** Output from the INDEGRA pipeline, hence filtering reads based on their 3'end position.

Z_{tj} as follows:

$$\mathbb{P}(Z_{tj} = k) = \frac{\sum_{i=1}^{n_{tj}} \mathbb{1}_{W_{tji}=k}}{n_{tj}}$$

where W_{tji} is the 3'end position of the ith read from transcript t in condition j, and n_{tj} is the total number of reads mapping to transcript t in condition j. An illustration for transcript TR is given in Figure 2.22.

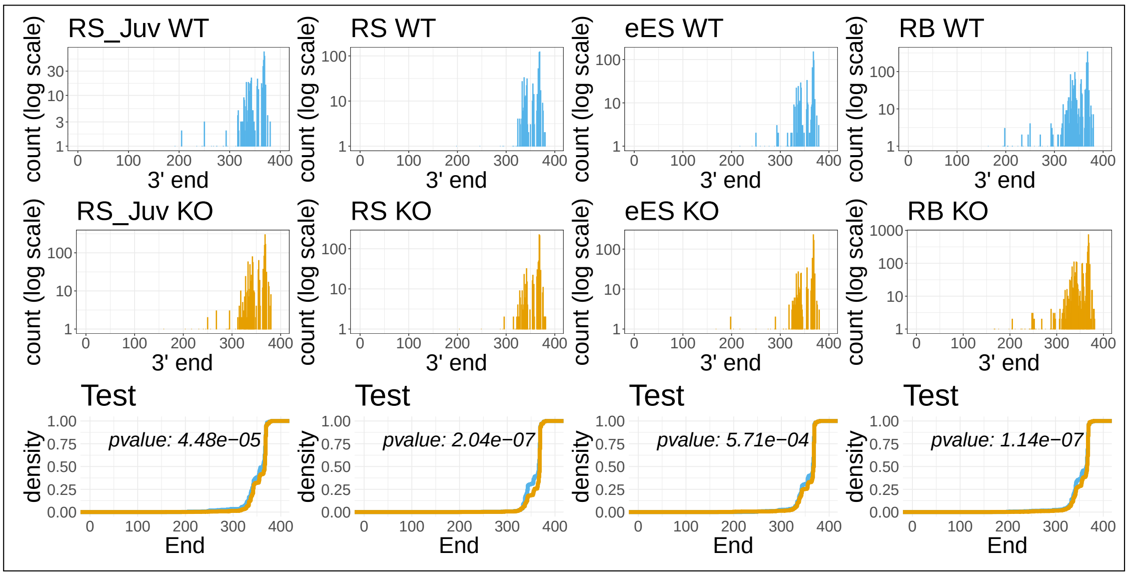

Figure 2.22: **Empirical distribution of transcript TR 3'end location** in the four different stages (columns) and the two different conditions (rows).

To test if the Wild-type condition is more degraded than the Knock-Out condition, we can then simply use a one-sided Kolmogorov-Smirnov test to compare the distributions of the 3' end positions of the transcripts. As illustrated on Figure 2.22, most of the mRNAs of interest in the study are significantly more degraded in the WT condition, and this degradation seems to occur mostly at the very early stage (RS_Juv) and at the latest stage (RB). Interestingly, Alice and Bob also performed an opposite experiment where the forced over-expression of their protein of interest (instead of knocking it down). Here we were able to show that the over-expressed condition is more degraded than the Wild-type (see for instance Figure 2.23), confirming the role of this protein in degradation.

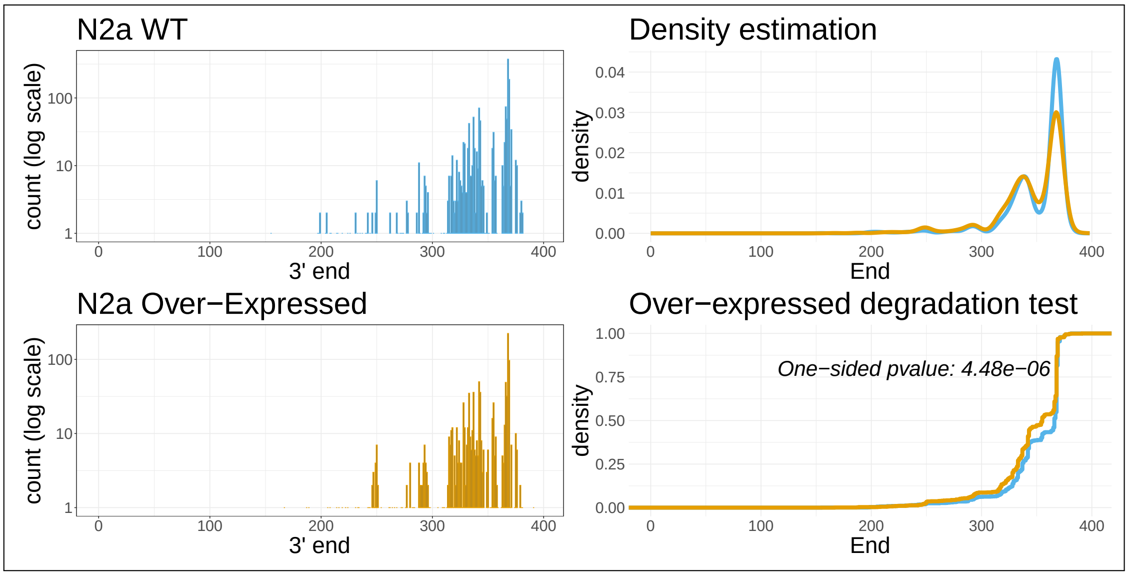

Figure 2.23: **Degradation test in the over-expressed experiment**

2.3 SWARM: Single-molecule Workflow for Analysing RNA Modifications

This work can be considered as still in progress. The methodological developments presented below have been proposed in strong collaboration with Stefan Prodic, PhD student at ANU, who has also performed extensive data processing and parallelisation of codes, and Akanksha Srivastava, postdoc at ANU, who has designed the neural networks used in the project. The project is led by Eduardo Eyras with the strong participation of Nikolay Shirokikh.

2.3.1 RNA modifications

Recent years have seen a remarkable expansion of our understanding of the chemical composition of messenger RNA (mRNA). In addition to the four canonical ribonucleotides, which compose the primary backbone of RNA, there are over 170 epitranscriptomic modifications that potentially provide an extra dimension of gene expression regulation (Martinez et al. 2022; H. Shi, J. Wei, and He 2019). Multiple research efforts have demonstrated the ability of modifications to dictate gene expression and the fate of mRNA through controlling structure, stability, splicing, translation efficiency, and subcellular localisation (Boo and Kim 2020). Critically, some of the RNA modifications are reversible, which provides an opportunity for regulation (Jia et al. 2011).

The evidence gathered thus far points to two main properties. One is that RNA modifications are highly dynamic, with modified mRNA sites changing across cellular conditions and disease states (Barbieri, Tzelepis, et al. 2017; Sas-Chen et al. 2020). The other is that the position of the modification appears to be relevant, with modification positions triggering specific mRNA processing outcomes, such as splicing (Martinez et al. 2022), translation (Meyer et al. 2015), and stability (Lee et al. 2020). Thus, the dynamic epitranscriptome is capable of mediating context-dependent changes in gene expression, underlying various biological processes such as animal development (Frye et al. 2018), sex determination (Haussmann et al. 2016), and cancer (Barbieri and Kouzarides 2020).

It is thus essential to develop methods that can identify modifications at nucleotide resolution and that are easily applicable across multiple conditions. Multiple experimental methods exist, including RNA immunoprecipitation (Koh, Y. T. Goh, and W. S. Goh 2019), protocols involving chemical conversion of modifications (Johnson et al. 2022), or specific signatures in reverse transcription (Werner et al. 2020). However, they usually detect one single modification at a time and involve complex protocols. Furthermore, these protocols only exist for a bunch of modifications and are not always easy to extend to new ones (Linder and Jaffrey 2019).

Direct RNA Sequencing offers a new opportunity to identify multiple modifications without resorting to chemical conversion. As mentioned before, DRS does not rely on PCR, sequencing the native RNA directly, hence preserving its chemical modifications. Moreover, the ionic current perturbation through the pores is directly related to the chemical composition of the sequence transiting through the pores. We therefore hypothesise that RNA modifications, such as methylation, should yield small perturbation of the expected currents that could be recovered by a deep-learning algorithm. In this Section, I will describe the approach we developed in SWARM (Single-molecule Workflow for Analysing RNA Modifications), a framework to enable the simultaneous detection of multiple RNA modifications at the transcriptome-wide level from the same sample. SWARM currently predicts m6A, m5C, ac4C, and Ψ at the individual read level as well as the transcriptome-site level. For sake of simplicity, all the results presented here are for the m6A modifications. Performance slightly varies between modifications, but allows to draw similar conclusions.

2.3.2 Data

Deep Neural Networks require to be trained on a set of positive and negative labels. To this aim, the group of Nikolay Shirokikh generated several sets of synthetic In-Vitro Transcripts (IVTs) where all representatives of a given nucleotide would be replaced by its modified version (e.g. replacing every A with m6A for the m6A model). In vitro transcription is a simple procedure that allows for template-directed synthesis of RNA molecules of any sequence from short oligonucleotides to those of several kilobases, and in large quantities (Beckert and Masquida 2011). In our context, we used two sets of templates, namely IVT-I and IVT-M, each containing 4 synthetic sequences of approximately 2000 nucleotides. IVT-I was designed by H. Liu, Begik, and Novoa 2021 to contain all possible 5mers and optimise the diversity of the 9mers in the sequences. IVT-M were directly selected from the mouse rRNA reference genome. For each template, 5 datasets were sequenced: one containing only pure nucleotides, one containing m5C in place of all Cs, one containing ac4C in place of all Cs, one containing m6A in place of all As and one containing Ψ in place of all Us.

At any time in the sequencing, a pore records a current corresponding to 5 nucleotides. The motor protein pulls the transcript slowly through the pore, so that the current of each 5mer of the sequence is sequentially recorded. Once the whole read is sequenced, basecaller algorithm typically use segmentation algorithms to break the signal into pieces corresponding to those 5mers. The consequence for us is that the impact of a modification on a specific nucleotide should be recorded in 5 successive 5mers, hence in a 9-mer signal (see Figure 2.24).

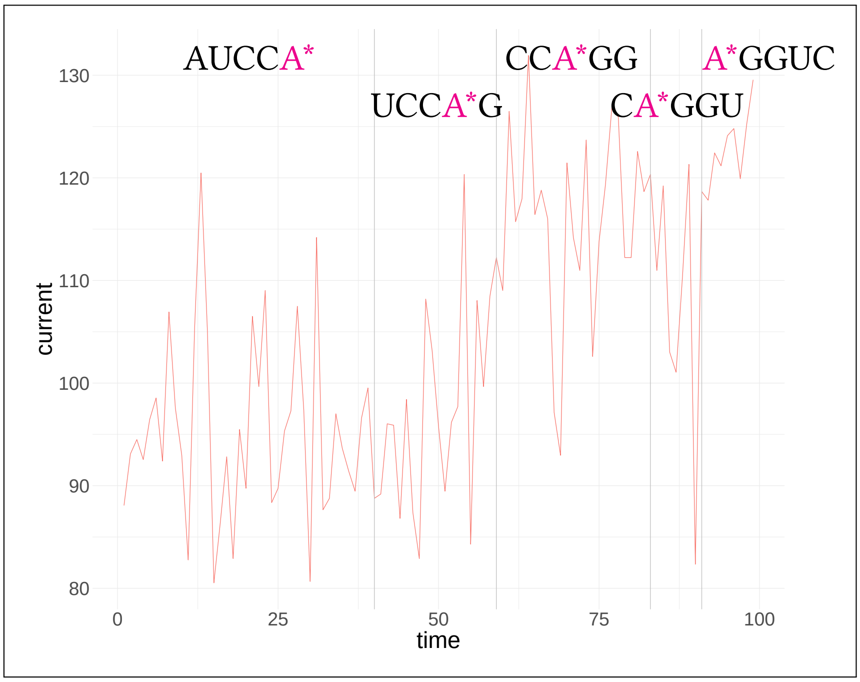

Figure 2.24: Example of raw signal output from the pore. Each segment corresponds to a 5mer in the sequence

For each modification (e.g. m6A), we therefore extracted all 9mers centered around the target nucleotide (e.g. A) from the IVT reference sequence, and for each of those 9mers extracted all signals from each of the sequencing data. We defined the labels as in the example of Figure 2.25: signals extracted from the non-modified IVTs were called *non-modified*; signals extracted from the target IVTs were called *modified*; and signals extracted from any other IVTs containing modifications were called *negative*.

AUCCAGGUC	⟶	non-modified
A*UCCA*GGUC	⟶	modified
AΨCCAGGΨC	⟶	negative
AUC*C*AGGUC*	⟶	negative
AUC*C*AGGUC*	⟶	negative

Figure 2.25: Example of signal labeling for 9mer AUCCAGGUC when targeting modification m6A

2.3.3 RNA modification detection at the read-level

Features

Neural networks require all training data to have the same dimension. However, as illustrated in Figure 2.24, some 5mers may transit faster than others through the pore, resulting in recorded currents of different lengths. We call this length the *dwelling time* of a 5mer within a read. (Note that a 5mer may have different dwelling times in different reads, as well as in different sequence contexts...) The first step of our approach is therefore to smooth ionic current events for each 5mer into current vectors of length 36 using linear interpolation. From these smoothed currents, we then computed the (signed) distance to the expected value of an unmodified 9mer where the central nucleotide is replaced by each possible (non-modified) nucleotide. Those expected values can either be taken from the ONT reference, which provides one value per 5mer (corresponding to the illustration in Figure 2.26), or computed from observed values on a fully unmodified independent IVT (in which case we compute an average smoothed signal, hence 36 different values). This creates 4 features of length 5×36 which we name distance to A, C, G or U. Finally, we also extract the per-base quality scores of the alignment to the reference (9 different values for a 9mer), and transform it into $180 = 9 \times 20$ feature by repeating each quality score 20 times.

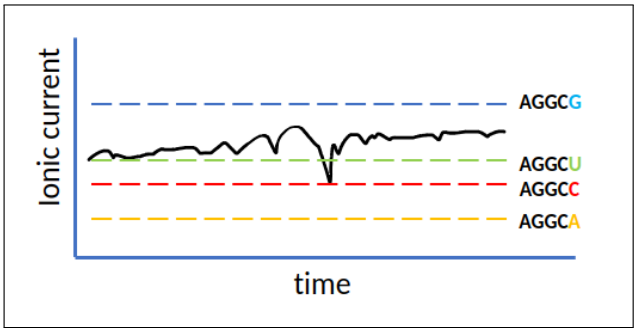

Figure 2.26: Example of smoothed raw signal from 5mer AGGCA*, and how distances to expected 5mers signals are computed. The dashed lines represent the reference current values for each 5mer starting with AGGC. Figure from Stefan Prodic

Hence our input data is a set of 7 features of size 180 corresponding to a 9mer that are summarized as follows:

- smoothed ionic current: 5 5mers smoothed into 36 values → 5×36
- distance to A: 5 smoothed 5mers currents compared to a reference → 5×36
- distance to C: 5 smoothed 5mers currents compared to a reference → 5×36

- distance to G: 5 smoothed 5mers currents compared to a reference → 5×36
- distance to U: 5 smoothed 5mers currents compared to a reference → 5×36
- dwelling time: 1 value per 5mers repeated 36 times → 5×36
- quality score: 1 value per nucleotide repeated 20 times → 9×20

Network architecture

The first task of Swarm is to decide if a current signal observed from a 9mer corresponds to a modified centered nucleotide or not. To this end, we tested several network architectures, all based on convolutional networks, and I present here the network that provided the best results on cell-line testing data (see below). This architecture was fully designed by Akanksha Srivastava. Briefly we used a 3-layered CNN with two dense layers at the end. More specifically, the network consists of

- Convolutional Layers that have successively 32, 64 and 128 filters with a kernel size of 3 and use ReLU activation;
- After each convulutional layer there is a max-pooling layer with a pool size of 2, and the output is flattened into a 1D tensor;
- The first dense layer has 128 units with ReLU activation.
- The second dense layer has 2 units with sigmoid activation, which is suitable for binary classification tasks.

For training, we used a balanced set of positive and negative labels, where both sets were composed of 500 signal representations of each 9mer with centered targeted nucleotide from the IVT-I data. The positive labels were composed of signals coming from the targeted modification IVT set only. The negative labels were composed of signals coming, for 70% of them, from the non-modified IVT set, and for 10% from each of the other modification IVT sets (for instance, for the m6A model, the negatives were composed of 70% non-modified, 10% ac4C, 10% m5C and 10% Ψ). This mixture of negatives allowed to drastically reduce the false positive rate on other modification data (see below).

Performance

This model achieves outstanding performances, both on signal generalisation (*i.e.* testing on data from the same IVT set, but on signals not used for training) and on 9mer generalisation (*i.e.* testing on data from the other IVT set, which contains 9mers not present in the training IVT set), as shown in Figure 2.27.

Those tests are performed on signals containing either no modification at all, or on signals containing all targeted nucleotide as modified. To ensure that our algorithms are modification-specific, we also tested their ability to distinguish the targeted modification from other modifications that might also alter the signal. We therefore applied our algorithms on the *negative* sets (as defined in Figure 2.25) and once again obtained a very clear separation of the target IVTs, as shown in Figure 2.28 (figures generated with the help of Stefan Prodic).

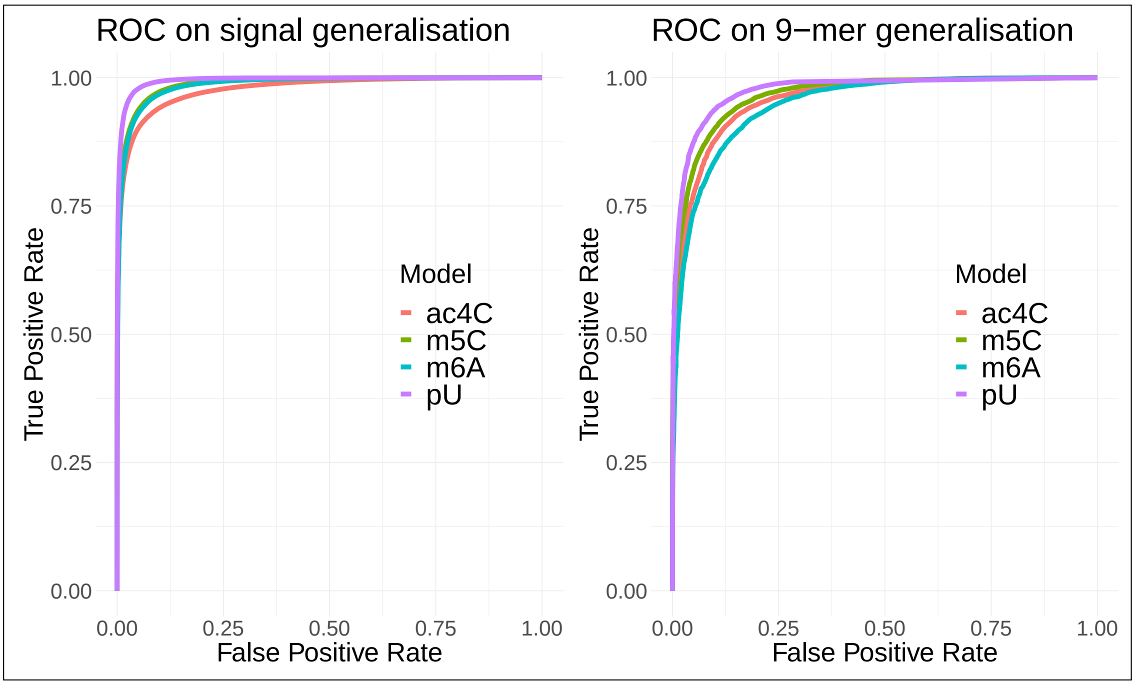

Figure 2.27: **ROC curves from our read-level prediction model on signal and 9mer generalisation.** Signal generalisation corresponds to testing on data from the same IVT set, but on signals not used for training) while 9mer generalisation corresponds to testing on data from the other IVT set, which contains 9mers not present in the training IVT set.

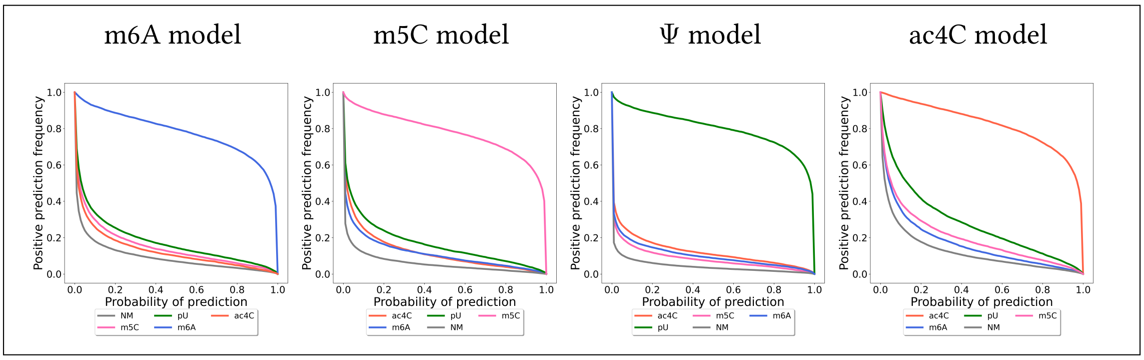

Figure 2.28: **Modification specificity.** Performance of our read-level algorithms in separating the targeted modification from all other modifications. For each target modification, we ran our read-level algorithm on all 5 orthogonal IVT datasets (Non-modified, m6A, m5C, Ψ and ac4C) and evaluated the recall as a function of the model threshold.

2.3.4 RNA modification detection at the site-level

The first part of SWARM hence predicts modification at the molecule-level, each read directly corresponding to a transcript molecule in the cell. To obtain a clean classification at the site level (*i.e.* looking at all molecules of a given transcript in a single picture), we propose to use a second neural network to predict the modification state, and provide a stoichiometry evaluation of the modification (not all molecules of a given transcript are necessarily modified, and this stoichiometry may be particularly important to evaluate the role of the RNA in the cell).

Features and network

Here again, we tested several architectures, and are currently investigating switching entirely the machine learning tool, using for instance for random forests or non-convolutional networks. However for simplicity, I will describe the approach using the same network architecture for our site-level and our read-level models. The input of the second model is a transformation of the output of the first model. More specifically, for each site, we extract all reads covering the RNA position and run the read-level prediction model. This results in a vector of n_S probabilities (n_S being the number of reads covering site S). To leverage the issue of fixed feature size, we then transform this vector into a histogram density estimation with window $h = 0.01$, resulting in a vector of length 100. The overall pipeline is illustrated in Figure 2.29
For training, we once again used a balanced set of positive and negative labels, where both sets were composed of 40 iterations of the following procedure:

- for each 9mer, select coverage C from a uniform distribution $\mathcal{U}([10, 1000])$ meant to correspond to the number of reads covering the site.
- for each of the labels,
    - if label is negative, then randomly select $0.9C$ signals from the unmodified IVT-I training sets and $0.1C$ signals from the modified training set (in both cases we used signals that were not used to train the read-level model) to artificially create noise in the negative labels. This prevents over-calling modifications.
    - if label is positive, select stoichiometry p from a uniform distribution $\mathcal{U}([0.1; 1])$ and then select $(1-p)C$ signals from the unmodified IVT-I training sets and $p * C$ signals from the modified training set.
- Apply read-level model to each signal and create histogram density estimator

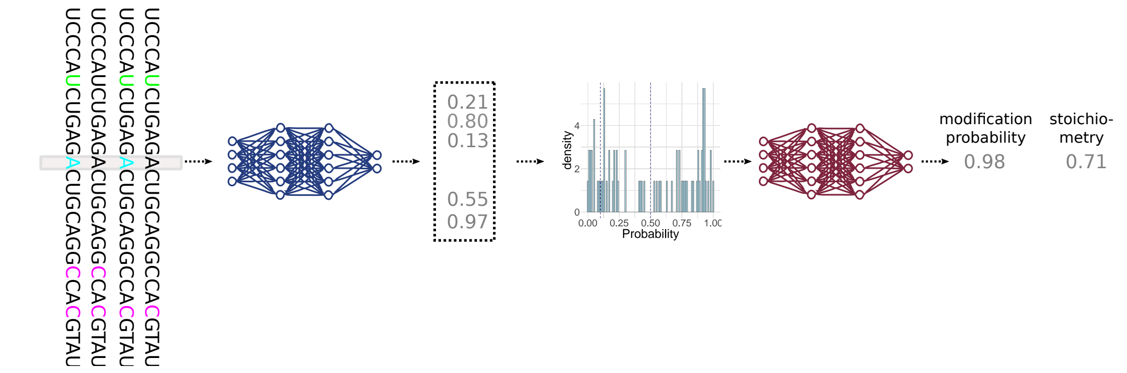
UCCCAUCUGAGACUGCAGGCCACGTAU
UCCCAUCUGAGACUGCAGGCCACGTAU
UCCCAUCUGAGACUGCAGGCCACGTAU
UCCCAUCUGAGACUGCAGGCCACGTAU
0.21
0.80
0.13
0.55
0.97
density
0
2
4
0.00
0.25
0.50
0.75
1.00
Probability
modification probability
0.98
stoichio-metry
0.71

Figure 2.29: **Site-prediction pipeline.** Each read covering the site of interest is processed through the read-level model, yielding a per-read modification probability. This vector is transformed into a histogram density estimator as input to the site-level prediction algorithm. Stoichiometry is then predicted as a ratio of positive and negative outputs from the read-level model, excluding reads with low-confidence probabilities.

The CNN is only used to predict the modification status of a given site, but it does not predict stoichiometry. To estimate the latter, we simply compute the proportion of reads predicted as modified by the read-level model. Because this model predicts unmodified reads with much higher confidence than modified ones, for each modification we identify thresholds that would discard the 20% of the data the worst predicted, *i.e* the thresholds s_1 predicting 20% of modified signals with probability below s_1, and the threshold s_2 predicting 20% of non-modified data with probability above s_2, as illustrated in the left panel of Figure 2.30. To predict stoichiometry on real data, we then simply compute the ratio of number of predictions with probability above s_2 and below s_1. This choice thus preserves the overall stoichiometry in the sample. Note that those reads with probability between s_1 and s_2 are used to predict the overall modification status of a site in our site-level model.

Performance

The right panel of Figure 2.30 illustrates the performance of our algorithm on synthetic testing, where we created artificial testing sites with controlled stoichiometry from the IVT-M dataset. For each modification, we achieve outstanding performance both at the precision/recall level and at the stoichiometry prediction.

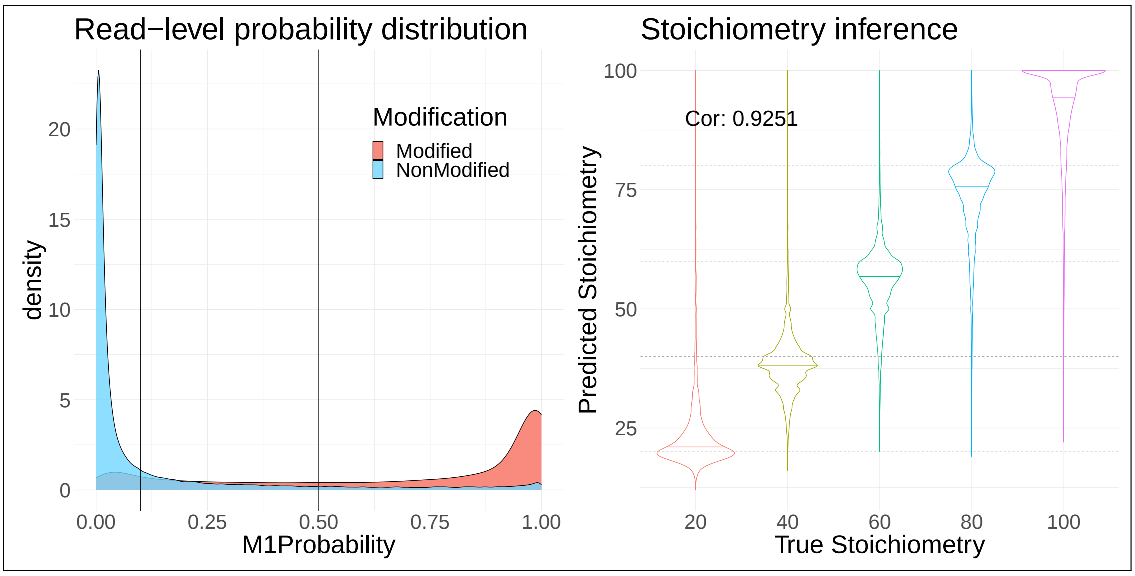

Figure 2.30: **Distribution of read-level probabilities on controlled test data and stoichiometry prediction performance** Left: The model predicts unmodified sites with stronger confidence than modified ones. Discarding reads with probability between 0.1 and 0.5 removes roughly 20% of modified and 20% of unmodified reads, hence preserving the stoichiometry in the sample. Right: our algorithm performs outstanding stoichiometry inference on synthetic testing data with controlled stoichiometry.

Our main test was however performed on real data from cell-lines with modification position and stoichiometry validated by chemical techniques. For instance, for m6A, methylated positions on the Hek293 cell-line were characterised by GLORI (W. Shen et al. 2024), a technique that removes all A nucleotides not methylated while leaving m6A intact. We therefore applied our whole pipeline to native Hek293 data and compared our predicted positions to the GLORI validated sites. We automated a training/testing pipeline allowing us to benchmark all choices of architectures, features, training data, etc against the best results on real-data. The pipeline (about 1000 lines of codes) runs in 48 hours, highly parallelized on a Australian National Computational Infrastructure in Canberra. Figure 2.31 illustrates the precision of our algorithm at different prediction thresholds. The blue circles indicate the results for different false positive rate values computed on independent cell-line IVTs.

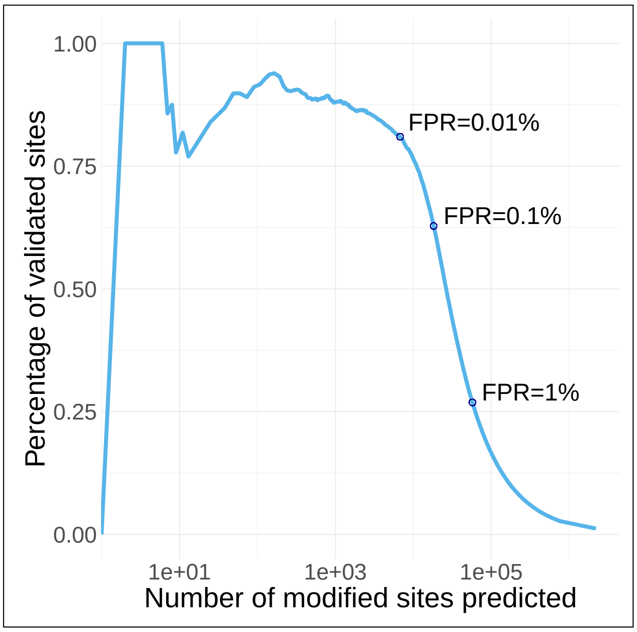

Figure 2.31: **Precision of SWARM on Hek293 cell-line at different prediction thresholds.** Precision is computed in terms of sites validated by GLORI. Results at different false positive rate values (computed on independent cell-line IVTs) are indicated by the blue circles.

2.3.5 Testing for stoichiometry differences in two or more conditions

Finally, we provide a procedure to test for differences in stoichiometry between different conditions. This procedure is based on a simple per-site GLM model, where for each condition j we denote p_j the it's stoichiometry and c_j it's coverage (*i.e.* the number of reads covering the site). We then construct

$$Y_{ij} = \begin{cases} 1 & \text{if } i < p_j c_j \\ 0 & \text{otherwise} \end{cases}$$

and consider the models

$$\begin{array}{lll} (\mathcal{M}1) & Y_{ij} \sim \mathcal{B}(p_j) & \text{logit} p_j = \alpha \\ (\mathcal{M}2) & Y_{ij} \sim \mathcal{B}(p_j) & \text{logit} p_j = \alpha_j \end{array}$$

and test $H_0 : \forall j, \alpha_j = \alpha$ vs $H_1 : \exists j, \alpha_j \neq \alpha_{j'}$ by a likelihood ratio test of models $(\mathcal{M}1)$ vs $(\mathcal{M}2)$. We then correct for multiple testing by using Benjamini and Hochberg's procedure (Benjamini and Hochberg 1995).

We for instance applied the whole pipeline to the comparison of a cancer cell-line, Hek293, in its natural (Wild-type) state and in a knock-out (KO) version where the main m6A writer, METTL3, was knocked out. METTL3 is known to target *DRACH* motifs, where D = A, G or U, R = G or A, and H = A,U or C, *e.g.* motifs of the type AGACU. We therefore expected a strong decrease in m6A modifications in the KO samples, in particular in DRACH motifs. Figure 2.32 shows the volcano plot of testing H_0 in all sites visible (*i.e.* covered by at least 20 reads) in both WT and KO samples, and significant in at least one sample (*i.e.* identified as modified with the site-level model). Out of 4255 tested sites, 2288 have significantly lower stoichiometry in the KO sample. Moreover, as illustrated in Figure 2.33, decreasing the p-value threshold (top figure) or increasing the stoichiometry difference threshold (bottom figure) significantly increases the proportion of DRACH motifs in the significant sites.

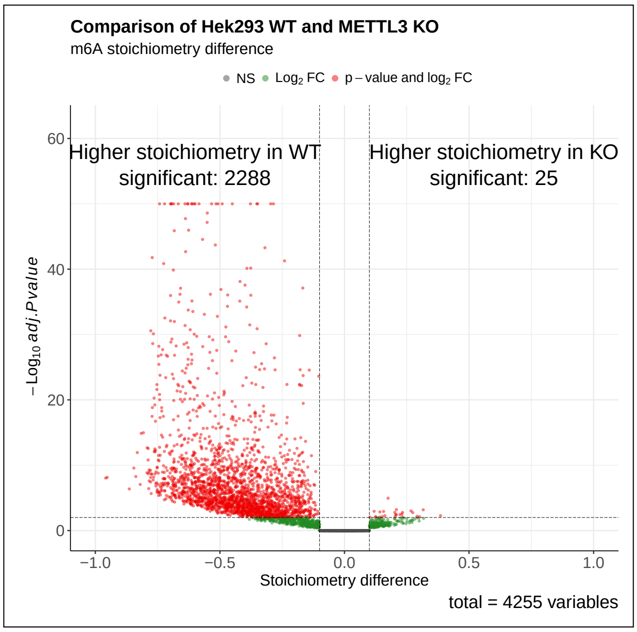

Figure 2.32: **Comparison of m6A stoichiometry in Wild-type and knock-out samples.** In the KO sample, the main m6A writer (METTL3) was knockout, resulting in a significantly decreased stoichiometry in almost all predicted modified sites.

2.3.6 Conclusion

SWARM provides the framework to identify four of the main RNA modifications directly from direct RNA sequencing without resorting to any chemical experiment. It shows excellent performances on the test we performed, in particular for m6A and pU. There are still a few tests we will want to perform before releasing the algorithm. For instance, the architecture of our site-level model is a convolutional neural network. This was chosen for convenience, allowing to use the same CNN for the read-level and site-level models. However, this choice has no other justification: the feature of this second model is a histogram density estimator with no structural property. Other machine-learning approaches, such as Random Forest, might perform better in this context.

Once the algorithm is released, we would like to understand the regulatory network between the RNA Modifying Proteins (RMPs) and the amount of detected modifications. Indeed, RNA modifications are naturally deposited by *writer* enzymes, recognised at specific regions of a transcript by *readers*, and can in some instances be removed by *erasers* enzymes. Collectively, writers, erasers and readers form RNA-modifying protein (RMP) complexes (Berdasco and Esteller 2022). Dysregulation of RMPs has been shown to have significant impacts on cancer development and progression (Esteve-Puig, Bueno-Costa, and Esteller 2020). Yet no clear model of expression of those RMPs has been provided to explain the differences of modifications in different tissues, organisms, or diseases. In a small project funded by *Campus France, FASIC* involving Stefan Prodic and myself, we plan on studying modifications and RMPs in a comprehensive nanopore DRS dataset that contains 6 tissues (Frontal cortex, Cerebellum, Hippocampus, Skeletal muscle, Liver, Testis) from multiple species (Human, Rat, Dog, Cow, Pig, Chicken), in order to decipher the regulatory networks of the main modifications, with a particular emphasis on pseudo-Uridine.

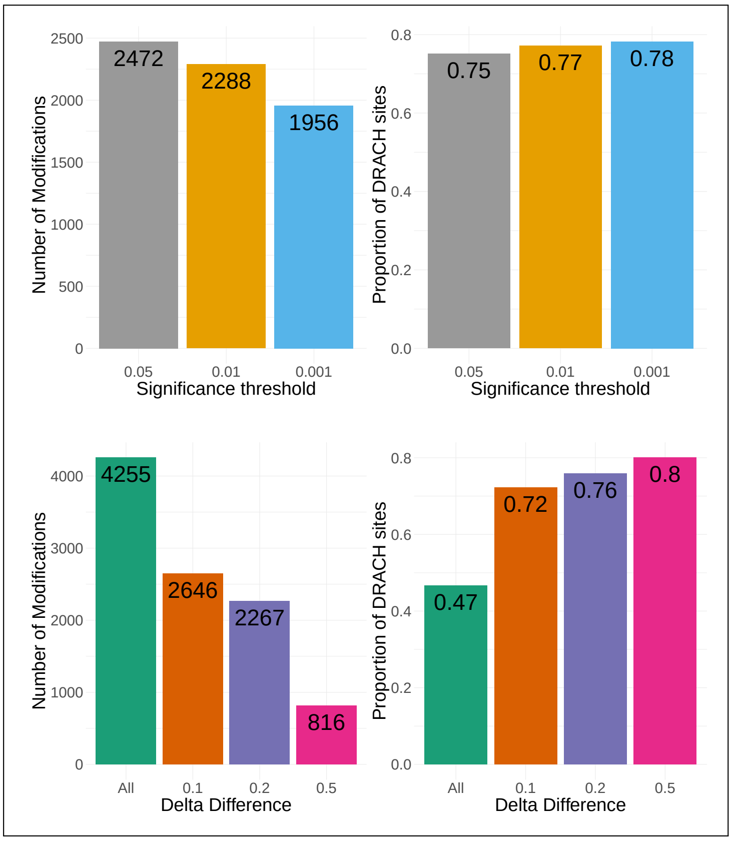

Figure 2.33: **Significant DRACH motif sites in the WT vs KO m6A modification comparison.** Decreasing the p-value threshold (top figure) or increasing the stoichiometry difference (bottom figure) threshold significantly increases the proportion of DRACH motifs in the significant sites.

Similarly, in the Multiple Myeloma project described at the end of this chapter, we will investigate the regulatory network of RMPs in cancer cells from patients.

2.4 DRS on Multiple Myeloma patient samples

This work is the pilot study of a large-scale project meant to paint the landscape of RNA modifications in Multiple Myeloma, and identify promising therapeutic targets in some of the sub-groups of patients with the worst prognosis. It has involved the participation of Agin Ravindran, who performed all the technical experiments of the project, and a collaboration with Dipti Talaulikar, MMBS at the Department of Cancer Biology at ANU and Eduardo Eyras and Nikolay Shirokikh, professors at JCSMR.

2.4.1 Pilot Study

The goal of this pilot study was mainly to evaluate the feasibility of DRS in patient samples, and if possible start drafting some analysis of the landscape of RNA modifications in 9 patient samples: 3 High-risk MM patients, 3 standard-risk MM patients, and 3 patients with Waldenström Macroglobulinemia, another type of cancer very similar to Myeloma and also producing high levels of monoclonal Ig proteins. The sequencing protocol is described in Figure 2.34. In a nutshell, each patient sample originates from a bone marrow aspirate, from which RNA was extracted. The resulting material was split so that each sample was sequenced with 2 technical replicates.

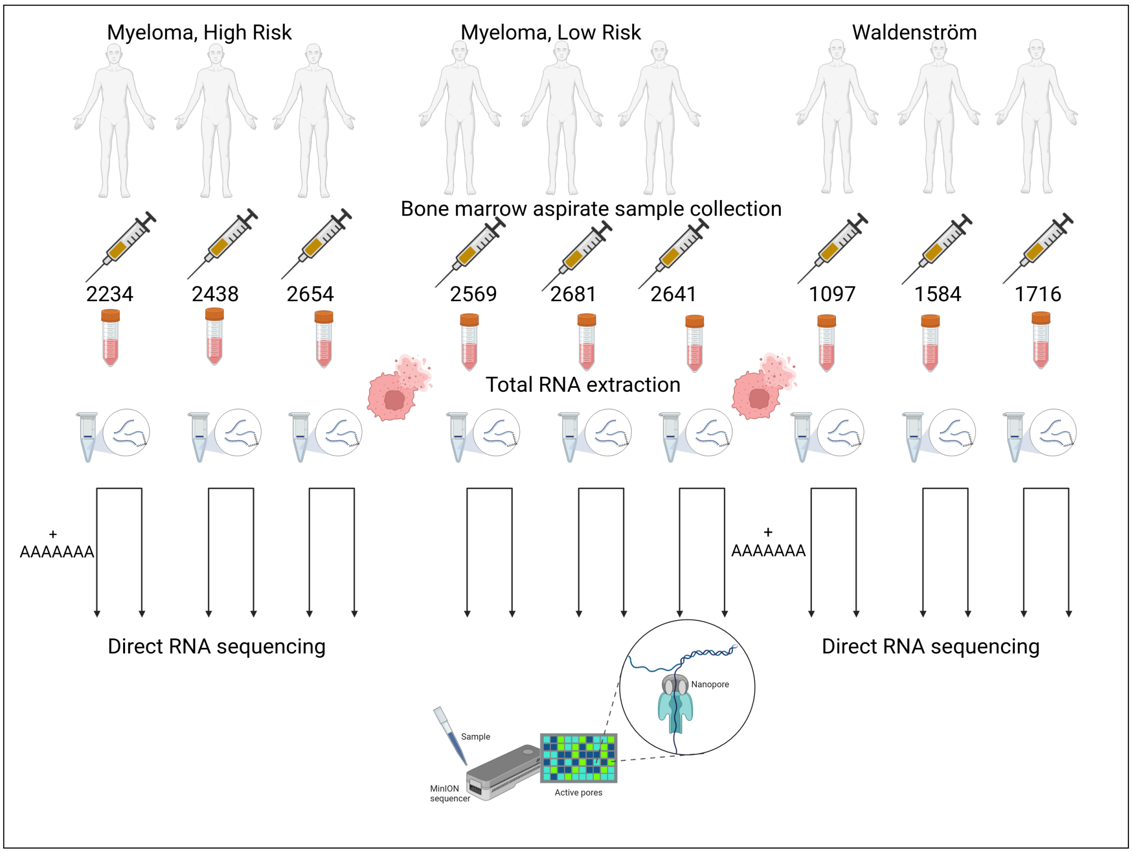

Figure 2.34: **Direct RNA Sequencing experiment on 9 patient samples** RNA was extracted from the bone-marrow aspirates, then separated in two technical replicates. For two patients, one technical replicate was additionnaly poly-adenylated in order to capture all species of RNA (mostly including ribosomal RNA, not naturally polyadenylated in its healthy state). All replicates were then sequenced using ONT DRS sequencing.

I will not describe here all the technical considerations and protocol steps that were discussed, but will simply summarise the lessons learnt: there are better ways to store the samples, extract the RNAs and perform the sequencing than what was done in this pilot study, and we are now ready to extend the project to larger cohorts. One interesting finding we did not expect is the representation of the different classes of RNA in the samples. Because MM and WM patients produce high levels of monoclonal antibodies, standard RNA-sequencing protocol tend to capture a large majority of reads mapping to the immunoglobulin genes (from 20% to 80% of the mapped reads). In our case, as shown in the left panel of Figure 2.35, the mitochondrial genes clearly dominated the sequencing runs, followed by Globin genes and finally Ig genes, the rest of the reads being more uniformly spread among the other mRNA genes. One solution we will use in our larger project is the addition of the RISER interface (Sneddon et al. 2022) which allows *in silico* depletion of MT and Globin genes during the sequencing run. This will allow us to sequence deeper the other mRNA transcripts and assess their modification landscape.

We applied our RNA integrity pipeline to the samples and obtained samples DTI ranging from 5 to 7.5 (see right panel of Figure 2.35), values corresponding to medium quality samples (as we expected from our quality checks). Importantly, the distribution of DTI did not depend on the patient group, which will facilitate downstream analysis such as differential transcript expression or modification landscape.

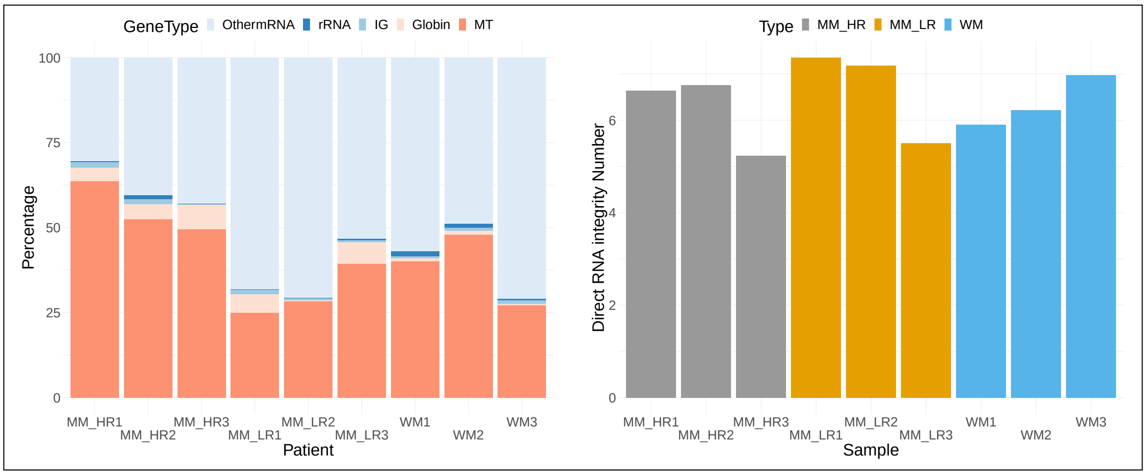

Figure 2.35: Left: RNA class representation in the 9 patient samples sequenced with DRS. Mitochondrial genes, together with globin genes, represent almost half the sequenced reads and should be depleted in further analysis. b. RNA integrity measured by Nanograd. All samples range between 5 and 7.5, corresponding to a reasonable quality that can be improved through a better RNA extraction protocol.

The limited number of patients and the sequencing depth did not allow us to produce significant results, but I will briefly summarize here some preliminary results we obtained on running SWARM to study post-transcriptional RNA modifications. One general belief is that the expression of modification writers is positively correlated with modification rates and abundance in the cells. Yet here a simple comparison of the high-risk and standard-risk patients showed precisely the opposite (see Figure 2.36). This suggests that the regulation network of writers, readers and erasers is vastly more complex than a general high expression / high rate scenario and that integrated analysis of regulators expression and modification abundance needs to be performed.

Modifications in ribosomal RNA has also been shown to be of major importance in the development and proliferation of cancer (see for instance Cerneckis et al. 2022; H. Peng et al. 2022;

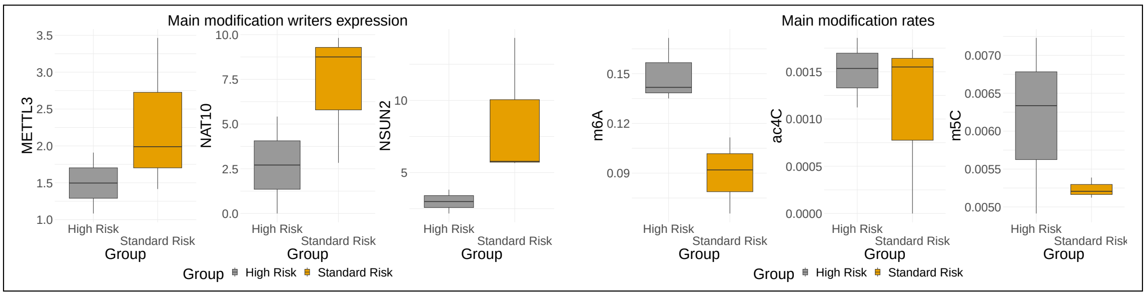

Figure 2.36: Left: Expression of the main modification writers: METTL3 for m6A, NAT10 for ac4C and NSUN2 for m5C (the main modification writers of pseudo-Uridine were not found expressed in the samples). Right: Modification rates in detected mRNA. The trend is opposite to the general expectation.

W. Zhang et al. 2023). We compared the high-risk and standard risk conditions as described in Section 2.3. Figure 2.37 represents the sites of siginificant different stoichiometries in both conditions, colored by modification type, focusing on 18S, the short sub-unit of the ribosomal complex. In particular, the cluster of ac4C modification around position 1750 (higher in standard-risk patients) corresponds to the known ac4C site in yeast. Studies such as Luo et al. 2023 have suggested that ac4C on rRNA is beneficial for protein translation efficiency. On the contrary, studies such as H. Peng et al. 2022 have shown that m6A deposition on ribosomal RNA promotes oncogenic transformation, which seems confirmed by its higher occurence in high-risk patients.

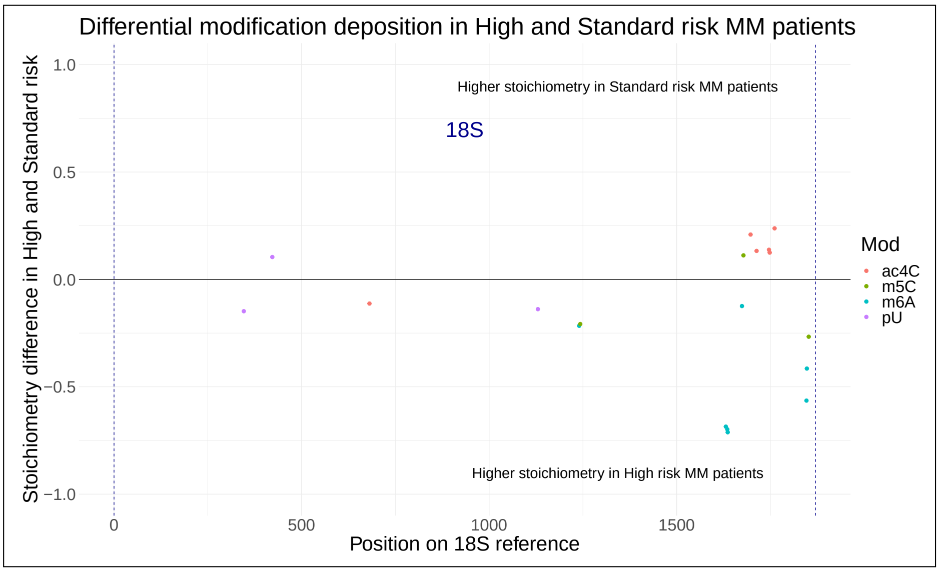

Figure 2.37: Sites of significant stoichiometry difference between High-risk and Standard-risk patients on the small sub-unit (18S) of the ribosomal complex, colored by modification type.

2.4.2 Future work

The results presented above were the pilot study of a large-scale project for which I have submitted a funding application to the International Myeloma Society. MM epitranscriptome represents an untapped opportunity to design refined sub-typing and identify new epitranscriptomic targets for sub-type-specific therapy. We hypothesize that RNA modifications represent very promising targets in MM and could reveal a breakthrough in improving the prognosis of most refractory sub-groups such as Del17p patients.

The first part of the project will focus on identifying the behaviors of modification depositers, erasers and readers in MM subgroups. We will investigate the expression of RNA modification regulators in a cohort of 363 patients from the IFM 2009 clinical trial for whom clinical information including most prevalent translocations (t(4;14), t(11;14), t(14;16)), deletions (13, 17p, 1p, 6q, 8p, 14, 16, 20p, 22q, X and Y) and amplifications (trisomies 5, 9 and 15, gain 1q, 6p and X) as well as survival follow-up (relapse and overall survival) for up to 12 years was recorded. We will investigate the expression pattern of all known modification regulators in multiple myeloma, and determine whether unsupervised clustering of patients and regulators can recapitulate the current classification of MM into main subgroups in parallel of defining main modification dysregulators associated to each group. We will then build modification-specific predictive models of survival that take into account the expression of all regulators to explain survival. Such models, trained on the IFM 2009 cohort will then be validated on independent datasets (for instance the MMRF CoMMpass cohort, see Skerget et al. 2021). Once the initial models are built, we will investigate each sub-group of patient separately in order to identify those that are most likely to respond to potential modification regulator targeted drugs.

The second part of the project will consist in using Direct RNA Sequencing on 50 newly diagnosed MM patients with our updated sequencing protocol. We will then use INDEGRA to study the degradation patterns in different groups of patients, and SWARM to study the modification abundances in different classes of RNA (ribosomal, messenger, etc). The main methodological contribution will be to define means of studying modifications (stoichiometries, locations, global abundances, etc) with respect to the regulatory network of the modification regulator proteins.

Finally, the last part of the project will consist in experimentally validating the functional impact of modifications and their regulators over MM phenotype and drug response, by knocking down the candidate regulator targets in cell-line and studying their behaviour (viability and proliferative capacity) with and without induced treatment stress. Here again, revealing a drug-specific epitranscriptomic function will be the main methodological challenge. The project proposal is summarised in the abstract Figure 2.38

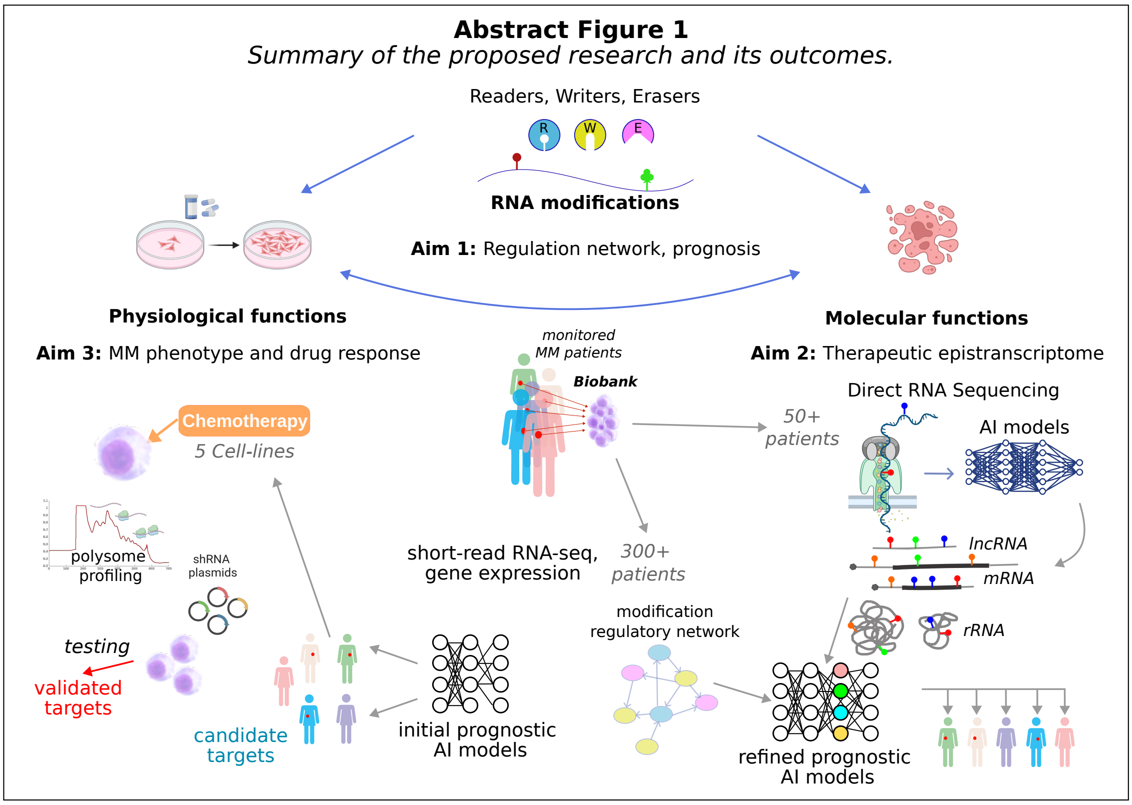

Figure 2.38: Overview of the project proposal: unraveling the therapeutic epitranscriptome of Multiple Myeloma.

References

Almudevar, Anthony (2001). “A Dynamic Programming Algorithm for the Optimal Control of Piecewise Deterministic Markov Processes”. In: *SIAM Journal on Control and Optimization* 40.2, pp. 525–539.

Attal, Michel et al. (2017). “Lenalidomide, bortezomib, and dexamethasone with transplantation for myeloma”. In: *New England Journal of Medicine* 376.14, pp. 1311–1320.

Azais, Romain and Florian Bouguet (2018). *Statistical Inference for Piecewise-deterministic Markov Processes*. John Wiley & Sons.

Barbieri, Isaia and Tony Kouzarides (2020). “Role of RNA modifications in cancer”. In: *Nature Reviews Cancer* 20.6, pp. 303–322.

Barbieri, Isaia, Konstantinos Tzelepis, et al. (2017). “Promoter-bound METTL3 maintains myeloid leukaemia by m6A-dependent translation control”. In: *Nature* 552.7683, pp. 126–131.

Baudrimont, Antoine et al. (2017). “Multiplexed gene control reveals rapid mRNA turnover”. In: *Science advances* 3.7, e1700006.

Bäuerle, N. and D. Lange (2018). “Optimal Control of Partially Observable Piecewise Deterministic Markov Processes”. In: *SIAM J. Control Optim.* 56.2, pp. 1441–1462.

Bäuerle, N. and U. Rieder (2011). *Markov decision processes with applications to finance*. Universitext. Springer, Heidelberg, pp. xvi+388.

Beckert, Bertrand and Benoit Masquida (2011). “Synthesis of RNA by in vitro transcription”. In: *RNA: Methods and protocols*, pp. 29–41.

Benjamini, Yoav and Yosef Hochberg (1995). “Controlling the false discovery rate: a practical and powerful approach to multiple testing”. In: *Journal of the Royal statistical society: series B (Methodological)* 57.1, pp. 289–300.

Berdasco, Maria and Manel Esteller (2022). “Towards a druggable epitranscriptome: Compounds that target RNA modifications in cancer”. In: *British journal of pharmacology* 179.12, pp. 2868–2889.

Boo, Sung Ho and Yoon Ki Kim (2020). “The emerging role of RNA modifications in the regulation of mRNA stability”. In: *Experimental & molecular medicine* 52.3, pp. 400–408.

Borchardt, Erin K, Nicole M Martinez, and Wendy V Gilbert (2020). “Regulation and function of RNA pseudouridylation in human cells”. In: *Annual review of genetics* 54, pp. 309–336.

Bouaziz, Olivier, Elodie Brunel, and Fabienne Comte (2019). “Nonparametric survival function estimation for data subject to interval censoring case 2”. In: *Journal of Nonparametric Statistics* 31.4, pp. 952–987.

Brandejsky, A., B. de Saporta, and F. Dufour (2013). “Optimal stopping for partially observed piecewise-deterministic Markov processes”. In: *Stochastic Process. Appl.* 123.8, pp. 3201–3238.

Bullard, James H et al. (2010). “Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments”. In: *BMC bioinformatics* 11.1, pp. 1–13.

Cerneckis, Jonas et al. (2022). “Decoding pseudouridine: an emerging target for therapeutic development”. In: *Trends in Pharmacological Sciences.*

Cleynen, Alice and Benoite de Saporta (2018). “Change-point detection for Piecewise Deterministic Markov Processes”. In: *Automatica* 97, pp. 234–247.

Cleynen, Alice and Benoite de Saporta (2023). "Numerical method to solve impulse control problems for partially observed piecewise deterministic Markov processes". In: *arXiv preprint arXiv:2112.09408.*

Cloez, Bertrand et al. (2017). "Probabilistic and piecewise deterministic models in biology". In: *ESAIM: Proceedings and Surveys* 60, pp. 225–245.

Cocozza-Thivent, Christiane (1997). *Processus stochastiques et fiabilité des systèmes.* Vol. 28. Mathématiques & Applications (Berlin) [Mathematics & Applications]. Berlin: Springer-Verlag.

Cooper, Geoffrey M and Kenneth W Adams (2023). *The cell: a molecular approach.* Oxford University Press.

Costa, O. and F. Dufour (2013). *Continuous average control of piecewise deterministic Markov processes.* SpringerBriefs in Mathematics. Springer, New York, pp. xii+116.

Davis, MHA. (1984). "Piecewise-deterministic Markov processes: a general class of nondiffusion stochastic models". In: *J. Roy. Statist. Soc. Ser. B* 46.3. With discussion, pp. 353–388.

— (1993). *Markov models and optimization.* Vol. 49. Monographs on Statistics and Applied Probability. Chapman & Hall, London, pp. xiv+295.

de Saporta, Benoite, Aymar Thierry d'Argenlieu, et al. (2024). "Medical follow-up optimization: A Monte-Carlo planning strategy". In: *arXiv preprint arXiv:2401.03972.*

de Saporta, Benoite, François Dufour, and Huilong Zhang (2015). *Numerical methods for simulation and optimization of piecewise deterministic Markov processes: application to reliability.* John Wiley & Sons.

Del Moral, Pierre (Mar. 1996). "Non Linear Filtering: Interacting Particle Solution". In: *Markov Processes and Related Fields* 2, pp. 555–580.

Esteve-Puig, Rosaura, Alberto Bueno-Costa, and Manel Esteller (2020). "Writers, readers and erasers of RNA modifications in cancer". In: *Cancer letters* 474, pp. 127–137.

Frye, Michaela et al. (2018). "RNA modifications modulate gene expression during development". In: *Science* 361.6409, pp. 1346–1349.

Gillespie, Daniel T (1977). "Exact stochastic simulation of coupled chemical reactions". In: *The journal of physical chemistry* 81.25, pp. 2340–2361.

Gu, Xinyu et al. (2023). "Vital roles of m5C RNA modification in cancer and immune cell biology". In: *Frontiers in Immunology* 14, p. 1207371.

Haussmann, Irmgard U et al. (2016). "m6A potentiates Sxl alternative pre-mRNA splicing for robust Drosophila sex determination". In: *Nature* 540.7632, pp. 301–304.

Helm, Mark and Juan D Alfonzo (2014). "Posttranscriptional RNA modifications: playing metabolic games in a cell's chemical Legoland". In: *Chemistry & biology* 21.2, pp. 174–185.

Hinderer, Karl and K Hinderer (1970). *Decision models.* Springer.

Houseley, Jonathan and David Tollervey (2009). "The many pathways of RNA degradation". In: *Cell* 136.4, pp. 763–776.

Ibrahim, Hend, Jeffrey Wilusz, and Carol J Wilusz (2008). "RNA recognition by 3'-to-5' exonucleases: the substrate perspective". In: *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms* 1779.4, pp. 256–265.

Jacobsen, Martin (2006). *Point Process Theory and Applications.* en. Probability and its Applications. Boston: Birkhäuser-Verlag. ISBN: 978-0-8176-4215-0. (Visited on 03/12/2024).

Jia, Guifang et al. (2011). "N 6-methyladenosine in nuclear RNA is a major substrate of the obesity-associated FTO". In: *Nature chemical biology* 7.12, pp. 885–887.

Jin, Gehui et al. (2020). "The processing, gene regulation, biological functions, and clinical relevance of N4-acetylcytidine on RNA: a systematic review". In: *Molecular Therapy-Nucleic Acids* 20, pp. 13–24.

Johnson, Zachary et al. (2022). "Systematic evaluation of parameters in RNA bisulfite sequencing data generation and analysis". In: *NAR Genomics and Bioinformatics* 4.2, lqac045.

Kaplan, Edward L and Paul Meier (1958). “Nonparametric estimation from incomplete observations”. In: *Journal of the American statistical association* 53.282, pp. 457–481.

Karikó, Katalin et al. (2008). “Incorporation of pseudouridine into mRNA yields superior nonimmunogenic vector with increased translational capacity and biological stability”. In: *Molecular therapy* 16.11, pp. 1833–1840.

Katt, Sammie, Frans A Oliehoek, and Christopher Amato (2017). “Learning in POMDPs with Monte Carlo tree search”. In: *International Conference on Machine Learning*. PMLR, pp. 1819–1827.

Koh, Casslynn WQ, Yeek Teck Goh, and WS Sho Goh (2019). “Atlas of quantitative single-base-resolution N 6-methyl-adenine methylomes”. In: *Nature communications* 10.1, p. 5636.

Krell, Nathalie and Emeline Schmisser (2021). “Nonparametric estimation of jump rates for a specific class of Piecewise Deterministic Markov Processes”. In: *Bernoulli* 27.4, pp. 2362–2388.

Lee, Yujin et al. (2020). “Molecular mechanisms driving mRNA degradation by m6A modification”. In: *Trends in Genetics* 36.3, pp. 177–188.

Lemaire, Vincent, Michèle Thieullen, and Nicolas Thomas (2020). “Thinning and multilevel Monte Carlo methods for piecewise deterministic (Markov) processes with an application to a stochastic Morris-Lecar model”. In: *Adv. in Appl. Probab.* 52.1, pp. 138–172.

Li, Shengbo Eben (2023). *Reinforcement learning for sequential decision and optimal control*. Springer.

Linder, Bastian and Samie R Jaffrey (2019). “Discovering and mapping the modified nucleotides that comprise the epitranscriptome of mRNA”. In: *Cold Spring Harbor Perspectives in Biology* 11.6, a032201.

Liu, Huanle, Oguzhan Begik, and Eva Maria Novoa (2021). “EpiNano: detection of m 6 A RNA modifications using oxford nanopore direct RNA sequencing”. In: *RNA Modifications: Methods and Protocols*, pp. 31–52.

Liu, Jianheng et al. (2022). “Developmental mRNA m5C landscape and regulatory innovations of massive m5C modification of maternal mRNAs in animals”. In: *Nature Communications* 13.1, p. 2484.

Love, Michael, Simon Anders, and Wolfgang Huber (2014). “Differential analysis of count data–the DESeq2 package”. In: *Genome Biol* 15.550, pp. 10–1186.

Luo, Jie et al. (2023). “Emerging role of RNA acetylation modification ac4C in diseases: current advences and future challenges”. In: *Biochemical Pharmacology*, p. 115628.

Lutz, Amelie M et al. (2008). “Cancer screening: a mathematical model relating secreted blood biomarker levels to tumor sizes”. In: *PLoS medicine* 5.8, e170.

Martin, James John (1967). “Bayesian decision problems and Markov chains”. In: *(No Title)*.

Martinez, Nicole M et al. (2022). “Pseudouridine synthases modify human pre-mRNA co-transcriptionally and affect pre-mRNA processing”. In: *Molecular Cell* 82.3, pp. 645–659.

Meyer, Kate D (2019). “m6A-mediated translation regulation”. In: *Biochimica et Biophysica Acta (Bba)-Gene Regulatory Mechanisms* 1862.3, pp. 301–309.

Meyer, Kate D et al. (2015). “5’ UTR m6A promotes cap-independent translation”. In: *Cell* 163.4, pp. 999–1010.

Mukherjee, Siddhartha (2010). *The emperor of all maladies: a biography of cancer*. Simon and Schuster.

National Cancer Institute, accessed April 2024 (2024). “What is cancer?” In: *Cancer.gov*.

Nicolò, Chiara et al. (2020). “Machine learning and mechanistic modeling for prediction of metastatic relapse in early-stage breast cancer”. In: *JCO clinical cancer informatics* 4, pp. 259–274.

Nombela, Paz, Borja Miguel-Lopez, and Sandra Blanco (2021). “The role of m6A, m5C and Ψ RNA modifications in cancer: Novel therapeutic opportunities”. In: *Molecular cancer* 20.1, pp. 1–30.

O’connell, Theodore X, Timothy J Horita, and Barsam Kasravi (2005). “Understanding and interpreting serum protein electrophoresis”. In: *American family physician* 71.1, pp. 105–112.

Peng, Hao et al. (2022). “N 6-methyladenosine (m6A) in 18S rRNA promotes fatty acid metabolism and oncogenic transformation”. In: *Nature Metabolism* 4.8, pp. 1041–1054.

Riedler, Martin G. (2013). “Almost sure convergence of numerical approximations for Piecewise Deterministic Markov Processes”. In: *Journal of Computational and Applied Mathematics* 239, pp. 50–71. ISSN: 0377-0427.

Risso, Davide et al. (2011). “GC-content normalization for RNA-Seq data”. In: *BMC bioinformatics* 12, pp. 1–17.

Rudnicki, Ryszard and Marta Tyran-Kamińska (2017). *Piecewise deterministic processes in biological models.* Vol. 1. Springer.

Saporta, B. de, F. Dufour, and H. Zhang (2016). *Numerical methods for simulation and optimization of piecewise deterministic Markov processes.* Mathematics and Statistics Series. Application to reliability. ISTE, London; John Wiley & Sons, Inc., Hoboken, NJ, pp. xiv+279.

Sas-Chen, Aldema et al. (2020). “Dynamic RNA acetylation revealed by quantitative cross-evolutionary mapping”. In: *Nature* 583.7817, pp. 638–643.

Schoenberg, Daniel R (2011). “Mechanisms of endonuclease-mediated mRNA decay”. In: *Wiley Interdisciplinary Reviews: RNA* 2.4, pp. 582–600.

Schroeder, Andreas et al. (2006). “The RIN: an RNA integrity number for assigning integrity values to RNA measurements”. In: *BMC molecular biology* 7.1, pp. 1–14.

Shen, Weiguo et al. (2024). “GLORI for absolute quantification of transcriptome-wide m6A at single-base resolution”. In: *Nature Protocols*, pp. 1–36.

Shi, Hailing, Jiangbo Wei, and Chuan He (2019). “Where, when, and how: context-dependent functions of RNA methylation writers, readers, and erasers”. In: *Molecular cell* 74.4, pp. 640–650.

Silver, David and Joel Veness (2010). “Monte-Carlo planning in large POMDPs”. In: *Advances in neural information processing systems* 23.

Skerget, S et al. (2021). “Genomic Basis of multiple myeloma subtypes from the MMRF CoMMpass study. medRxiv. 2021 2021.2008. 2002.21261211”. In: *DOI* 10.2021.08, pp. 02–21261211.

Sneddon, Alexandra et al. (2022). “Real-time biochemical-free targeted sequencing of RNA species with riser”. In: *bioRxiv*, pp. 2022–11.

Song, Hang et al. (2022). “Biological roles of RNA m5C modification and its implications in Cancer immunotherapy”. In: *Biomarker Research* 10.1, p. 15.

Sun, Ting, Ruiyan Wu, and Liang Ming (2019). “The role of m6A RNA methylation in cancer”. In: *Biomedicine & Pharmacotherapy* 112, p. 108613.

Sutton, Richard S and Andrew G Barto (2018). *Reinforcement learning: An introduction.* MIT press.

Veltz, Romain (2015). *A new twist for the simulation of hybrid systems using the true jump method.*

Wang, Shanshan et al. (2022). “Dynamic regulation and functions of mRNA m6A modification”. In: *Cancer cell international* 22.1, p. 48.

Werner, Stephan et al. (2020). “Machine learning of reverse transcription signatures of variegated polymerases allows mapping and discrimination of methylated purines in limited transcriptomes”. In: *Nucleic Acids Research* 48.7, pp. 3734–3746.

Wick, Ryan R, Louise M Judd, and Kathryn E Holt (2019). “Performance of neural network basecalling tools for Oxford Nanopore sequencing”. In: *Genome biology* 20, pp. 1–10.

Willrich, Maria AV and Jerry A Katzmann (2016). “Laboratory testing requirements for diagnosis and follow-up of multiple myeloma and related plasma cell dyscrasias”. In: *Clinical Chemistry and Laboratory Medicine (CCLM)* 54.6, pp. 907–919.

Xu, Jiangping, Guillermo Vilanova, and Hector Gomez (2016). “A mathematical model coupling tumor growth and angiogenesis”. In: *PloS one* 11.2, e0149422.

Yang, Xin et al. (2017). “5-methylcytosine promotes mRNA export—NSUN2 as the methyltransferase and ALYREF as an m5C reader”. In: *Cell research* 27.5, pp. 606–625.

Zhang, Wenjie et al. (2023). “ac4C acetylation regulates mRNA stability and translation efficiency in osteosarcoma”. In: *Heliyon*.

Zhao, Wanqing et al. (2019). “PACES: prediction of N4-acetylcytidine (ac4C) modification sites in mRNA”. In: *Scientific reports* 9.1, p. 11112.