Multiscale analysis of transcriptome : methodological and algorithmic developments

Arnaud Liehrmann

université PARIS-SACLAY

THESE DE DOCTORAT

NNT : 2023UPASL111

Multiscale analysis of transcriptome : methodological and algorithmic developments

Analyse multi-échelle du transcriptome : développements méthodologiques et algorithmiques

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 577, Structure et Dynamique des Systèmes Vivants (SDSV)
Spécialité de doctorat : Biologie Computationnelle
Graduate School Life Sciences and Health
Référent : Université d'Évry Val d'Essonne

Thèse préparée dans les unités de recherche **Institute of Plant Sciences Paris-Saclay (IPS2)** (Université Paris-Saclay, CNRS, INRAE, Univ Évry) et **Laboratoire de Mathématiques et Modélisation d'Évry** (Université Paris-Saclay, CNRS, Univ Évry) sous la direction de **Guillem RIGAILL**, Directeur de recherche et la co-direction de **Benoît CASTANDET**, Maître de conférence

Thèse soutenue à Paris-Saclay, le 28 novembre 2023, par

Arnaud LIEHRMANN

Composition du jury

Membres du jury avec voix délibérative

Ingrid LAFONTAINE Professeure, Sorbonne Université	Présidente
Émilie LEBARBIER Professeure, Université Paris Nanterre	Rapporteure
Ian SMALL Professeur, University of Western Australia	Rapporteur
Ciarán CONDON Directeur de recherche, CNRS, Institut de Biologie Physico-Chimique	Examinateur
Hélène TOUZET Directrice de recherche, CNRS, Centre de Recherche en Informatique, Signal et Automatique de Lille	Examinatrice
Pierre NEUVIAL Directeur de recherche, CNRS, Institut de Mathématiques de Toulouse	Examinateur

université PARIS-SACLAY
ÉCOLE DOCTORALE
Structure et dynamique des systèmes vivants (SDSV)

Titre : Analyse multi-échelle du transcriptome : développements méthodologiques et algorithmiques

Mots clés : Transcriptomique, Analyse Différentielle, Méthodes Data-driven, Programmation Dynamique, Détection de Ruptures Multiples, Statistiques Computationnelles

Résumé : *Mon travail peut être divisé en deux parties principales. Premièrement, j'ai conçu des outils dédiés à l'analyse différentielle du transcriptome. Deuxièmement, j'ai développé et appliqué des méthodes de détection de ruptures sur des ensembles de données génomiques.*

La diversité remarquable des isoformes d'ARN est principalement attribuable à des modifications post-transcriptionnelles, en plus des sites alternatifs d'initiation de la transcription. Ces modifications couvrent un ensemble d'événements qui peuvent se produire le long des molécules d'ARN, comprenant l'épissage, la maturation des extrémités, la polyadénylation alternative, l'édition, et la modification de base azotée. L'avènement de la transcriptomique à haut débit a catalysé une compréhension sans précédent de cette diversité. Cependant, l'analyse de ces données présente des défis statistiques, informatiques, techniques et biologiques considérables.

J'ai activement contribué au développment de deux méthodes, DiffSegR et comaturationTrackeR, dédiées à l'analyse différentielle du transcriptome. Ces méthodes sont conçues pour atténuer les difficultés liées à l'étude des isoformes individuelles, souvent non annotées, en se concentrant plutôt sur des analyses événement par événement ou par paire d'événements. DiffSegR permet d'identifier les différences d'expression à l'échelle du transcriptome entre deux conditions biologiques à partir de données RNA-Seq. Grâce à l'intégration d'un algorithme de détection de ruptures multiples, il délimite avec précision les frontières des régions/événements différentiellement exprimés, éliminant ainsi la nécessité d'annotations préalables. D'autre part, comaturationTrackeR, qui utilise des données RNA-seq à lectures longues, est conçu pour détecter les co-maturations à l'échelle du transcriptome, c'est-à-dire les dépendances entre les paires d'événements de maturation tels que l'édition et l'épissage. Les deux méthodes sont intégrées au cadre statistique DESeq2. Cette intégration permet de tester rigoureusement les différences d'expression et les co-maturations. De plus, ces méthodes ont été intuitivement encapsulées dans des packages R, ce qui garantit leur convivialité tant pour les biologistes que pour les bioinformaticiens. En effet, la sortie de ces packages est conçue pour créer des pistes IGV (Integrated Genome Viewer) et/ou des objets Bioconductor. Ces approches ont été appliquées et ont prouvé leur efficacité sur le transcriptome du chloroplaste, de la mitochondrie et d'une bactérie. En outre, il est important de noter que de nombreux résultats ont été validés au niveau moléculaire. Cela inclut une liste publiée d'événements co-maturés dans le chloroplaste d'*Arabidopsis thaliana*, une liste d'extensions 3' et 5' de transcrits, ainsi que l'accumulation d'ARN antisens et d'introns dans deux mutants d'*A. thaliana* pour les ribonucléases du chloroplaste—Mini-III et PNPase. Elle inclut également des candidats potentiels à la dégradation directe par Rae1 dans *Bacillus subtilis*.

Une autre facette de ma thèse concerne le développement et l'application de méthodologies de détection de ruptures multiples sur des ensembles de données génomiques. La popularité de ces modèles en génomique provient de leur capacité inhérente à révéler des événements biologiques non annotés le long du génome, tels que les différences d'expression résultant de variations d'épissage (comme le montre l'exemple de DiffSegR). Divers algorithmes de programmation dynamique visant à maximiser une vraisemblance pénalisée ont été proposés. Ces algorithmes et les contrastes qu'ils optimisent présentent des propriétés informatiques et statistiques remarquables, leur rapidité justifiant leur utilisation avec des données génomiques. Dans cette lignée, j'ai conçu et mis en œuvre un algorithme de programmation dynamique exact et efficace, Ms.FPOP. Cet algorithme optimise un critère des moindres carrés et incorpore une pénalité multi-échelle, dont il a été démontré qu'elle possédait des propriétés statistiques supérieures au critère des moindres carrés pénalisé avec un critère d'information bayésien. Ms.FPOP utilise des techniques d'élagage fonctionnel pour accélérer le temps de calcul de quadratique (l'algorithme le plus rapide connu à ce jour) à en moyenne log-linéaire en la longueur du signal. Ms.FPOP est implémenté en C++ et est interfacé avec R pour un accès convivial. J'ai effectué des simulations approfondies de Ms.FPOP avec une grande variété de scénarios, et les résultats sont prometteurs. Parallèlement, j'ai appliqué des méthodes de détection de ruptures multiples à des ensembles de données génomiques et j'ai observé que ces méthodes amélioraient l'état de l'art pour la détection des régions différentiellement exprimées dans les données RNA-Seq et des pics dans les données ChIP-Seq.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Structure et dynamique des systèmes vivants (SDSV)

Title : Multiscale analysis of transcriptome : methodological and algorithmic developments

Keywords : Transcriptomics, Differential analysis, Data-driven Methods, Dynamic Programming, Multiple Changepoint Detection, Computational Statistics

Abstract : *My work can be divided into two main parts. First, I have designed tools dedicated to the differential analysis of the transcriptome. Second, I have developed and applied multiple changepoint detection methods for genomic datasets.*

The remarkable diversity of RNA isoforms, besides alternative transcription initiation sites, is primarily attributable to post-transcriptional modifications. These alterations span an array of events that can occur along RNA molecules including splicing, processing, alternative polyadenylation, editing, and base modification. The advent of high-throughput transcriptomics has catalyzed an unprecedented understanding of this diversity. However, the analysis of such data presents substantial statistical, computational, technical, and biological challenges.

I actively contributed to the development of two methods, DiffSegR and comaturationTrackeR, dedicated to the differential analysis of transcriptomes. These methods are built to alleviate the complications arising from studying, often unannotated, individual isoforms, focusing instead on event-by-event or pairwise analyses. DiffSegR empowers the identification of transcriptome-wide expression differences across two biological conditions using RNA-Seq data. With the integration of a multiple changepoint detection algorithm, it precisely delineates the boundaries of differentially expressed regions/events, eliminating the necessity for prior annotations. On the other hand, comaturationTrackeR, utilizing long-read RNA-seq data, is tailored for the detection of transcriptome-wide co-maturations—dependencies between pairs of maturation events such as editing and splicing. Crucially, both methods are integrated with the DESeq2 statistical framework. This inclusion allows for rigorous testing of expression differences and co-maturations. Furthermore, these methods have been intuitively encapsulated into R packages, ensuring user-friendliness for both biologists and bioinformaticians. The output from these packages is designed to create IGV (Integrated Genome Viewer) tracks and/or Bioconductor objects. These approaches have proven their effectiveness through practical applications on the transcriptomes of chloroplasts, mitochondria, and bacteria. Importantly, many of the findings have been validated molecularly. This includes a published list of co-matured events within the chloroplast of *Arabidopsis thaliana*, an comprehensive list of 3' and 5' termini extension of transcripts, as well as the accumulation of antisense RNA and introns from two *A. thaliana* mutants for chloroplast ribonucleases—Mini-III and PNPase. It also includes potential candidates for direct degradation by Rae1 in *Bacillus subtilis*.

Another facet of my thesis involves the development and application of multiple changepoint detection methodologies on genomic datasets. The popularity of these models in genomics stems from their inherent capability to reveal unannotated biological events along the genome, such as expression differences resulting from splicing variations (as exemplified in DiffSegR). Various dynamic programming algorithms aimed at maximizing a penalized likelihood have been proposed over the years. These algorithms and the contrasts they optimize display remarkable computational and statistical properties, with their speed performance being a rationale for their use with genomic data. Building upon this line of research, I have designed and implemented an exact and efficient dynamic programming algorithm, Ms.FPOP. This algorithm optimizes a least squares criterion and incorporates a multiscale penalty, which has been demonstrated to possess superior statistical properties compared to the standard least squares criterion with a bayesian information criterion. Ms.FPOP employs functional pruning techniques to accelerate the computation time from quadratic (the best-known algorithmic speed so far) to on average log-linear relative to the length of the signal. Ms.FPOP is implemented in C++ and is interfaced with R for user-friendly access. I have conducted extensive testing of Ms.FPOP across a wide variety of simulated scenarios, and the results have been promising. Concurrently, I have applied multiple changepoint detection algorithms to genomic datasets, and observed that these methods improve the current state-of-the-art methods for detecting differentially expressed regions in RNA-Seq data and peaks in ChIP-Seq data.

Remerciements

Tout d'abord, je tiens à remercier Émilie Lebarbier et Ian small d'avoir accepté de rapporter ma thèse, ainsi que Ciarán Condon, Hélène Touzet, Ingrid Lafontaine et Pierre Neuvial d'avoir accepté de faire partie de mon jury.

Je remercie ensuite mon directeur de thèse Guillem Rigaill et mon co-directeur Benoît Castandet. Je suis reconnaissant pour leur confiance, leur soutien, leur aide et leurs nombreux conseils précieux. Bienheureux les futurs étudiants qui auront la chance d'apprendre à vos côtés.

Mes remerciements s'étendent également à Hakim Mireau, Nathalie Vialaneix et Pierre Nicolas qui ont accepté de faire partie de mon comité de thèse. Je les remercie pour le temps qu'ils m'ont consacré. Je suis également reconnaissant envers Toby Dylan Hocking pour m'avoir fait découvrir la recherche au-delà des frontières françaises. Je remercie aussi Gaetano Romano, Vincent Runge et Charles Truong pour m'avoir donné l'opportunité de présenter mes travaux devant un auditoire international.

Je remercie Marie-Laure Martin, Étienne Delannoy et Christophe Ambroise pour m'avoir accueilli au sein de l'IPS2 et du LaMME, plus précisément dans leurs équipes respectives : GNet, OGE, et Stats et Génome. Grâce à vous, j'ai eu le privilège de travailler au carrefour de plusieurs disciplines : la biologie, la bioinformatique et les statistiques, aux côtés de chercheurs véritablement convaincus de l'utilité du dialogue interdisciplinaire. Vive l'interdisciplinarité !

Je tiens aussi à remercier tous ceux que j'ai eu l'occasion de côtoyer à l'IPS2, au LaMME, ainsi que lors des différents séminaires et conférences. J'ai beaucoup appris grâce à vous. Les moments partagés autour des repas conviviaux et les (nombreuses) pauses café resteront pour moi des instants précieux. J'ai une pensée toute spéciale pour mes collègues de bureau, Claire et Luda au LaMME, qui ont fait du bureau 403 un lieu toujours bien approvisionné (en chocolat surtout !). À l'IPS2, merci à Dario, Jeanne Marie et Sébastien, pour m'avoir réservé une place bien au chaud chaque hiver dans le bureau 1.47.

Je remercie toutes les personnes qui m'ont aidé à rédiger ce manuscrit et/ou ont relu ce dernier : Guillem, Benoît, Véronique, Chloé et Claire.

Je remercie la coloc' Saint-Marcel, un havre de paix qui a été mon refuge durant ces deux dernières années. Lorsque vous vous promenez le long du boulevard Saint-Marcel en soirée, prêtez l'oreille : vous pourrez peut-être saisir les mélodies d'un piano et la voix exceptionnelle de trois choristes, voire les harmonies d'une basse, seulement si la chance vous sourit. Gardez les yeux ouverts et, sur l'une des terrasses, vous pourriez apercevoir quelques morceaux de mimolette extra-vieille (détail non négligeable) lors des soirées estivales. Je remercie en particulier Chloé pour son soutien au quotidien et sa patience ces derniers mois.

Enfin, je remercie mes amis et ma famille pour leur soutien indéfectible, tant avant que durant toute la période de cette thèse. Les mots me manquent pour traduire toute l'étendue de ma reconnaissance envers vous.

Table of Contents

List of Figures

List of Abbreviations

APA Alternative Polyadenylation

ATI Alternative Transcription Initiation

AS Alternative Splicing

BIC Bayesian Information Criterion

cDNA complementary DNA

ChIP-Seq Chromatin Immunoprecipitation followed by Sequencing

DERs Differentially Expressed Regions

DNA Deoxyribonucleic Acid

FDP False Discovery Proportion

FDR False Discovery Rate

FPOP Functional Pruning Optimal Partitioning

GFPOP General Functional Pruning Optimal Partitioning

GLM Generalized Linear Model

HMM Hidden Markov Model

H3K36me3 trimethylation at the 36^{th} lysine residue of histone H3

H3K4me3 trimethylation at the 4^{th} lysine residue of histone H3

IGV Integrative Genomics Viewe

LSC Least Squares Criterion

Ms.FPOP Multiscale Functional Pruning Optimal Partitioning

nt nucleotides

OP Optimal Partitioning

PAS Polyadenylation Sites

PCR Polymerase Chain Reaction

PPR Pentatricopeptide Repeat

PRDS Positive Regression Dependency on a Subset

RBPs RNA-Binding Proteins

RNA Ribonucleic Acid

RIP-Seq RNA Immunoprecipitation Sequencing

RNA-Seq RNA sequencing

SIC Schwarz Information Criterion

SNE Single Nucleotide Editing

TSS Transcription Start Site

TTS Transcription Terminator Site

UTRs Untranslated Regions

Chapter 1

Scientific valorisation and teaching

1.1 Scientific publications

1. **Arnaud Liehrmann**, Étienne Delannoy, Alexandra Launay-Avon, Élodie Gilbault, Olivier Loudet, Benoît Castandet and Guillem Rigaill. "DiffSegR : An RNA-Seq data driven method for differential expression analysis using changepoint detection", *NAR Genomics and Bioinformatics*, 2023, https://doi.org/10.1093/nargab/lqad098

2. Huy Cuong Tran, Vivian Schmitt, Sbatie Lama, Chuande Wang, Alexandra Launay-Avon, Katja Bernfur, Kristin Sultan, Kasim Khan, Véronique Brunaud, **Arnaud Liehrmann**, Benoît Castandet, Fredrik Levander, Allan G Rasmusson, Hakim Mireau, Etienne Delannoy and Olivier Van Aken. "An mTRAN-mRNA interaction mediates mitochondrial translation initiation in plants", *Science*, 2023, https://doi.org/10.1126/science.adg0995

3. **Arnaud Liehrmann** and Guillem Rigaill. "Ms.FPOP : an exact and fast segmentation algorithm with a multiscale penalty", *arXiv*, 2023, https://doi.org/10.48550/arXiv.2303.08723 (Submitted)

4. Vincent Runge, **Arnaud Liehrmann** and Pauline Spinga. "Exponential integral solutions for fixation time in Wright-Fisher model with selection", *arXiv*, 2022, https://doi.org/10.48550/arXiv.2205.06480

5. Marine Guilcher, **Arnaud Liehrmann**, Chloé Seyman, Thomas Blein, Guillem Rigaill, Benoît Castandet and Étienne Delannoy. "Full length transcriptome highlights the coordination of plastid transcript processing", *International Journal of Molecular Sciences*, 2021, https://doi.org/10.3390/ijms222011297

6. **Arnaud Liehrmann**, Guillem Rigaill and Toby Dylan Hocking. "Increased peak detection accuracy in over-dispersed ChIP-seq data with supervised segmentation models", *BMC Bioinformatics*, 2021, https://doi.org/10.1186/s12859-021-04221-5

1.2 Oral communications

1.2.1 Invited

1. **Arnaud Liehrmann** and Guillem Rigaill. "Ms.FPOP : An exact and fast segmentation algorithm with a multi-scale penalty", StatScale ECR Meeting, **United Kingdom**, 2022
(Talk)

2. **Arnaud Liehrmann**, Benoît Castandet and Guillem Rigaill. "Systematic Identification of Differential Regulation Events in RNA-Seq Data with DiffSegR", Gordon Research Seminar : Post-Transcriptional Gene Regulation, **United States**, 2022
(Talk)

1.2.2 Others

1. **Arnaud Liehrmann**, Benoît Castandet and Guillem Rigaill. "Systematic Identification Regulation Events in RNA-Seq Data with DiffSegR", Journées Jeunes Chercheurs - Biologie et Amélioration des Plantes, France, 2023
(Talk)

2. **Arnaud Liehrmann** and Guillem Rigaill. "Ms.FPOP : An exact and fast segmentation algorithm with a multi-scale penalty", Paris-Saclay Change-Point Workshop, France, 2023
(Talk)

3. **Arnaud Liehrmann**, Benoît Castandet and Guillem Rigaill. "Systematic Identification of Differential Regulation Events in RNA-Seq Data with DiffSegR", Gordon Research Conference : Post-Transcriptional Gene Regulation, **United States**, 2022
(Poster)

4. Benjamin Vacus, **Arnaud Liehrmann**, Guillem Rigaill, Benoît Castandet and Étienne Delannoy. "Using contrast to study RNA transcripts co-maturations", Open Days in Biology, Computer Science and Mathematics (JOBIM), France, 2022
(Poster)

5. **Arnaud Liehrmann**, Benoît Castandet and Guillem Rigaill. "Automatic differential analysis of transcription variants in the chloroplast with changepoint detection", Open Days in Biology, Computer Science and Mathematics (JOBIM), France, 2021
(Poster)

6. **Arnaud Liehrmann**, Benoît Castandet and Guillem Rigaill. "Automatic differential analysis of transcription variants in the chloroplast with changepoint detection (DiffSegR)", Congress of French Society of Biochemistry and Molecular Biology, France, 2021
(Talk)

7. **Arnaud Liehrmann** and Guillem Rigaill. "Ψ-FPOP : an exact and fast segmentation algorithm with a multi-scale penalty", 52nd days of statistics of the French Statistical Society (SFDS), France, 2021
   (Talk)

1.3 R packages

1. **DiffSegR** : An R package that accepts *BAM* files from an RNA-Seq dataset encompassing two distinct biological conditions, and outputs transcriptome-wide expression differences between these two conditions without using pre-existing annotations (typically genes). The R package is currently (2023) accessible at the following GitHub repository : https://github.com/aLiehrmann/DiffSegR

2. **comaturationTrackeR** : An R package that accepts *BAM* files from an RNA-Seq dataset along with an annotation file (inclusive of editing sites and introns), and outputs a list of co-maturated events. The R package is currently (2023) accessible at the following GitHub repository : https://github.com/SimiliSerpent/comaturationTracker

1.4 Teaching

1. Fall 2022 and 2023 : **Detection of transcription variants**, 1^{st} year master's students in Biology, University of Paris-Saclay, 4h

2. Fall 2021 : **Differential expression analysis**, PhD students, Genopole summer school, with Guillem Rigaill, 3h

3. Fall 2021 : **Statistics**, 2^{nd} year bachelor's students in Biology, University of Evry Val d'Essonne, 27h

1.5 Co-supervisions

1. **Benjamin Vacus**, 2^{nd} year master's student, 2022 (6 months), "Analysis of nanopore data to study co-maturations of the chloroplast transcriptome", with Benoît Castandet and Guillem Rigaill

2. **Chloé Seyman**, 3^{rd} year bachelor's student, 2021 (3 months), "Coordination of plastid transcript processing analysis using nanopore sequencing data", with Guillem Rigaill

Chapter 2

How to read this manuscript

2.1 On what and with whom

During my doctoral research, at the crossroad of biology, statistics, bioinformatics and computer science, I worked on the development and the application of statistical models, algorithms and methods for the analysis and interpretation of high-throughput biological (sequencing) data. I submitted or published three research articles as the first author, along with another article as the second author :

1. Liehrmann et al. [2021] is a modeling research article where, in collaboration with Guillem Rigaill and Toby Hocking (Northern Arizona University), I compared different multiple changepoint detection models and specialized bioinformatics heuristics within the context of epigenetic mark detection ;
2. Liehrmann et al. [2023] is a methodological and applied research article where, in collaboration with Étienne Delannoy, Guillem Rigaill and Benoît Castandet, I introduced DiffSegR, a method designed to identify transcriptome-wide expression differences across two biological conditions in Ribonucleic Acid (RNA) sequencing data ;
3. Liehrmann and Rigaill [2023] is an algorithmic research article where, in collaboration with Guillem Rigaill, I introduced Multiscale Functional Pruning Optimal Partitioning (Ms.FPOP), a fast and exact multiple changepoint detection algorithm incorporating a multiscale penalty [Verzelen et al., 2020] ;
4. Guilcher et al. [2021] is an applied research article where we studied the coordination of chloroplast RNA maturation events at the transcriptome-scale using Nanopore-based RNA sequencing data.

This last paper was made possible through the development of a method called comaturationTrackeR. This was a collaborative project I initially embarked on with Chloé Seyman, a bachelor's student, and later continued with Benjamin Vacus, a master's student. I had the opportunity to co-supervise Chloé and Benjamin during the initial two years of my doctoral research.

In the ensuing chapters of this manuscript, I provide different perspectives on one or more of these research articles, which can be found in the Appendix. I recommend that the reader first

goes through the introductory Chapters 3 and 4 in their entirety, then refers back to Figure 2.1 for a deeper investigation of a particular problem of interest. As depicted in Figure 2.1, Chapters 5 and 6, which introduce more technical aspects of my papers, can be read in any order that aligns with the reader's preference. Nonetheless, it may be beneficial to first familiarize oneself with the standard changepoints model presented in Chapter 5, as it forms the core of the DiffSegR method introduced in Chapter 6.

2.2 Chapter 3

My thesis predominantly explores the transcriptome, which refers to the comprehensive set of RNA molecules generated within a specific cell, tissue, or organism during a particular developmental or physiological stage. Two of my research papers, Liehrmann et al. [2023] and Guilcher et al. [2021], directly engage with its analysis. To contextualize these articles, in Chapter 3, which also serves as a general introduction, I illustrate a multiscale perspective of transcriptome analysis (Section 3.2.1), spanning from the gene-level, event-level, pair of events level, to the isoform-level. I highlight a series of challenges that encompass technical, statistical, and biological factors encountered at each scale (Section 3.2.2). These challenges are particularly acute at the isoform-level. In conclusion, I suggest two strategies, Strategy 1 and Strategy 2, to improve transcriptome analysis (Section 3.2.3). With my co-authors, I employed Strategy 1 and Strategy 2 in Liehrmann et al. [2023]. We also applied Strategy 1 in Guilcher et al. [2021].

2.3 Chapter 4

Throughout my doctoral research, I have had the privilege to work at the intersection of several disciplines—biology, statistics, bioinformatics and computer science—each offering unique insights and challenges in the study of high-throughput sequencing data. This interdisciplinary collaboration involved close discussions with biologists, statisticians, bioinformaticians. Not without effort, I tried to interpret biological questions and appropriately exploit statistical tools to navigate the complexity of sequencing data. This was only made possible by adopting a patient and attentive approach that values dialogue between disciplines. I was lucky enough to land in research teams where such interdisciplinary dialogue was already an established practice, and supported by researchers who are truly convinced of its usefulness. In this chapter, I try to make these disciplines dialogue by providing a concise overview of :

1. the precise biological questions that I investigated ;
2. the corresponding statistical problems ; and
3. the statistical models that I proposed to tackle these specific problems.

2.4 Chapters 5 to 6

I have written Chapters 5 and 6 as technical introductions to my four research articles.

In Chapter 5, I begin by introducing a standard model for multiple changepoint detection (Sections 5.1 to 5.3), a model used in my research articles Liehrmann et al. [2021], Liehrmann et al. [2023] and Liehrmann and Rigaill [2023]. Subsequently, I present a comprehensive review of dynamic programming techniques to optimize this model (Section 5.4). I leveraged an extension of one of these techniques (functional pruning) in Ms.FPOP Liehrmann and Rigaill [2023]. Towards the end of this chapter (Section 5.5), I introduce the Ms.FPOP algorithm, beginning by highlighting the statistical advantages of the multiscale penalty it employs, along with the algorithmic challenge linked to optimizing the standard changepoints model with this type of penalty. Lastly, I offer a brief description of how functional pruning operates within Ms.FPOP.

In Chapter 6, I articulate a rigorous strategy for the differential analysis of genes, events, and pairs of event sites. This generic strategy prominently rests on a negative binomial Generalized Linear Model (GLM), and an adaptive error control approach through a post-hoc procedure. The methods to which I have actively contributed in development, namely DiffSegR (Section 6.4) and comaturationTrackeR (Section 6.5), implement this strategy.

2.5 Chapters 7

In Chapter 7, I provide some perspectives pertaining to the studies carried out in the course of this thesis.

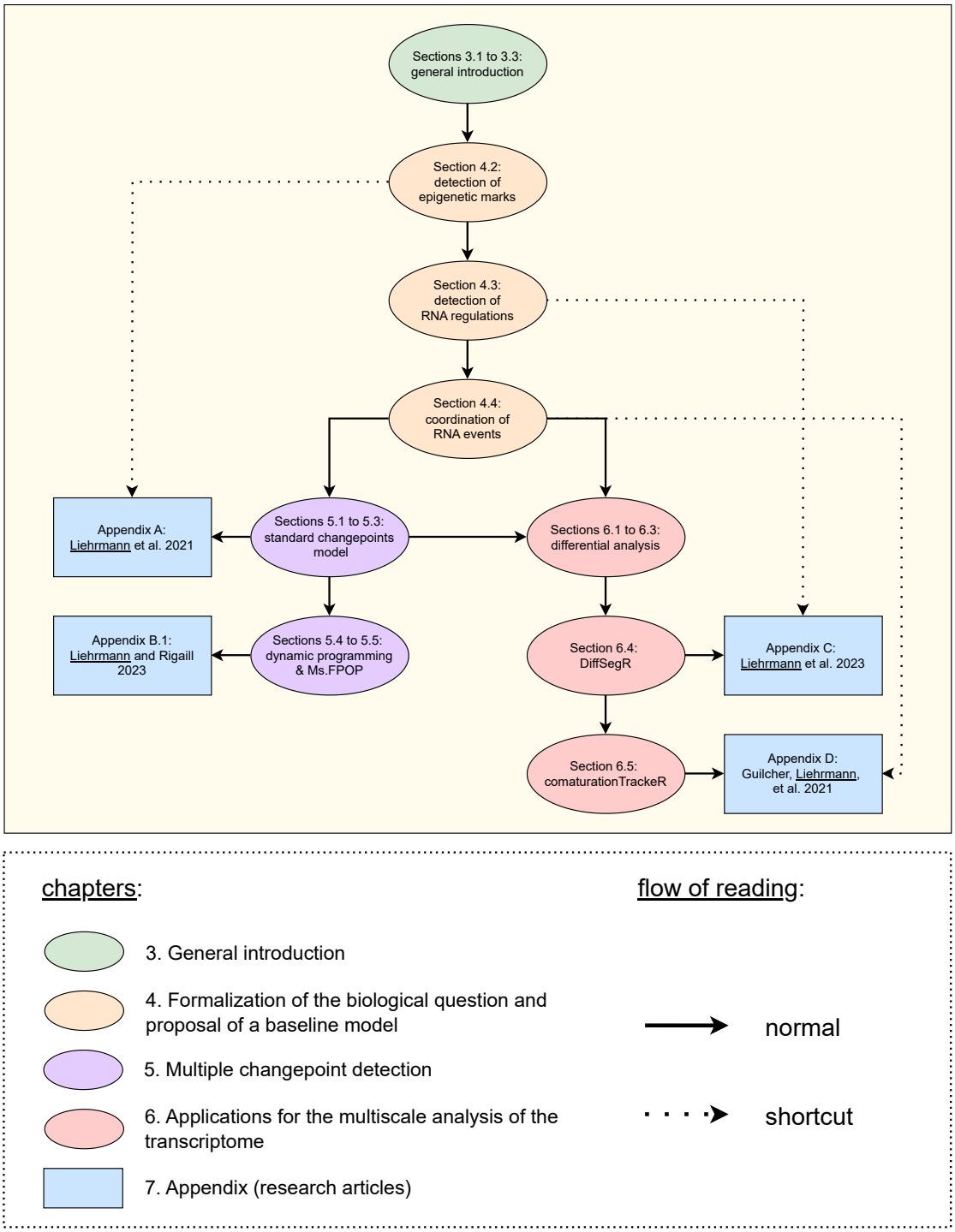


FIGURE 2.1 – Dependencies between the sections of this manuscript.

Chapter 3

Introduction

3.1 The transcriptome

3.1.1 RNA's pivotal role in the central dogma of molecular biology

3.1.1.1 The central dogma

The central dogma of molecular biology is a fundamental principle that describes the flow of genetic "information" within a cell through the use of three major classes of biopolymers : Deoxyribonucleic Acid (DNA), RNA and proteins. It was first proposed by Francis Crick in 1957, subsequently published in 1958 [Crick, 1958], and specified by the same author in 1970 [Crick, 1970]. The term "information" here refers to the precise determination of sequence, either of nucleotides in the nucleic acid or of amino acid residues in the protein. According to Crick's definition, once genetic information has passed into a protein, it cannot be transferred back to nucleic acids. In more detail, the transfer of information from nucleic acid to nucleic acid, or from nucleic acid to protein, is possible. However, the transfer of information from protein to protein or from protein to nucleic acid is not (Figure 3.1). It is worth noting that the central dogma of molecular biology has been rigorously tested, and not contradicted, through countless experiments conducted in the latter half of the 20^{th} century and the early 21^{st} century, and continues to provide a foundation for understanding biological processes at the molecular level.

3.1.1.2 The RNA, a central biopolymer in the central dogma

Within this framework, RNA plays a critical role in connecting DNA, which carries the genotype (the complete set of an organism's genetic information) [Johnson et al., 2002], and proteins, which constitute the highest level of biopolymers that link genotype to phenotype (the observable physical and functional traits of an organism) [Hartwell et al., 1999, Nussinov et al., 2019]. Therefore, one key aspect of deciphering the genotype-phenotype relationship is to thoroughly investigate the transcriptome, which includes the complete set of RNA molecules, or simply transcripts, produced within a given cell, a tissue, or an organism at a specific developmental or physiological stage.

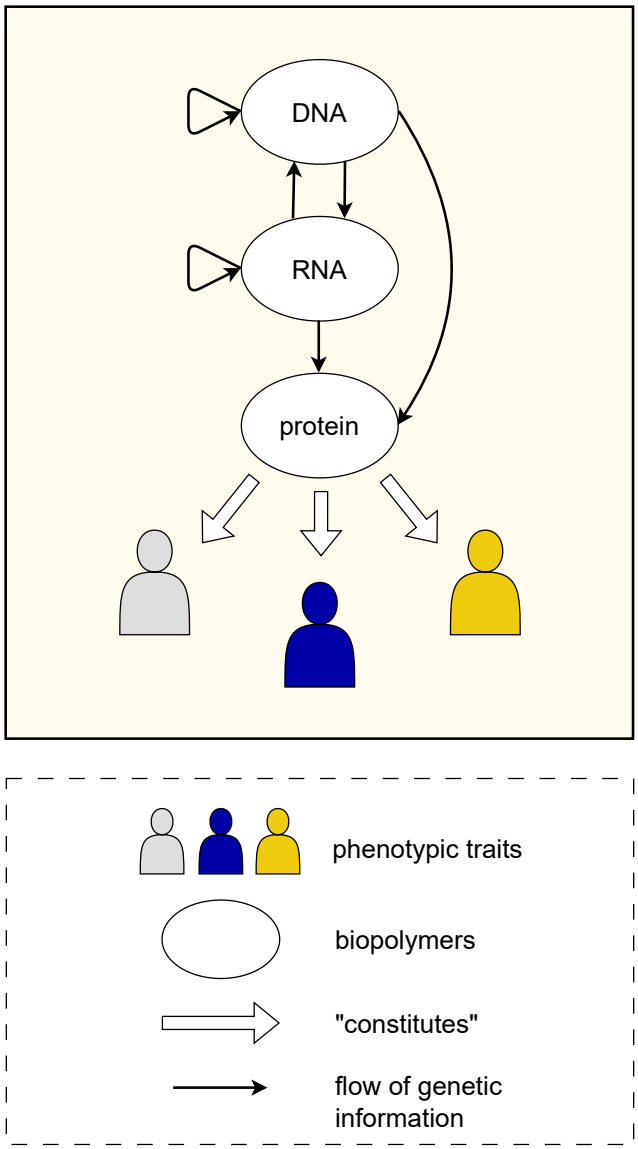


FIGURE 3.1 – The central dogma of molecular biology. Published in 1958 by Cricks, the central dogma of molecular biology describes the flow of genetic information through three major classes of biopolymers : DNA, RNA (both types of nucleic acids) and proteins. Crick posited that the transfer of information from one nucleic acid to another, as well as from nucleic acids to proteins, is possible. However, the reverse transfer of information from proteins to nucleic acids is not possible. Within this framework, the RNA plays a central role in connecting DNA, which carries the genotype, and the proteins, which constitute the highest level of biopolymers that link genotype to phenotype. Consequently, the transcriptome—the comprehensive set of RNA molecules generated within a specific cell, tissue, or organism during a particular developmental or physiological stage—becomes an ideal subject for exploring the genotype-phenotype relationship. Notably, since its description, the central dogma has withstood the test of time, remaining consistent with all experimental findings to date.

In the subsequent sections, I will provide a comprehensive overview of two fundamental biological processes that contribute to RNA metabolism :

RNA transcription —the process through which genetic information from a specific DNA segment is copied into RNA, and

RNA maturation —the process through which a transcript is modified.

These biological processes play a crucial role in shaping transcriptome diversity, which in turn plays a crucial role in determining phenotypic outcomes.

3.1.2 RNA metabolism

3.1.2.1 The RNA transcription, first stage of gene expression

The DNA template. The DNA is a biopolymer consisting of a well-ordered sequence of four nucleotides most commonly referred to as bases : adenine (A), cytosine (C), thymine (T), and guanine (G). As mentioned above, the intricate arrangement of nucleotides constitutes the genetic information. In the cell, DNA molecules consist of two complementary oriented strands, with A pairing with T and C pairing with G, forming a distinctive double helix structure. Both strands undergo transcription, with the transcribed DNA segment encompassing one or more genes. Transcription is the first stage of a series of biological processes known as "gene expression", which is responsible for producing the functional RNAs and the proteins.

The transcription in three steps. The transcription process can be broadly divided into three steps common among various life forms :

(initiation) the process starts with the binding of an RNA polymerase to a promoter region which is located upstream of the Transcription Start Site (TSS), i.e. the first base being transcribed (Figure 3.2.A) ;

(elongation) then the RNA polymerase unwinds the DNA molecule and starts synthesizing the RNA molecule from the TSS using the template strand ($3' \rightarrow 5'$) of the DNA. The adenine of the template strand is paired with uracil (in RNA, uracil 'U' replaces T present in DNA), T with A, G with C, and C with G (Figure 3.2.B) ;

(termination) and finally the RNA polymerase dissociates from the newly synthesized RNA molecule and the DNA molecule at the level of the Transcription Terminator Site (TTS), i.e. the last base being transcribed (Figure 3.2.C).

In the remainder of this subsection I review a key transcription mechanism that allows a single gene to potentially produce several transcripts, known as alternative isoforms, in varying amounts.

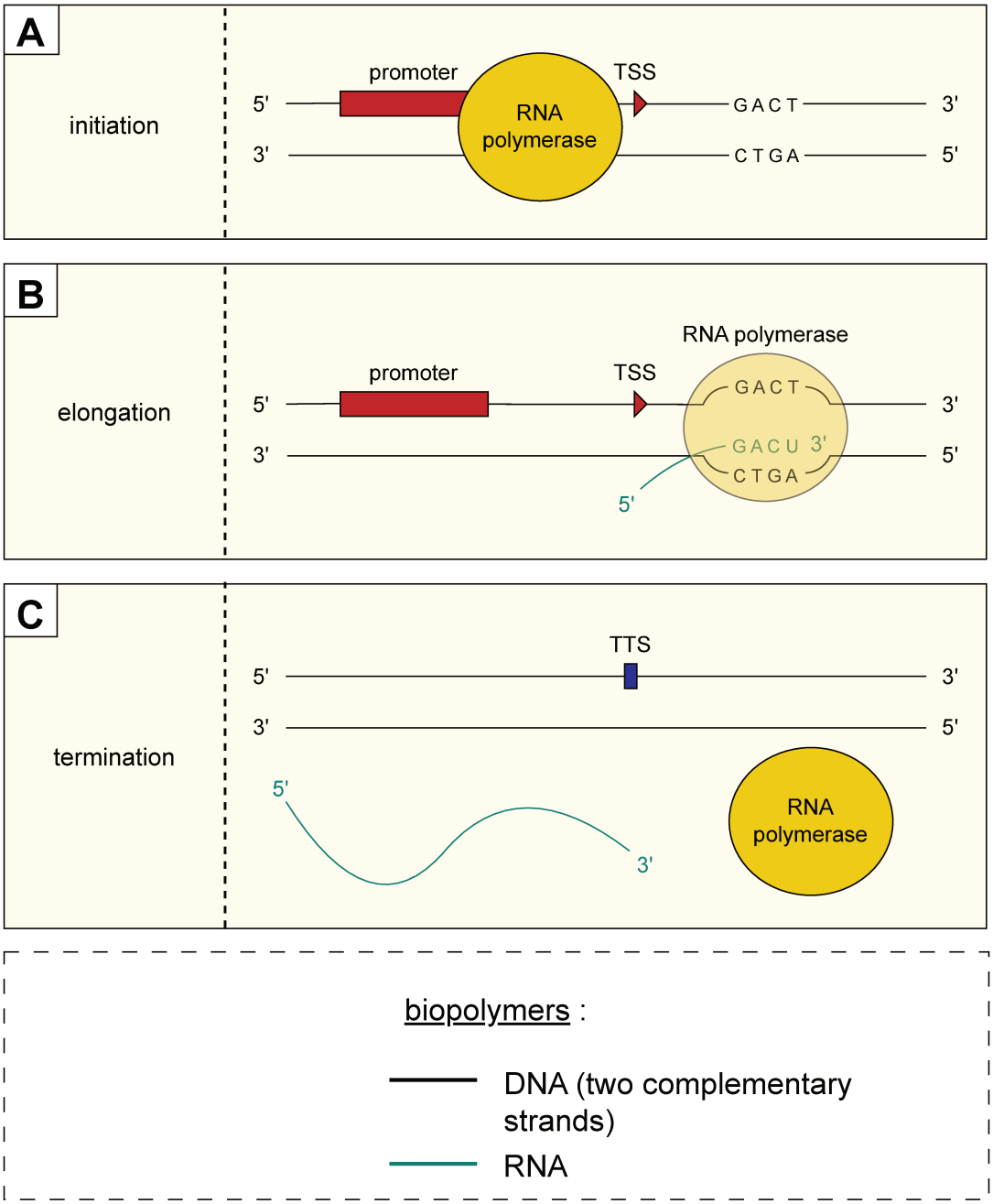


FIGURE 3.2 – The transcription of RNA, first stage of gene expression. The transcription process consists of three main steps : initiation, elongation, and termination. **(A)** Initiation begins with RNA polymerase binding to a promoter region, located upstream of the transcription start site TSS. **(B)** During elongation, the RNA polymerase unwinds the DNA and synthesizes the RNA molecule using the DNA template strand ($3' \rightarrow 5'$), pairing complementary bases. **(C)** Termination occurs when the RNA polymerase detaches from the RNA and DNA molecules at the transcription terminator site TTS.

The alternative transcription initiation. Near a gene promoter, it is common to find not just a single TSS, but instead a group of TSS known as a transcription start region. Additionally, a gene may contain several transcription start regions, suggesting the existence of alternative promoters whose choice is known to influence transcription efficiency [Juven-Gershon et al., 2008, Lenhard et al., 2012]. When the RNA polymerases start synthesizing transcripts from this collection of TSS, they produce varying amounts of isoforms with different lengths and distinct 5' end positions (Figure 3.3.B). This widespread biological process is well known as Alternative Transcription Initiation (ATI) [Policastro and Zentner, 2021].

Several studies have described large-scale shifts in patterns of transcription initiation during development [Batut et al., 2012, Zhang et al., 2017b, Cvetesic et al., 2018, Danks et al., 2018]. ATI has also been implicated in human diseases, including cancer [Sandelin et al., 2007, Davuluri et al., 2008, Demircioğlu et al., 2019]. Additionally, in bacterial life forms, it operates as an adaptive response to environmental fluctuations [Ishihama, 2000, Liu and Wulf, 2004, Typas et al., 2007]. Broadly speaking, ATI has been demonstrated to regulate RNA stability, translation efficiency [Leppek et al., 2017, Kurihara et al., 2018], and the generation of alternative isoforms with distinct protein-coding potential [Mejía-Guerra et al., 2015, Ushijima et al., 2017].

Besides ATI, the main sources of alternative isoforms come from maturation mechanisms that occur simultaneously with or after RNA transcription. In the following subsection we examine key maturation mechanisms that substantially contribute to the transcriptome complexity.

3.1.2.2 The RNA maturation

The alternative splicing. Groundbreaking research in the 1970s [Berget et al., 1977, Aloni et al., 1977, Breathnach et al., 1977, Doel et al., 1977] revealed that eukaryotic gene organization does not consist of continuous nucleotide sequences encoding proteins. Instead, genes are segmented with protein-coding exons separated by intervening sequences known as "introns" (a term introduced by Gilbert in 1978 for intragenic regions). During transcription, these introns are excised from a precursor, and the remaining exons are joined together in a process called RNA splicing. This process enables the formation of alternative isoforms through Alternative Splicing (AS) [Gilbert, 1978]. Various AS modes have been observed, with two common ones being exon skipping and intron retention [Wang et al., 2014, Gehring and Roignant, 2021]. In these modes, a specific exon or intron may be included or excluded resulting in two alternative isoforms (Figure 3.3.C). It is worth noting that AS has also been observed in many precursors of non-coding RNA [Khan et al., 2021].

Several studies have revealed that pervasive AS events vary across different tissue types and developmental stages [Wang et al., 2008, Pan et al., 2008, Kalsotra and Cooper, 2011]. Specific AS events have also been implicated in human diseases such as cancer, amyotrophic lateral sclerosis or Alzheimer's disease [Scotti and Swanson, 2015, Love et al., 2015, Bonnal et al., 2020]. In general, AS serves as a prevalent mechanism for generating alternative isoforms that possess unique protein-coding capacities [Nilsen and Graveley, 2010].

The processing of extremities. The 5' end or 3' end of transcripts are not always defined by their TSS or TTS, but instead by the cleavage and/or the degradation of a precursor, possibly accompanied by the subsequent synthesis of additional bases (see below), a phenomenon observed in all kingdoms of life [Condon, 2003, Clouet-d'Orval et al., 2018, Kim et al., 2004, Gregory et al., 2008]. For instance, the mature 3' ends of nearly all eukaryotic messenger RNAs (mRNAs)—an RNA that encodes a protein—are created by a two-step reaction that involves an endonucleolytic cleavage of a precursor, followed by the synthesis of a polyadenylate tail onto the upstream cleavage product. Like TSS, it is common to find several of these cleavage sites, also called Polyadenylation Sites (PAS), by gene. The PAS can be located within the 3' Untranslated Regions (UTRs), introns, or exons. Like ATI, alternative usage of PAS, or simply Alternative Polyadenylation (APA), allows a single gene to encode multiple alternative isoforms [Giammartino et al., 2011, Tian and Manley, 2016] (Figure 3.3.D).

Numerous research findings indicate that APA plays a role in activating oncogenes and promoting cell proliferation in cancer cells [Sandberg et al., 2008, Mayr and Bartel, 2009]. Additionally, APA has also been implicated in development [Shepard et al., 2011, Agarwal et al., 2021]. Investigations have also revealed that APA impacts neuronal signaling and function [Flavell et al., 2008, Miura et al., 2013, Tushev et al., 2018]. At the molecular level, APA can modify the coding potential of mRNA or change the length of the 3' UTR, which in turn affects mRNA fate in various ways, such as by altering binding sites for proteins and microRNAs—a specific type of small RNAs [Neve et al., 2017, Hong and Jeong, 2023].

Single nucleotide editing. After transcription, an RNA molecule can undergo Single Nucleotide Editing (SNE), which involves the precise conversion/alteration of individual nucleotides. This process leads to a difference in sequence between the original DNA template and the edited RNA product. Referring to Knoop's classification [Knoop, 2010], RNA SNE includes any nucleotide conversions and any chemical alterations to the four standard nucleotides (A,U,G,C). For instance, the most common form of RNA SNE in metazoans is the conversion of A to Inosine (I) by an adenosine deaminase (Figure 3.3.E). I is interpreted as G by cellular machinery, leading to alterations in structural properties of RNA and protein sequences [Nishikura, 2006, 2010, 2015, Eisenberg and Levanon, 2018].

In various organs and tissues of model metazoans, RNA SNE has been shown to regulate developmental processes [Buchumenski et al., 2021, Graveley et al., 2010], neural network plasticity [Behm and Öhman, 2016, Rosenthal and Seeburg, 2012], immune responses [Mannion et al., 2014, Liddicoat et al., 2015], skeletal muscle formation [Noda et al., 2022], and organismal adaptation to environmental changes [Buchumenski et al., 2021]. Deficiencies in the RNA editing machinery have been linked to neurological disorders, autoimmune diseases, and even cancers in humans [Zipeto et al., 2015, Ben-Aroya and Levanon, 2018].

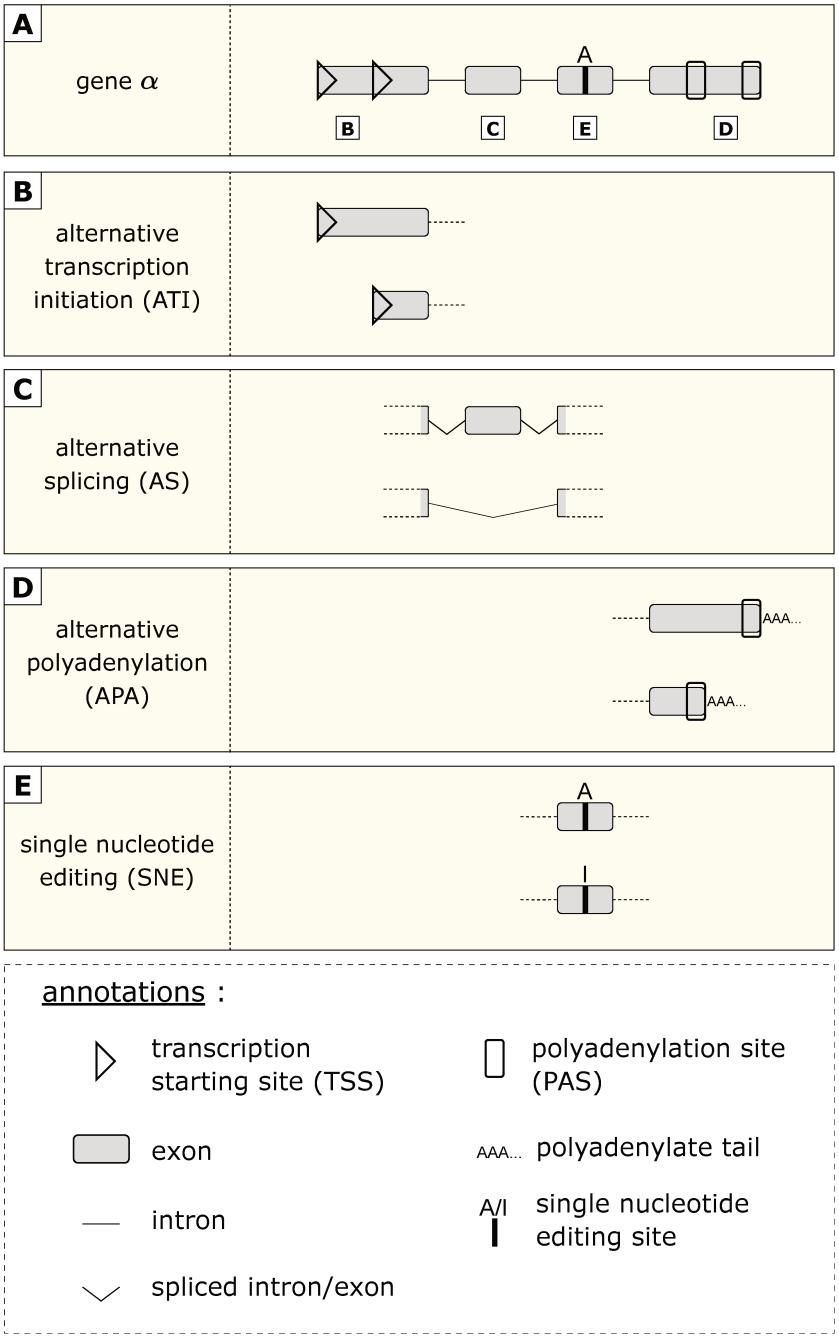


FIGURE 3.3 – Key transcriptional and maturation mechanisms enable a single gene to generate multiple RNA isoforms. **(B)** Alternative Transcription Initiation (ATI) : In the gene α (Panel **A**), there are two TSS. RNA polymerase can initiate transcription from either of these sites, leading to the production of two alternative transcripts with distinct 5' end positions. ATI has been shown to control the stability of RNA, the efficiency of translation, and the production of alternative isoforms that possess unique protein-coding capabilities. **(C)** Alternative Splicing (AS) : The gene α contains an exon-skipping event, in which the second exon can be either included or excluded during the RNA splicing process. This process generates two alternative transcripts with unique protein-coding capacities and potentially different biological functions. **(D)** Alternative Polyadenylation (APA) : In gene α, there are two polyadenylation sites that can be used to create different mature 3' ends of the mRNA. APA has the ability to alter the coding capacity of mRNA or adjust the length of the 3' UTR, which in turn affects mRNA fate in various ways, such as by altering binding sites for proteins and microRNAs. **(E)** The gene α features a nucleotide within its third exon that can undergo precise conversion A to I on the corresponding RNA molecule. Single nucleotide editing (SNE) can lead to alterations in structural properties of RNA and protein sequences.

3.1.3 RNA events

3.1.3.1 Definition

I will refer to the regions of RNA that are affected by ATI or maturation processes described in the previous section as "variable regions", or more simply, "variables". These variables are intended to be transcribed/not-transcribed, eliminated/not-eliminated, or edited/not-edited. Furthermore, ATI and maturation processes can accumulate, creating a sequence of events along the RNA molecule, resulting in the formation of an isoform. For example, if a gene, which we will call α, has 2 alternative TSS, 1 alternative exon, 1 editing site and 2 alternative PAS (Figure 3.3.A), then 4 events can occur (or not) and accumulate along the corresponding RNA molecule :

event		**complementary event**
the first TSS is chosen	or	the second TSS is chosen ;
the alternative exon is included	or	the alternative exon is spliced ;
the nucleotide is edited (I)	or	the nucleotide is not edited (A) ;
the first PAS is chosen	or	the second PAS is chosen.

When a gene possesses multiple TSS or PAS, it is beneficial to define an event for each TSS or PAS. These events are mutually exclusive, meaning they cannot occur simultaneously, since each isoform is linked to a distinct TSS or PAS. Not all mutually exclusive events are as evident as the case of multiple TSS or PAS. For instance, this phenomenon can also occur between two exons [Pohl et al., 2013] and all such coordinations are currently not known.

3.1.3.2 Upper bound complexity

At least theoretically, the number of isoforms we could generate from a set of events of size K is exponential, formally of the order of 2^K. For instance, $2^4 = 16$ theoretical isoforms could be produced from the 4 events annotated on gene α (Figure 3.4). However, it is essential to recognize this calculation as an upper bound, since some events may, as exemplified above, be mutually exclusive, and hence not all combinations may be feasible. This raises the question : *Will all combinations of events actually be expressed ?* While it may not always be the case, it is certainly not impossible. For instance, in the *Drosophila melanogaster* transcriptome, the DSCAM gene demonstrated an astonishing 18,496 observed isoforms, almost reaching the theoretical limit of 19,008 possible combinations [Sun et al., 2013].

For the sake of clarity and brevity in the ensuing discussion of this manuscript, I will use the upper bound (2^K) to refer the number of isoforms generated by a specific gene.

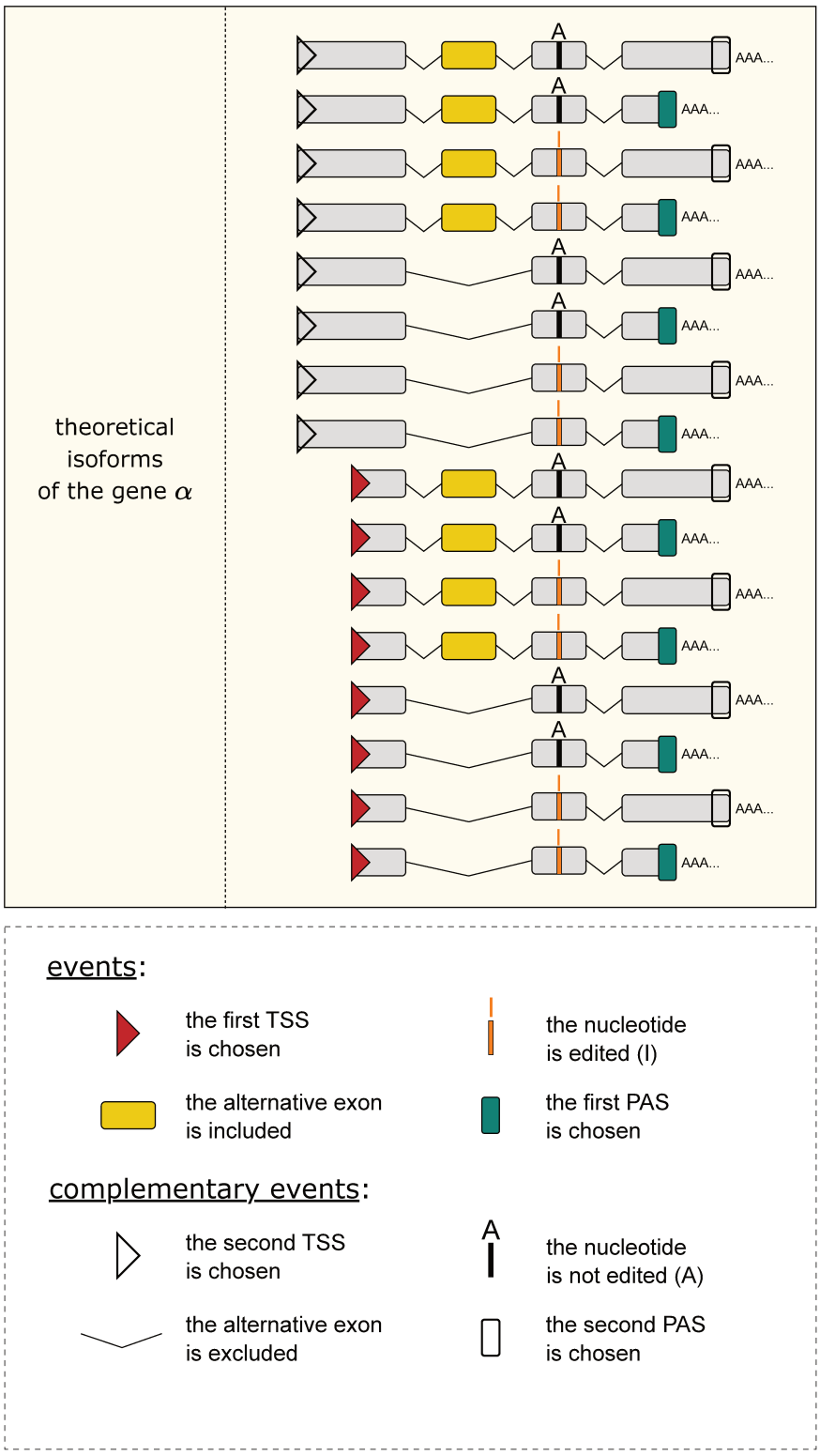


FIGURE 3.4 – RNA events. ATI and RNA maturation processes may result in a series of events occurring along RNA molecules. In the case of the gene α, there are four possible events that can accumulate : the selection of the first TSS ; the inclusion of the alternative exon (second exon) ; the choice of the first PAS ; and the nucleotide editing in the third exon. Various combinations of these events can generate $2^4 = 16$ theoretical RNA isoforms for the gene α.

3.2 The multiscale analysis of the transcriptome

Today biologists and bioinformaticians are exploring the transcriptome[1] at multiple scales [Berge et al., 2019]. These scales are defined by the number of events jointly studied along RNA molecules.

3.2.1 Definition of the scales

3.2.1.1 Analysis of the transcriptome at the gene-level

In the majority of studies, researchers seek to understand the global patterns of gene expression, possibly under various biological conditions such as normal, developmental, or pathological conditions [Kim et al., 2001, Merryweather-Clarke et al., 2011, Tello-Ruiz et al., 2015, Yang et al., 2016, Singh et al., 2017, Hahn et al., 2021]. Numerous specialized tools have been designed or predominantly employed for this type of analysis [Robinson et al., 2009, Hardcastle and Kelly, 2010, Tarazona et al., 2012, Ritchie et al., 2015, Pimentel et al., 2017]. DESeq2 [Love et al., 2014], a package initially developed to investigate systematic changes of expression at gene-level across various experimental conditions, serves as a prime example, boasting over 49,000 citations in 2023. Throughout this analysis, RNA isoforms are considered indistinctly, meaning that events happening along RNAs are overlooked.

3.2.1.2 Analysis of the transcriptome at isoform-level

Focusing solely on the aggregate of all isoforms for a gene can be an overly simplistic approach in some research contexts. For instance, the differential transcript usage analysis in Alzheimer's disease human brains reveals gene expression alterations overlooked in differential gene expression analysis [Marques-Coelho et al., 2021]. Similar alterations have been observed in Parkinson's disease [Marques-Coelho et al., 2021, Rhinn et al., 2012, Dick et al., 2020]. To address this limitation, some molecular biologists place more emphasis on examining individual isoforms using dedicated tools like RSEM [Li and Dewey, 2011], Cufflink [Trapnell et al., 2012], Salmon [Patro et al., 2017], and many others [Glaus et al., 2012, Bernard et al., 2014, Bray et al., 2016, Tang et al., 2020, Hu et al., 2021, Gleeson et al., 2021, Prjibelski et al., 2023, Hu et al., 2023]. Throughout the analysis at isoform-level all events are monitored jointly along RNAs, which is indeed equal to investigating isoforms.

3.2.1.3 Analysis of the transcriptome at event-level

An intermediate approach lies between analyzing an aggregate of all isoforms and examining each isoform individually. This approach focuses on investigating the occurrence of events independently along RNAs, possibly under various biological conditions. DEXSeq [Anders et al., 2012]

1. As a reminder, the transcriptome is the comprehensive set of transcripts generated within a specific cell, tissue, or organism during a particular developmental or physiological stage.

is a widely recognized method employed for such analyses. In addition, the field is continually expanding with the development of newer tools [Shen et al., 2014, Tran et al., 2016, Li et al., 2017, Yalamanchili et al., 2020, Policastro and Zentner, 2021]. As a result, it enables researchers to gain insights into gene regulation with greater resolution compared to aggregate gene-level analysis. Simultaneously, it avoids the complexity associated with exploring each individual isoform (as described hereafter).

3.2.1.4 Analysis of the transcriptome by pair of events (or more)

However, it is highly improbable that all biological processes affecting RNA variables are independent. In fact, numerous dependencies have already been identified. For example, two exons may exhibit coordination if both are under similar control of polymerase speed [Fededa et al., 2005]. Mechanisms connecting the selection of PAS and exon inclusion at the 3' ends of genes have also been suggested [Black, 2003, Movassat et al., 2016, Hardwick et al., 2022]. Moreover, distinct promoter usage can affect splicing decisions, resulting in non-random pairing of transcription start sites TSS and exons [Cramer et al., 1997, Xin et al., 2008]. At this scale, researchers are monitoring events in pairs, triplets, or even larger groups along the RNAs to account for these dependencies. While there are fewer tools designed for this type of analyses, some specialized packages do exist, such as Insplico which focuses on investigating the splicing order of neighboring introns [Gohr et al., 2023].

In the following subsection, we will discuss how an increase in the number of jointly studied events along RNA molecules makes the analysis of the transcriptome technically, statistically and biologically more challenging.

3.2.2 Challenges in transcriptome analysis

3.2.2.1 Overview of an RNA sequencing experiment

Over time, researchers have designed technologies, including complementary DNA (cDNA) microarrays, tilling arrays, Illumina-based RNA sequencing (RNA-Seq) and Nanopore/Pacbio-based RNA-Seq, whose results have been used as a proxy of RNAs produced by genes [Schena et al., 1995, Lockhart et al., 1996, Perou et al., 2000, Johnson et al., 2005, Mortazavi et al., 2008, Branton et al., 2008, Wang et al., 2009]. Currently, the most widely adopted technology is the Illumina-based RNA-Seq, which provides a quantitative, large-scale approach for analyzing transcriptional outcomes.

A conventional RNA-Seq experiment (Figure 3.5) begins with the extraction and the selection of RNAs, which are then fragmented into smaller segments typically spanning from 300 to 500 nucleotides (nt) in length (a common unit of length for single-stranded nucleic acids). These RNA fragments then undergo reverse transcription into cDNA, subsequently prepared for the sequencing stage. The sequencing output is a collection of "reads"—essentially fragments of the original transcripts, with lengths ranging from 50 to 300 nt [Kukurba and Montgomery, 2015, Stark et al., 2019].

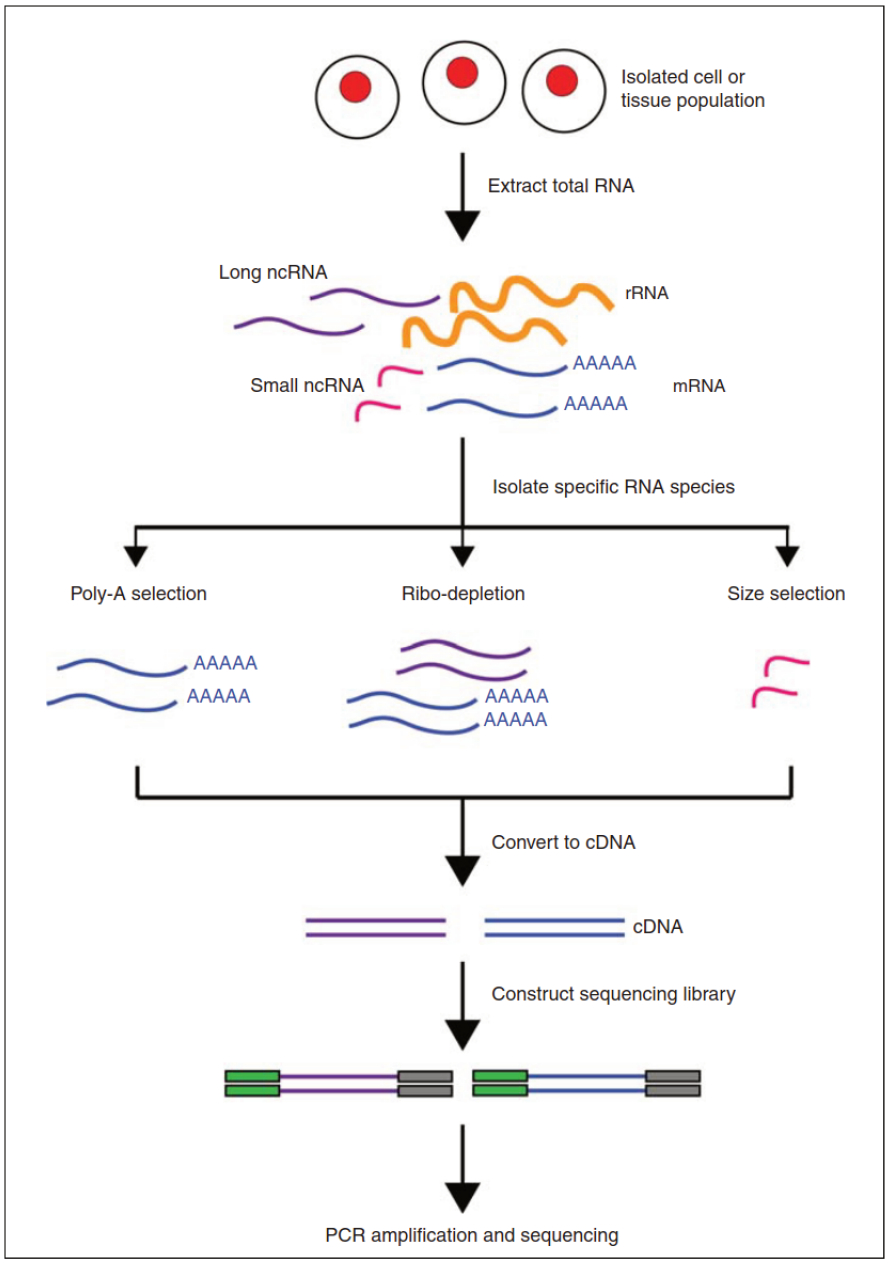


FIGURE 3.5 – RNA-Seq experiment in a nutshell. The process of preparing an RNA-Seq library begins with the extraction of RNAs from a selected biological material, like cells or tissues. This is followed by the isolation of specific RNA molecules using a defined protocol. One such protocol could be the poly-A selection method, which is employed to concentrate polyadenylated transcripts, or a ribo-depletion protocol that aids in removing ribosomal RNAs (rRNAs), or a selection based on the size of transcripts. Upon extraction and selection, the RNA molecules are typically fragmented (not shown on this diagram). In the subsequent phase, the RNA is converted into cDNA through a procedure known as reverse transcription. Once the cDNA is formed, sequencing adaptors—small pieces of known DNA sequences—are affixed to the cDNA fragment ends. Finally, a Polymerase Chain Reaction (PCR) is used to amplify these fragments. This step increases the amount of DNA available for sequencing. Post-amplification, the RNA-Seq library is sequenced using a high-throughput sequencing platform, such as Illumina. The figure is derived from Kukurba and Montgomery [2015].

3.2.2.2 Quantification challenge

After completing an RNA-Seq experiment, the analyst is presented with files containing millions of sequencing reads. Quality control is conducted on the sequencing reactions [Li et al., 2014], and these reads are then aligned to a reference genome or a reference transcriptome [Srivastava et al., 2020] (Figure 3.6). The reads are subsequently assigned to compatible genes (Figure 3.7.A), events (Figure 3.7.B), event pairs (Figure 3.7.C), and so on up to the isoforms (Figure 3.7.D), depending on the analysis scope.

Nevertheless, when addressing multiple events concurrently, such as at the isoform-level, a notable challenge surfaces due to the limitation in read size that prevents complete coverage of events along RNA molecules. Consider, for instance, that typical human RNA molecules extend beyond 2000 nt [Leung et al., 2021, Lopes et al., 2021], approximately seven times the reach of the longest reads, which are confined to around 300 nt. This discrepancy inevitably results in ambiguity regarding the origin of numerous reads, thus complicating their assignment. As illustrated in Figure 3.7.D, the read that overlaps the second and third exons of gene β could potentially derive from either isoform 1 or isoform 3. These isoforms are characterized by unique TSS, which are not captured by the read in question. Indeed, the nearest TSS is located more than 100 nt away from the second exon, which is more than the length of the reads (100 nt). There is also an uncertainty with the read that overlaps with exons 1 and 3 but bypasses exon 2, suggesting potential origin from either isoform 2 or isoform 4. Probabilistic models based on maximum likelihood or bayesian inference are required for these assignments ; however, the accuracy of such models is markedly variable [Steijger et al., 2013, Mehmood et al., 2019, Sarantopoulou et al., 2021]. Furthermore, as highlighted by [Zhang et al., 2017a], a noticeable decline in accuracy is observed with a rise in the number of isoforms.

The challenge associated with read size mechanically lessen as the scale of analysis is adjusted to involve fewer concurrent events, such as single events or pairs of proximal events. This adjustment allows for straightforward quantification (Figure 3.7.B-C). It is worth highlighting that substantial enhancements in read length have been achieved thanks to recent advancements in RNA-Seq technology, specifically the development of long-read sequencing, including Nanopore and Pacbio platforms. This technological evolution potentially increases the number of events that can be captured on a single molecule [Byrne et al., 2017, Sessegolo et al., 2019, Soneson et al., 2019, Wang et al., 2021, Kovaka et al., 2023].

Apart from the fragmentation challenge, various biases inherent to RNA-Seq protocols can result in the preferential selection of certain RNAs, leading to a skewed representation of the transcriptome [Shi et al., 2021]. Furthermore, the accuracy of the quantification process is highly dependent on the quality of annotations [Angelini et al., 2014, Soneson et al., 2016], which are recognized to be incomplete for genes, events, and by extension, isoforms (see Box 1 : *Shall we ever reach a complete reference transcriptome ?*).

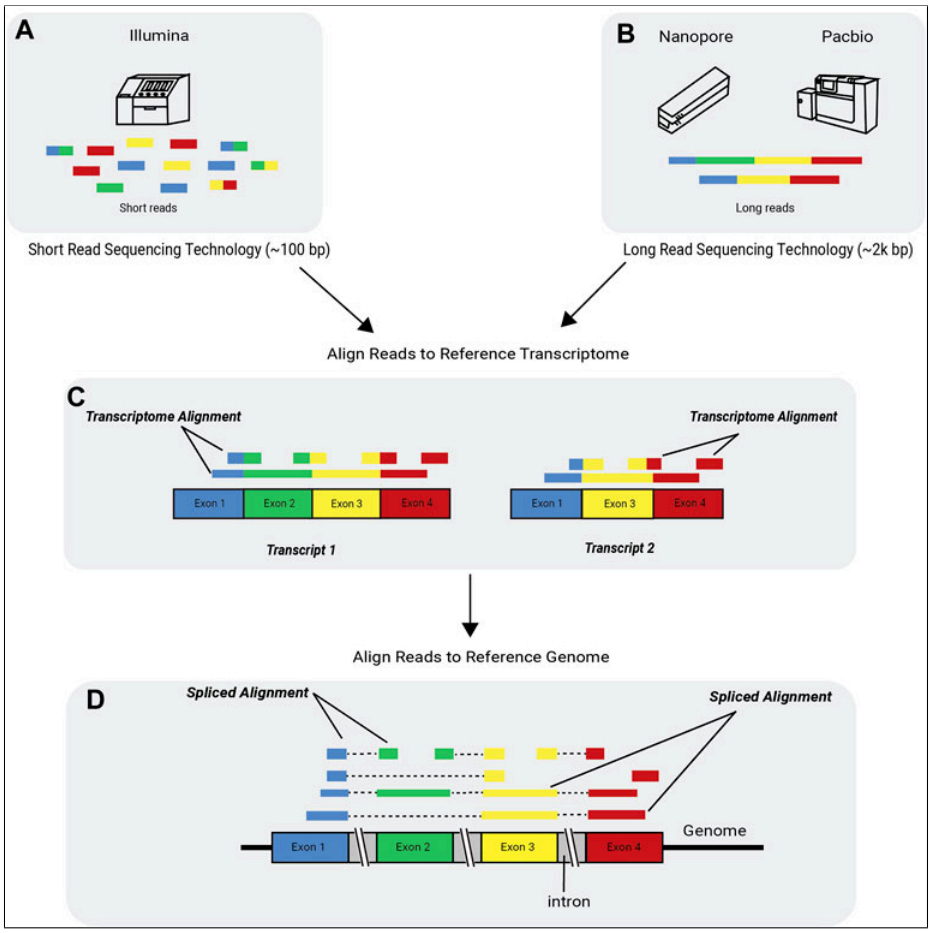


FIGURE 3.6 – Alignment of sequencing reads. Various sequencing technologies, including **(A)** the Illumina platform and **(B)** Nanopore or PacBio platforms, generate sequencing reads whose precise genomic origin—both specific region and strand—remains unknown. To identify their source, these reads are mapped to either **(C)** a reference transcriptome, encompassing all annotated RNA isoforms from all genes, or **(D)** a reference genome, consisting of all annotated genes. A notable computational hurdle in aligning RNA-seq reads to a reference genome is managing spliced junctions. These are instances where a segment of the read corresponds to the terminal region of one exon while the remainder associates with another exon, potentially thousands of base pairs distant from the first. In response to this challenge, the development of spliced-aware aligners such as STAR [Dobin et al., 2012] and HISAT [Kim et al., 2015] has taken place. The figure is derived from [Deshpande et al., 2023].

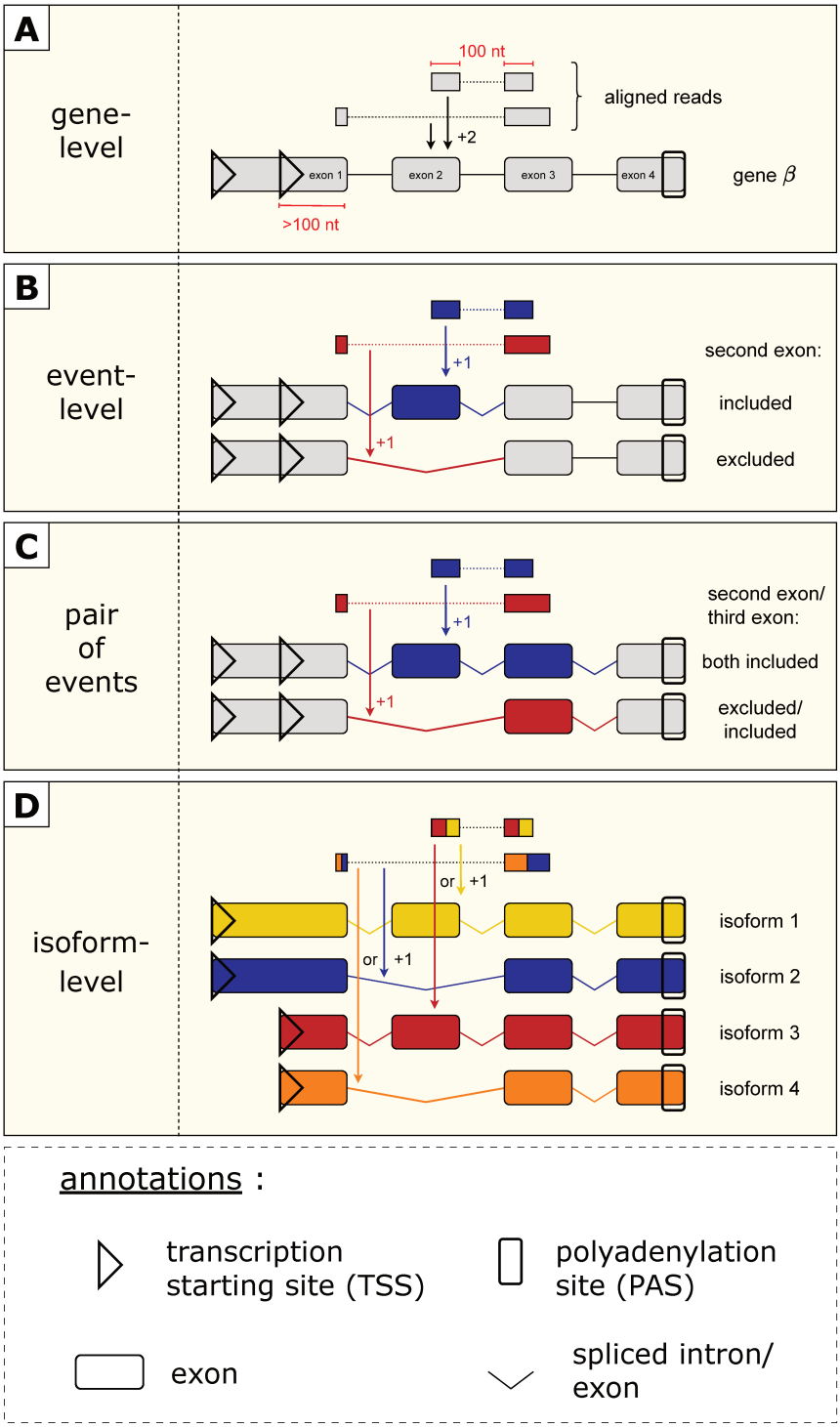


FIGURE 3.7 – Quantification challenge at each scale. **(A)** Gene-level analysis : both reads overlap with gene β and are therefore assigned to it. **(B)** Event-level analysis : studying the second exon solely, two potential outcomes are illustrated ; either the exon is excluded or included. The first read corresponds to the scenario of exon inclusion, while the second read indicates exon exclusion. Both reads are assigned to respective scenario. **(C)** Pair of events analysis : focusing on the pair formed by the second and third exons, the first read is attributed to the scenario where both exons are included. In contrast, the second read reflects the outcome where the second exon is excluded while the third exon is included. **(D)** Isoform-level analysis : the origin of both reads is ambiguous. The first read could potentially originate from either isoform 1 or 3, while the second read could derive from either isoform 2 or 4.

3.2.2.3 Statistical challenge

In a conventional RNA-Seq experiment, the study design usually involves the comparison of two distinct conditions, each represented by several biological replicates. Under this experimental design, researchers typically aim to identify : genes, events and isoforms with significant difference of expression or pairs of events with significant difference of dependency.

After the quantification phase is completed, each of the aforementioned differences, also referred to as "contrasts" can be estimated from the counts (Chapter 6). Following this estimation, statistical tests are performed to evaluate whether these contrasts significantly deviate from zero, thus addressing the original research objective [Berge et al., 2019].

It is noteworthy to mention that the number of statistical tests increases in tandem with the number of events being jointly examined. Indeed, the sequence of tests performed per gene scales as follows : a single test at the gene level, K tests at the event-level (one for each event), $\binom{K}{2}$ tests when examining events by pair (one for each pair of events), and an exponential increase to 2^K tests at the isoform-level (one for each isoform).

The escalating number of tests necessitates the implementation of multiple testing correction procedures to mitigate the risk of false positives. However, this introduces a trade-off with the statistical power of the study, making it more difficult to discern genuine differences [Goeman and Solari, 2014]. Particularly at the isoform-level, the exponential increase in the required number of tests, coupled with their intricate dependencies, accentuates this challenge.

3.2.2.4 Biological challenge

The last step in a standard transcriptomics study is often the characterization of the molecular functions or pathways in which differentially expressed genes or isoforms are involved.

At this stage the emphasis on gene-level analysis has primarily been driven by our limited knowledge of the functional differences between distinct isoforms arising from ATI and RNA maturations. In fact, despite the significance of ATI and RNA maturations, information on the cellular functions, endogenous expression and localization, and signaling pathways associated with individual isoforms, is known for only a small number of genes [Lerch et al., 2012, Kelemen et al., 2013, Yap and Makeyev, 2016, Baralle and Giudice, 2017, Bhuiyan et al., 2018]. Consequently, to date, making scientific sense out of such data is still a complicated task [Karlebach et al., 2022]. At the event-level, the analyst can leverage on the regulatory signals or functional domains in which the corresponding RNA variable is involved as valuable biological interpretation.

Moreover, gene-level discoveries are more experimentally actionable than isoform-level discoveries due to the difficulty of knocking down single isoforms [Kisielow et al., 2002]. Transgene-mediated overexpression of splicing variants of interest is also used for studying isoform-specific functions and subcellular localization in specific cells. However, it is well documented that overexpressed corresponding proteins often do not mimic the endogenous proteins in their spatio-temporal expression, localization, and functions [Prelich, 2012, Moriya, 2015].

Box 1: *Shall we ever reach a complete reference transcriptome ?*

Since the beginning of the 21^{st} century, thanks to the rapid development of high-throughput RNA-Seq technologies, major efforts have been made to build a comprehensive picture of the transcriptome generated by organisms, also known as the reference transcriptome. To achieve this, bioinformatics pipelines have been used to annotate genes and their isoforms from the sequencing data, before validation by expert biologists and bioinformaticians. These structural annotations are then made available to the scientific community via genomic databases and can be visualized in genome browsers.

As expected, the transcriptomes available in these databases are characterized by the abundance of alternative isoforms. For example, the version 43 of the human reference transcriptome proposed by the GENCODE database contains 252,913 transcripts (+497 compared to version 42) associated to 62,703 genes (+7 compared to version 42), i.e. an average ratio of observed isoforms per gene of 4.03 (`https://www.gencodegenes.org/human/stats.html`).

The average number of isoforms observed per gene in humans (4.03) seems low compared to the diversity of biological processes leading to the formation of a new isoform, and recent research suggests that the actual number of isoforms is indeed underestimated [Pertea et al., 2018, You et al., 2017]. In particular, a large number of studies have identified new variable regions of RNAs that play a prominent role in disease development [Whiffin et al., 2020, Griesemer et al., 2021, Makhnovskii et al., 2022]. In this context, part of the scientific community believes that achieving an exhaustive description of transcriptomes is ultimately a Sisyphean task (Figure 3.8) [Nellore et al., 2016, Deveson et al., 2018, Morillon and Gautheret, 2019].

3.2.3 A roadmap for improving transcriptome analysis

In previous subsections, we discussed how studying an increasing number of events along RNA molecules concurrently can make the transcriptome analysis more complex from statistical, technical, and biological standpoints. To circumvent the exponential complexity of investigating each individual isoform while still allowing researchers to gain more detailed insights into gene regulation compared to aggregate gene-level analysis, a promising approach involves :

Strategy 1

☞ developing methods that simultaneously examine a manageable number of events.

This can be done either by studying each event independently or by jointly analyzing a few events (for example, in pairs). In this context, utilizing long-read technologies can be beneficial for jointly monitoring RNA events that may be separated by hundreds or even thousands of nucleotides.

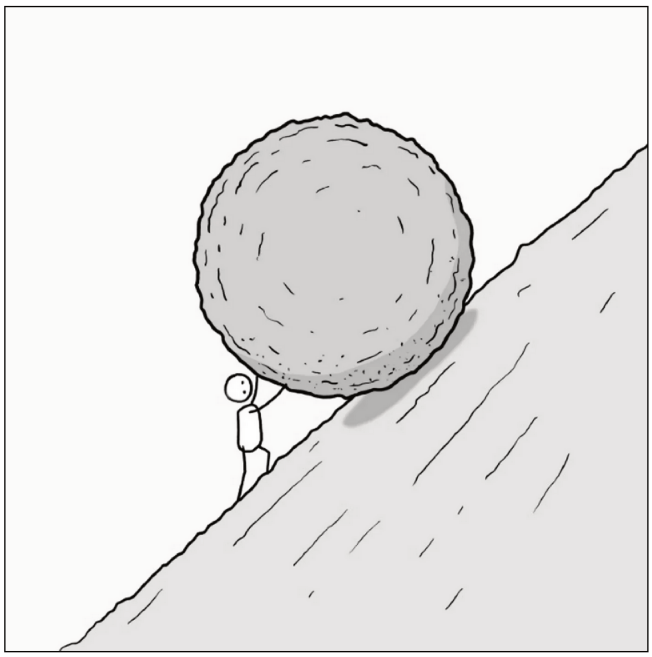


FIGURE 3.8 – The rock of Sisyphus. Sisyphus was an ancient Greek king condemned by the gods to roll a boulder up a hill forever, only to see it fall back down the hill each time he reached the top. A "Sisyphean task" thus refers to a difficult job that must be done over and over again. © The Sisyphean Task of Drawing Sisyphus / Chaz Hutton

An additional consideration is that the accuracy of the analysis results at each scale is heavily reliant on the quality of the annotations, which are known to be incomplete for genes, events, and subsequently, isoforms. As a result, to improve transcriptome analysis, another promising approach involves :

Strategy 2

☞ developing methods that analyze the transcriptome without relying on pre-existing annotations.

Such techniques are commonly recognized as data-driven approaches.

3.3 Scientific context

3.3.1 Foreword

My thesis project is part of a collaboration between two biologists, Benoît Castandet and Étienne Delannoy, from the Organellar Gene Expression team located at the Institut des Sciences des Plantes de Paris-Saclay and a statistician, Guillem Rigaill, from the Genomic Networks team and the Statistics and Genomes team located respectively at the Institut des Sciences des Plantes de Paris-Saclay and at the Laboratoire de Mathématiques et Modélisation d'Évry. The goal of this collaboration is to develop efficient bioinformatics tools that use RNA-Seq data (small reads and long reads) to study the chloroplast transcriptome.

3.3.2 The chloroplast

The chloroplast (Figure 3.9), an organelle located in cytoplasm of plant cells, is the key site of photosynthesis, a bioenergetic reaction crucial for life on Earth. It was formed by the primary endosymbiosis of a cyanobacteria-like organism [Archibald, 2009, Keeling, 2010], followed by successive transfers of genes involved in photosynthesis and its metabolism to the nuclear genome of the host cell [Timmis et al., 2004, Barbrook et al., 2006, Ponce-Toledo et al., 2019]. Today, the chloroplast contains a reduced genome (about 1.5×10^4 base pairs [2] ; for comparison, the nuclear genome of *Arabidopsis thaliana* comprises about 1.35×10^8 base pairs and that of *Homo sapiens* about 3.09×10^9), the expression of which is essential for photosynthetic activity, retrograde signaling or plant development [Fey et al., 2005].

The chloroplast is an interesting model for the effective examination and testing of methods geared towards transcriptome analysis—including that of the nuclear transcriptome—using RNA-Seq data.

1. Firstly, its genome and the expression of its approximately 120 genes (in the chloroplast of *A. thaliana*) have been extensively documented in the scientific literature [Zhang et al., 2023, Small et al., 2023].
2. Secondly, the compact size of the chloroplast facilitates a faster validation of these methods' outcomes. Indeed, the results can be quickly assessed by directly visualizing the RNA-Seq data using, for instance, the integrative genomics viewer [Thorvaldsdottir et al., 2012].
3. Finally, as I intend to illustrate in the following section, there is no indication that the fundamental processes at play in the metabolism of chloroplast RNAs are simpler than those involved in the metabolism of nuclear RNAs.

3.3.3 The metabolism of chloroplast RNAs

3.3.3.1 Extensive transcriptional activity of the chloroplast genome

In plant cells, nearly the entire chloroplast genome is transcribed, as established by a number of studies [Hotto et al., 2011, Lima and Smith, 2017, Smith, 2018]. This observation can be explained by a relative flexibility of the transcriptional process, which typically initiates from multiple promoter regions per gene and often exhibits ineffective termination. The resulting primary transcriptome is highly heterogeneous, including polycistronic mRNAs—an RNA that encodes several proteins—with a broad spectrum of start and end positions [Stern and Gruissem, 1987, Germain et al., 2011]. Evidence of this complexity is found in the chloroplast of *A. thaliana* which have been shown to possess over 200 distinct TSS [Castandet et al., 2019]. This is on average more than one per gene. Following transcription, the primary transcriptome is subject to a series of maturation steps, ultimately resulting in the formation of the mature RNA population.

2. a common unit of length for double-stranded nucleic acids

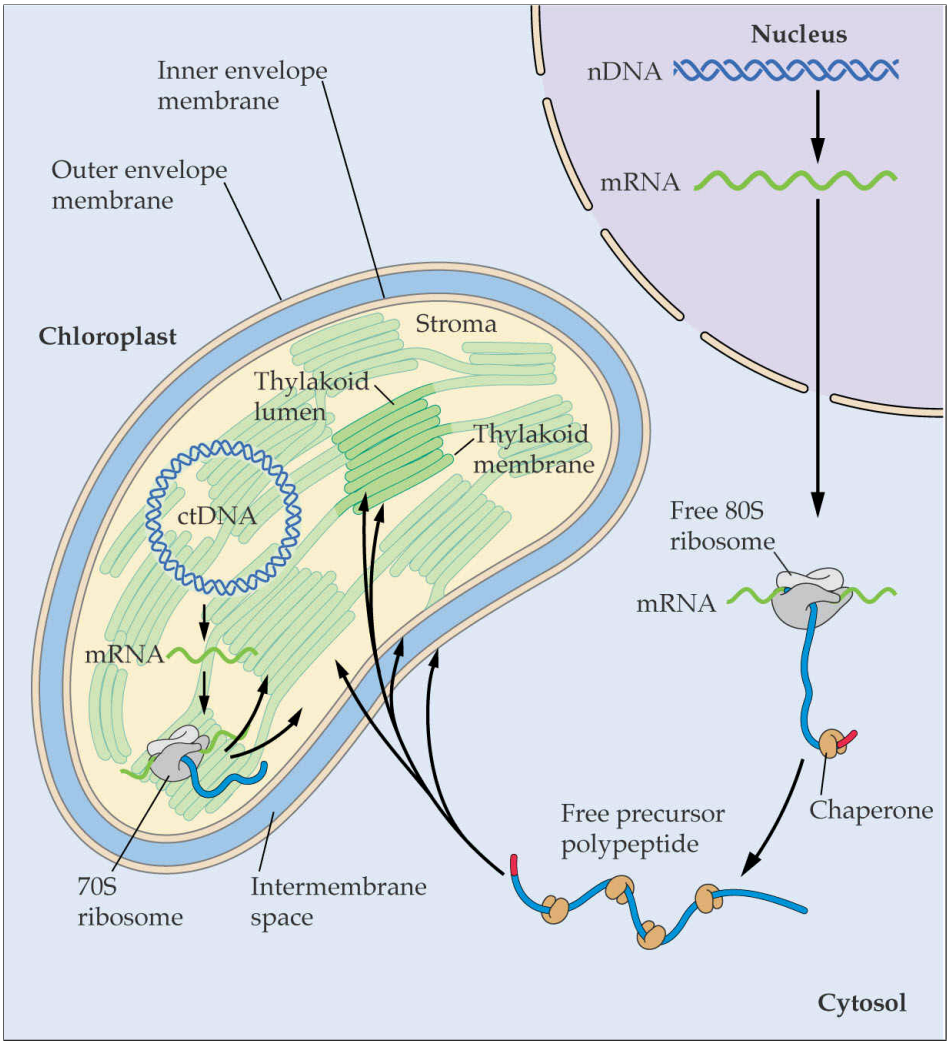


FIGURE 3.9 – The chloroplast. The chloroplast is an organelle measuring between 5 and 10 micrometers and located in the cytoplasm of plant cells. It is enveloped by two membranes—an outer and an inner membrane. Within the chloroplast lies a complex membrane network known as the thylakoids. The interior space of the thylakoids is called the lumen. One of the most important functions of these thylakoid membranes is to host the electron transport chain, which converts the energy of photons into chemical energy. Each chloroplast can contain thousands of thylakoids, which facilitate these light-dependent reactions of photosynthesis. Additionally, the chloroplast carries its own DNA (ctDNA) of about 1.5×10^4 base pairs and typically consisting of around 120 genes. This reduced genome is distinct from the cell's nuclear genome (nDNA). The chloroplast mRNA translation is conducted by bacterial-type 70S ribosomes. Interestingly, the chloroplast doesn't operate in isolation from the rest of the cell; it is interconnected with the nuclear gene expression system. Notably, numerous nucleus-encoded proteins are translated in the cytosol and imported into the chloroplast, where they control chloroplast gene expression. The figure is derived from Buchanan et al. [2015].

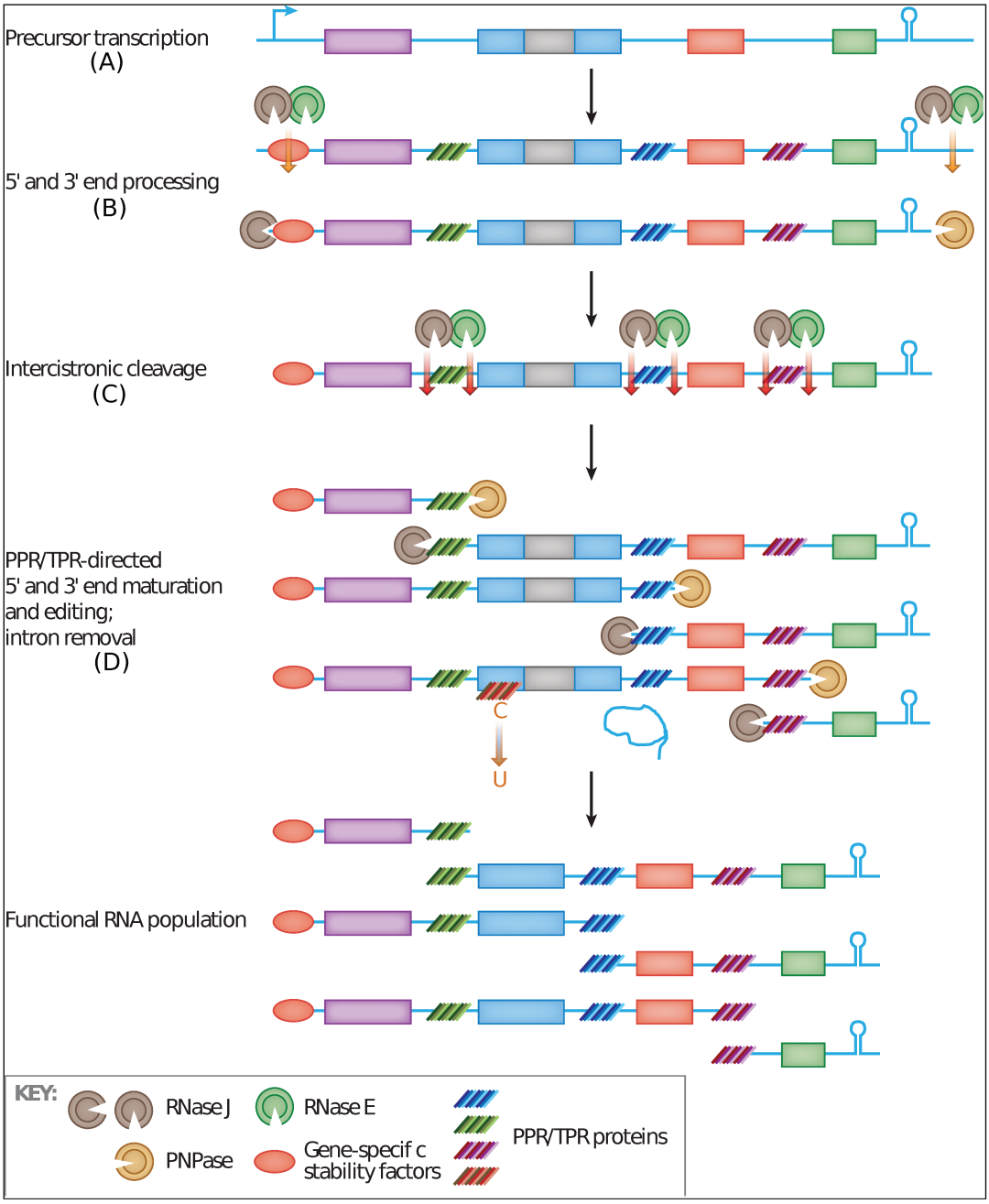


FIGURE 3.10 – Chloroplast RNA maturation. **(A)** A gene cluster that is initiated from a single TSS, indicated by a bent arrow, and concludes with a stem-loop-forming inverted repeat is depicted. Note that some gene clusters may contain several TSS (not shown here). There are different types of RBPs, specifically pentatricopeptide repeat and tetratricopeptide repeat (PPR/TPR) proteins, which are differentiated by color in the figure to indicate their unique binding sites. **(B & C & D)** The precursor transcript is subjected to intercistronic cleavage by endoribonucleases (RNase J and RNase E) and the 5' and 3' ends are digested by exoribonucleases (RNase J and PNPase). Certain RNA molecules are safeguarded from such nuclease attacks through their interactions with PPR/TPR or certain stability factors specific to genes. The 3' to 5' exoribonuclease activity of the PNPase can also be hindered by RNA secondary structures. **(D)** The transcripts are further edited at specific sites and the introns (grey rectangle) are spliced. The figure is derived from Stern et al. [2010].

3.3.3.2 Extensive maturations of chloroplast RNAs

The maturation of chloropast RNAs involves a set of events, including the formation of new ends, intron splicing, and site-specific editing. Significantly, these maturation events are primarily facilitated by ribonucleases [MacIntosh and Castandet, 2020] and a wide array of nucleus-encoded RNA-Binding Proteins (RBPs). For example, 185 RBPs are estimated to be present in the chloroplast of *A. thaliana* (more than one per gene). For comparison, the number of RBPs per gene in nucleus of *A. thaliana* is less than 0.05 [Small et al., 2023]. Among RBPs, the majority belong to the Pentatricopeptide Repeat (PPR) family [Lurin et al., 2004].

The processing of extremities. Transcripts are subjected to intercistronic cleavage by endoribonucleases, creating new 5' and 3' ends that can be further digested by exoribonucleases (Figure 3.10.B). Certain RNA molecules are safeguarded from such nuclease attacks through their interactions with proteins, specifically PPRs. Secondary structures may also perform a similar protective role [Germain et al., 2013, Barkan and Small, 2014]. The processing of these extremities considerably amplifies the complexity of the chloroplast transcriptome. For instance, 1628 processed 5' ends and 1299 3' ends were identified in Castandet et al. [2019]. For illustration, Figure 3.11 displays the positions of the 5' and 3' ends of each isoform originating from the polycistronic gene cluster *psbB-psbT-psbN-psbH-petB-petD*. Interestingly, the combination of 5' and 3' ends in this region significantly exceeds the number of genes.

The splicing of introns. In the chloroplast genome of *A. thaliana*, 20 introns are present : six in tRNAs—an RNA that carries an amino acid to the protein synthesizing machinery—and fourteen in mRNAs. The primary mechanism of splicing is cis-splicing, wherein the two exons separated by the spliced intron reside within the same RNA molecule (Figure 3.10.D). However, instances of trans-splicing do exist, where the exons to be joined are located on two separate RNA molecules [Choquet et al., 1988, Germain et al., 2013].

The single nucleotide editing. RNA editing in chloloroplast implies the transformation of C into U via deamination [Baudry, 2019]. In the context of *A. thaliana*, 43 editing sites have been found in chloroplast RNAs [Ruwe et al., 2013], although more are likely to exist. These sites can occur within coding or non-coding sequences. For instance, *ndhD* editing by PPR CRR4 restores a start codon (Figure 3.10.D) [Okuda et al., 2006].

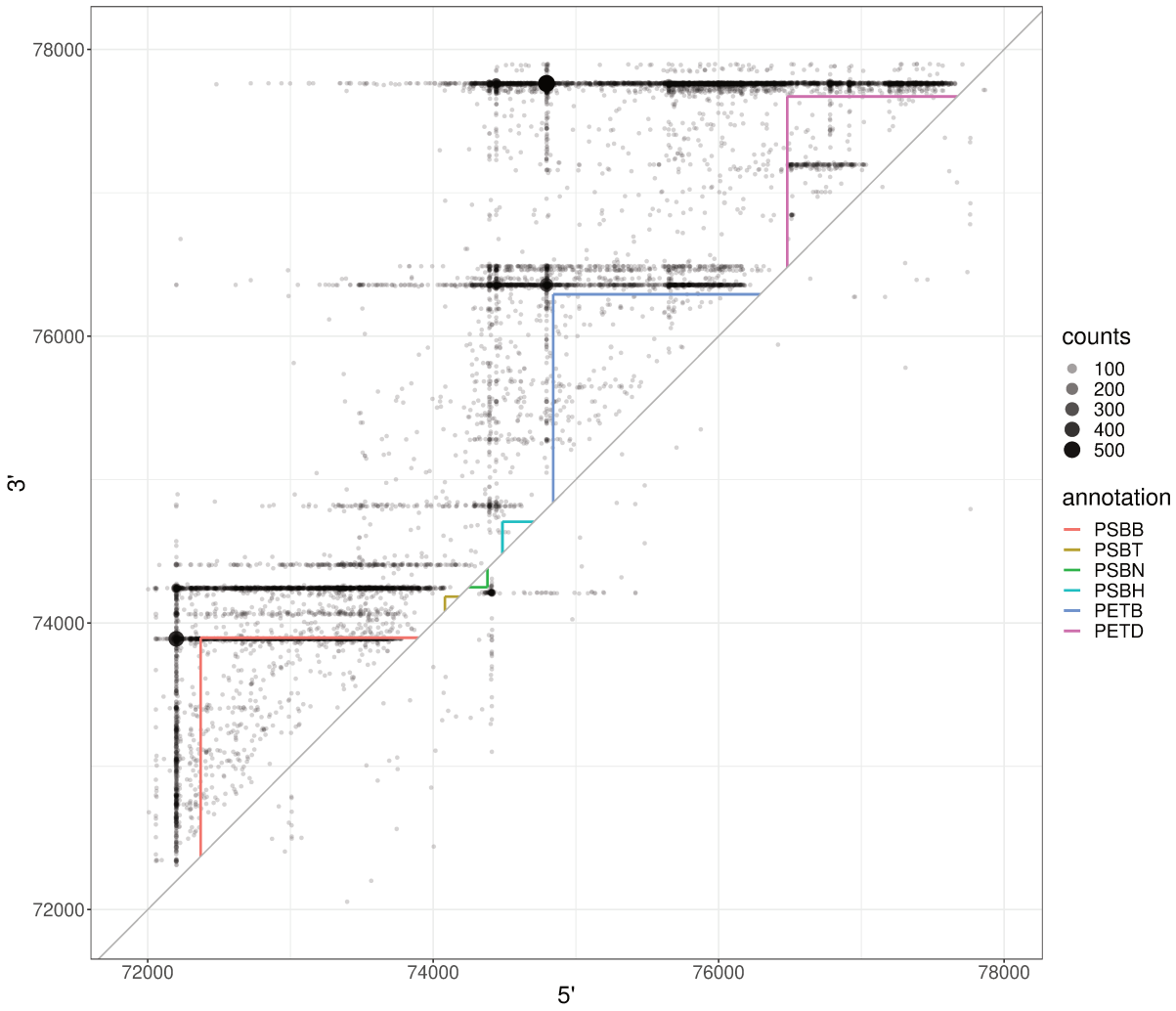


FIGURE 3.11 – Visualization of isoform extremities for the polycistronic gene cluster *psbB-psbT-psbN-psbH-petB-petD* in *A. thaliana* (wild type). The 3' (X-axis) and 5' (Y-axis) ends have been derived from the same long-read sequencing data that were utilized in Guilcher et al. [2021]. The figure was created using the custom-built vizExtremities R package, which is accessible at `https://github.com/aLiehrmann/vizExtremities`. This package employs the Shiny framework, offering an interactive method for visualizing the extremities of long-read sequencing data.

The coordination of maturation events. The question of whether one (or more) of the aforementioned chloroplast RNA maturation events is required for another maturation event to occur remains largely unexplored. Nevertheless, a handful of examples have been documented. For instance, it has been observed that the splicing of the *ndhA* intron is needed for editing the second exon of the same gene [Schmitz-Linneweber, 2001]. Additionally, more subtle coordinations have been identified, such as the concurrent loss of editing at the *atpF_12707* site and the reduction in splicing of the *atpF* transcript in the *aef1* mutant [Yap et al., 2015]. In the same vein, the *pnp1-1* mutant, marked by a decline in the trimming activity of transcripts at their 3' ends, also displays editing defects [Ruwe et al., 2013].

Chapter 4

Formalization of the biological question and proposal of a baseline model

In this chapter, I present the biological questions that I have investigated, along with the corresponding statistical problems, and the statistical models that I proposed to tackle these specific problems. My objective was to put forth straightforward models that would simplify data interpretation for biologists, and in turn, enhance interdisciplinary communication. Additionally, I sought to leverage existing methodologies whenever practical. To be specific, I have learned through a first experience on the detection of epigenetic marks (Section 4.2), and then confirmed by another experience on the detection of RNA regulations (Section 4.3), that simpler models, despite being mathematically unsatisfying, can be simultaneously (1) easier for non-specialists to understand, (2) easier to implement and calibrate, and (3) surprisingly efficient or even superior at addressing the biological question. Therefore, it is my opinion that such simpler models should be given priority. Furthermore, acknowledging that in the worst-case scenario these models may be less effective, they nonetheless serve a crucial role in substantiating the necessity to develop and implement more sophisticated models. This principle of parsimony, to which I fully subscribe, guided me throughout my doctoral research, particularly when working on the detection of RNA regulations (Section 4.3) and co-maturations (Section 4.4).

4.1 Chapter summary at glance

1. In Section 4.2, I discuss the problem of detecting epigenetic marks, starting from the biological objective (Biological question 1) and proceeding to the formulation of the respective statistical problem (Statistical problem 1). A comprehensive review of the state-of-the-art methods, recently devised to address the statistical issue, is subsequently delivered. Finally I introduce a baseline model (Baseline model 1), which is purely based on the conventional principles of signal transformation and segmentation, developed during the 1940s and 1980s respectively. The effectiveness of this baseline solution is observed to be as accurate, if not superior, to the recent advancements as elucidated in Liehrmann et al. [2021].
2. In Section 4.3, mirroring the structure from Section 4.2, I delve into the problem of de-

tecting RNA regulations. Finally, I recall the standard changepoints model previously employed in the detection of epigenetic marks. Once more, the standard changepoints model is found to outperform state-of-the-art methods in the detection of RNA regulations, as detailed in Liehrmann et al. [2023].

3. In Section 4.4, I briefly present the problem of studying the coordination of RNA events, a problem which I co-supervised two interns on during the first and second year of my thesis.

4.2 Detection of epigenetic marks

4.2.1 Foreword

During my Master's research internship and at the beginning of my thesis, I worked on the detection of epigenetic marks in data obtained from Chromatin Immunoprecipitation followed by Sequencing (ChIP-Seq). These epigenetic marks, pivotal in a multitude of essential biological processes including gene transcription, modulate DNA region accessibility to regulatory proteins. As a result, in addition to being a valuable example of high-throughput sequencing data analysis, studying epigenetic marks in ChIP-Seq data helped me to understand gene expression at an early stage. I tackled this challenge by understanding the biological objective of the analysis in order to propose a statistically relevant model.

4.2.2 Biological goal

In response to environmental stress or during developmental stages, the accessibility of various DNA regions in eukaryotic organisms can undergo significant transformations [Gao et al., 2010, Widiez et al., 2014, Donkin and Barrès, 2018, Iwagawa and Watanabe, 2019]. This process is partially facilitated by modifications to the tails of histones—proteins associated with DNA. These modifications can locally alter the chemical interactions between DNA and regulatory proteins, influencing gene expression.

Histone modifications are diverse, encompassing methylation, acetylation, ubiquitination, and phosphorylation, among others. A notable modification is the lysine methylation on the N-terminal tail of histone H3, which has been the subject of extensive study. For instance, the trimethylation at the 4^{th} lysine residue of histone H3 (H3K4me3) is a modification strongly associated with TSS. As such, H3K4me3 is often considered a reliable marker for TSS [Lloret-Llinares et al., 2012, Benayoun et al., 2014].

Another significant modification is the trimethylation at the 36^{th} lysine residue of histone H3 (H3K36me3). This modification typically occurs within the body of actively transcribed genes, where it is involved in finely regulated processes like RNA elongation and splicing. For example, variations in exon inclusion/exclusion have been linked with intragenic H3K36me3 levels in gene bodies. This is facilitated by the recruitment of H3K36me3 reader proteins (such as *MRG15* and

ZMYND11), which directly modulate the activity of splicing factors [Zhang et al., 2006, Kim et al., 2011, Guo et al., 2014].

In this context, to study either the regulation of transcription initiation or the regulation of RNA splicing, biologists can perform a ChIP-Seq experiment [Park, 2009] (Figure 4.1) with the aim of knowing :

Biological question 1

☞ Which regions of the genome are enriched with specific epigenetic marks ?

Indeed, in the results of this experiment, regions enriched in epigenetic marks are characterized by a higher density of aligned reads than in non-enriched regions. Biologists curious enough to visualize the ChIP-Seq data they generate often associate the enriched regions with peaks, in reference to their shapes (Figure 4.2). The reads can be counted at each genomic position, and this results in a series of non-negative integer count data ordered along the genome, hereafter called coverage profile. Please be aware that additional heuristics for calculating coverage profiles exists, as outlined in Note S1 of [Liehrmann et al., 2023].

Bearing in mind that ChIP-Seq reads are a biased proxy of epigenetic marks [Diaz et al., 2012], we can reformulate Biological question 1 as :

Statistical problem 1

☞ Where are the start and end of each peak in the coverage profile ?

4.2.3 Statistical model for peak calling

4.2.3.1 Survey of peak calling methods for epigenetic marks enrichment

In the past few years, several teams have developed methods to provide practical solutions to Statistical problem 1 [Fejes et al., 2008, Spyrou et al., 2009, Zang et al., 2009, Rozowsky et al., 2009, Xu et al., 2010, Rashid et al., 2011, Xing et al., 2012, Harmanci et al., 2014]. While a comprehensive review of all the methods would be beyond the scope of this manuscript given the vast number of techniques developed, I will focus on highlighting a selected few that have made considerable strides in enhancing the detection accuracy of one or both epigenetic marks H3K4me3 and H3K36me3.

MACS. MACS [Zhang et al., 2008] is a widely recognized method in the field of bioinformatics, garnering over 13,000 citations, evidence to its credibility and usefulness in the scientific community. It is particularly good at pinpointing sharp peaks that correspond to epigenetic marks such as H3K4me3. MACS implements a two-stage procedure. At its core, MACS operates a one-sided exact Poisson test within a sliding, constant-length window across the genomic landscape. The test accommodates local biases in the genome, including factors such as chromatin structure, GC

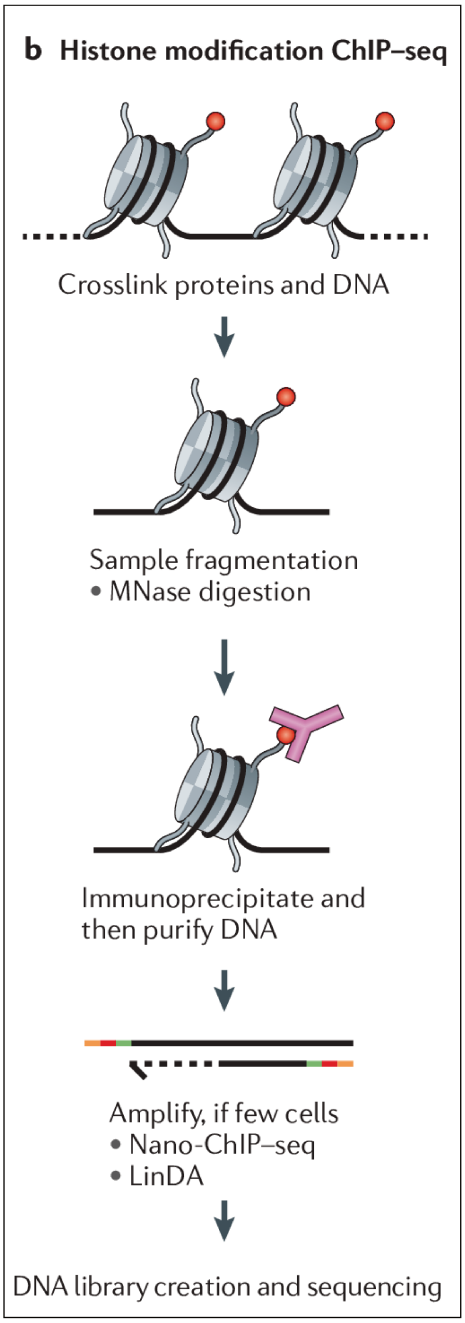


FIGURE 4.1 – ChIP-Seq experiment in a nutshell. The process of preparing an ChIP-Seq library begins with cross-linking, mainly with formaldehyde, to secure the interactions between the DNA and proteins. The structure formed by the DNA and proteins, known as chromatin, subsequently undergoes fragmentation. This is followed by the central process of the protocol where the sheared chromatin is incubated with an antibody that targets the protein of interest. The DNA fragments bound to the target protein are then separated from non-specific DNA by centrifugation. Subsequently, the cross-links between DNA and protein are reversed, which allows the DNA to be separated and purified through an extraction process. Then, adaptors are ligated to the ends of the DNA fragments. The adapted DNA fragments are amplified using PCR. Finally, the amplified DNA fragments are sequenced. The figure is derived from Furey [2012].

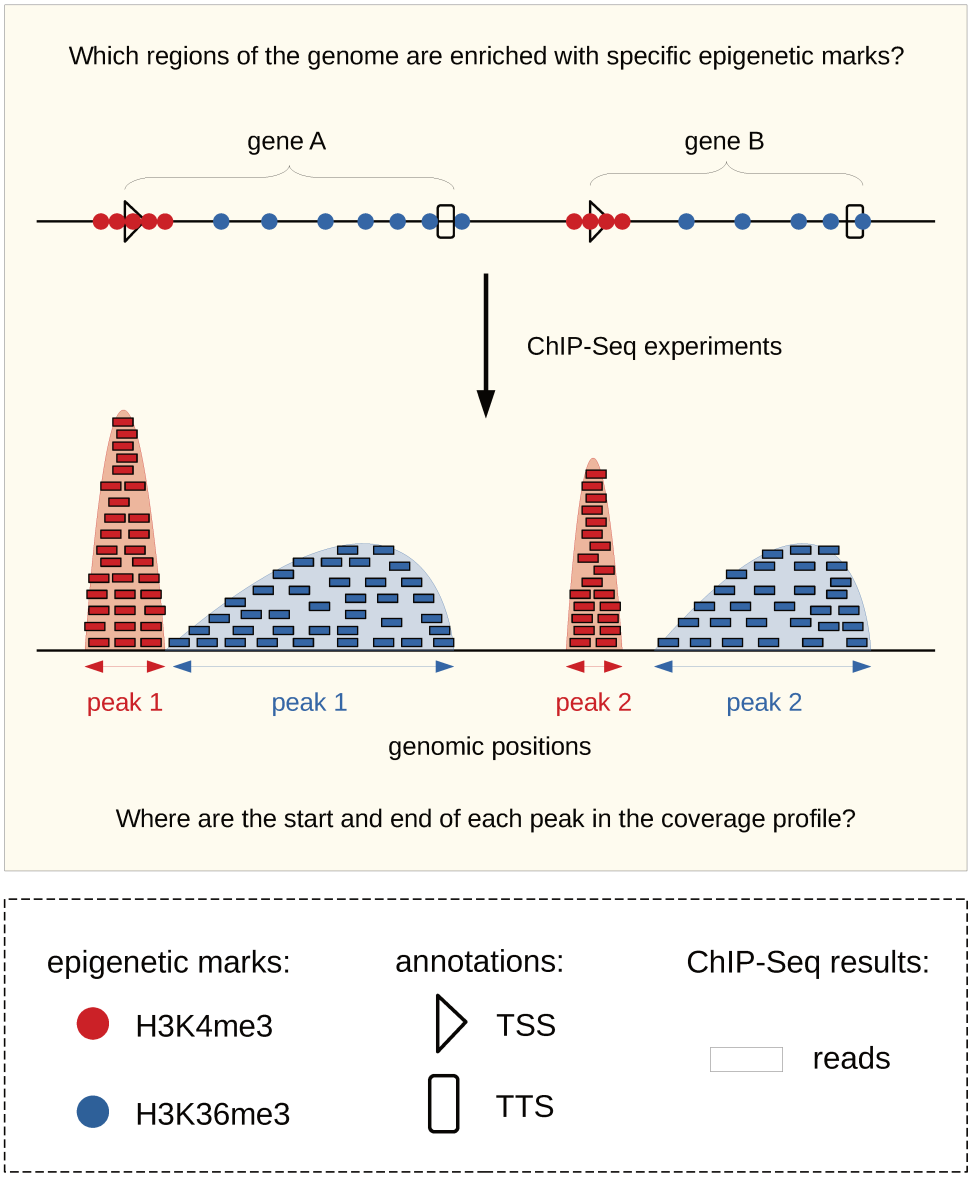


FIGURE 4.2 – Results of two separate ChIP-Seq experiments focusing on H3K4me3 and H3K36me3 histone modifications. The TSS of both gene A and gene B are enriched of H3K4me3, signifying its role in regulating transcription initiation. When a ChIP-Seq experiment specifically targets H3K4me3, it results in a significant density of reads that mapped directly to the TSS of these genes. We observe a sharp peak at TSS levels in the coverage profile. Similarly, the body of both gene A and gene B are enriched of H3K36me3, signifying its role in transcription elongation and RNA maturation. At the end of a ChIP-Seq experiment targeting H3K36me3, we notice a broad peak within the gene body in the coverage profile. In practical scenarios we can use the results from ChIP-Seq experiments as a viable proxy of regions enriched in epigenetic marks. This involves scanning for peaks throughout the genome.

content and copy number variations. Subsequently, MACS applies the Benjamini-Hochberg procedure to correct p-values for multiple comparisons. A region is deemed to demonstrate significant enrichment if the corrected p-value falls below a user-defined threshold. Secondly, MACS merges any overlapping significant peaks. Importantly, MACS provides at least five user-adjustable parameters that influence both the position and the number of peaks identified.

HMCan. HMCan [Ashoor et al., 2013] is another notable method that shows good performances in detecting broad peaks that correspond to epigenetic marks such as H3K36me3. Therefore, its functionality complements MACS. HMCan also implements a two-stage procedure. First, similar to MACS, it conducts a one-sided exact Poisson test. Secondly, it uses regions identified as significantly enriched to estimate the parameters of a two-state (peak, not-peak) Hidden Markov Model (HMM). Following this estimation, an iterative Viterbi algorithm is applied to the coverage profile to infer the location of peaks across the genome. Importantly, HMCan adjusts the coverage profile to account for GC content and copy number variations. Similar to MACS, HMCan provides at least five user-adjustable parameters that influence both the position and the number of peaks identified.

Constrained segmentation. Segmentation analysis, in simple terms, is the process of pinpointing locations where there is a change, also called changepoint, in statistical properties of the data. Peak calling can naturally be thought of as a specialized form of this process. It involves identifying multiple changepoints within a coverage profile, but with an added nuance : there is a directional constraint on these changes. This means that if an upward change is observed from a genomic region of sparse coverage to an adjacent region with substantial coverage, it is inevitably followed by a downward change, and vice versa (Up-Down). The Figure 4.3.A provides a comprehensive schematic illustration of the Up-Down rules. Additionally, a mathematical definition can be found in Equation 2 of Liehrmann et al. [2021].

Building upon a succession of previous studies within the Gaussian homoscedastic framework [Auger and Lawrence, 1989, Rigaill, 2015, Maidstone et al., 2016] and concurrently extended to the Poisson and negative binomial[1] cases [Cleynen and Lebarbier, 2014a], a segmentation model that encapsulates the Up-Down rules was first introduced by [Hocking et al., 2015] for the Poisson case. In the same study, the authors introduced a heuristic for estimating the model's parameters (including the start and end position of the peaks). It is a heuristic in the sense that it is not guaranteed to find the maximum likelihood estimator. A few years later, the General Functional Pruning Optimal Partitioning (GFPOP) method, an exact segmentation algorithm, was developed to address this limitation. GFPOP is available in the PeakSegDisk R package [Hocking et al., 2022] as well as the gfpop R package [Runge et al., 2023]. The Up-Down model has demonstrated slightly higher accuracy compared to MACS and HMCan on

1. The Gaussian, Poisson, and negative binomial distributions are distinct types of statistical noises utilized to represent unexplained variability within the data.

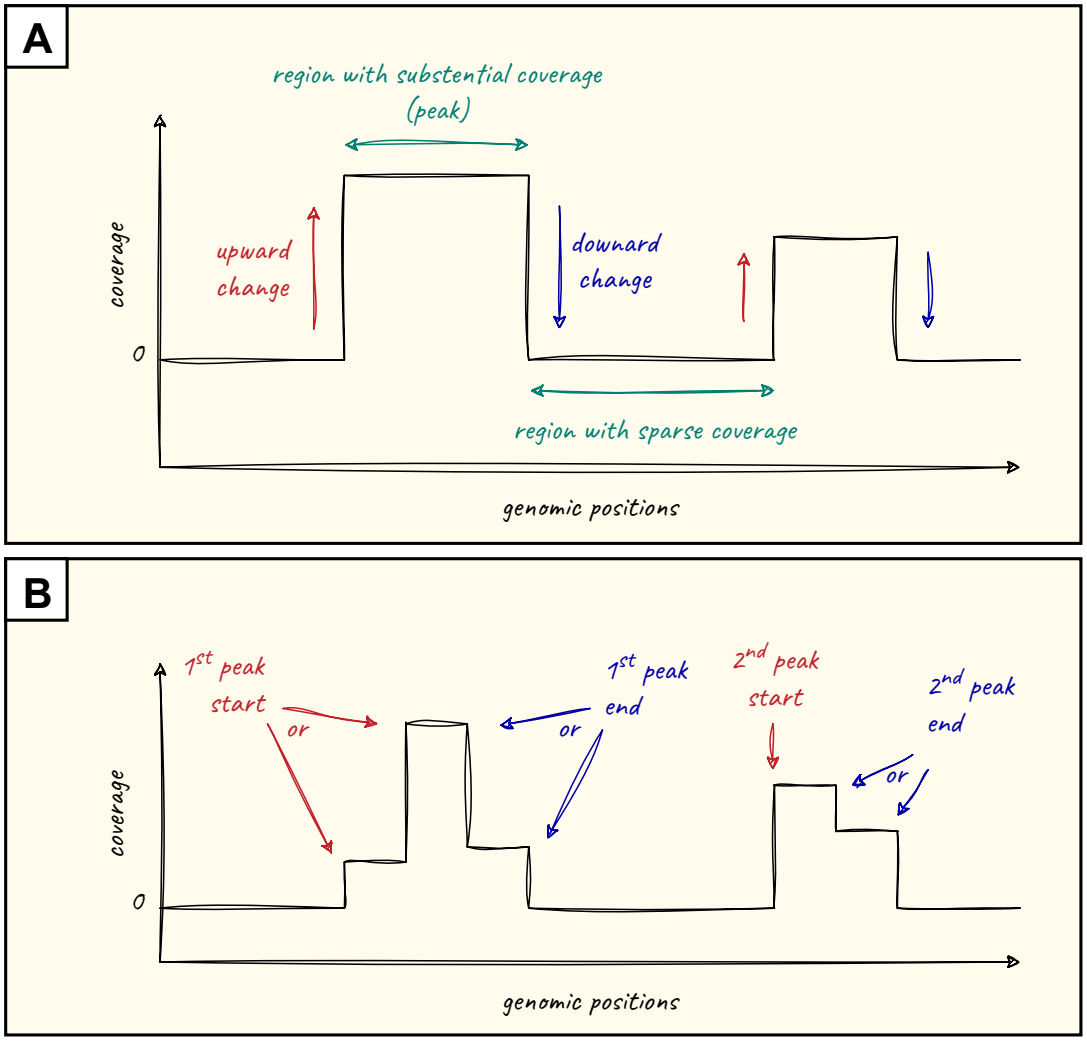


FIGURE 4.3 – Changepoint models for peak calling. (A) Schematic illustration of the directional rules on the changes incorporated within the constrained segmentation model (B) Schematic illustration of the standard changepoints model (the directional rules are dropout). In order to be interpretable in terms of peaks, one will have to define a post-processing rule which choose within each sucessive increases the start of the peak, and within each sucessive decreases the end of the peak.

H3K36me3 and H3K4me3 epigenetic marks, respectively [Hocking et al., 2020]. Furthermore, the number of peaks identified by GFPOP is governed by one parameter, thereby simplifying both calibration. Moreover, the number of peaks is a decreasing function of the parameter value, which substantially simplifies its interpretation.

Continuing this line of research, we presented an Up-Down segmentation model for the negative binomial case in Liehrmann et al. [2021]. This model was designed to accommodate a higher level of data variability than expected by the Poisson model (Figure 4 of Liehrmann et al. [2021]), aiming to boost the method's accuracy. Unfortunately, we did not observe this expected improvement (Figure 6 of Liehrmann et al. [2021]).

The Up-Down model is certainly a useful model that required several years of development and contributed to the enhancement of epigenetic mark detection. However, it is not without its shortcomings. Specifically, upon analyzing the shapes of peaks in coverage profiles, it becomes evident that the background noise and peak tops are sometimes divided by one or more subtle variations (Figures 1 and 2 of Liehrmann et al. [2021]). The Up-Down segmentation model cannot capture these subtle changes while a segmentation model without the directional constraint—and thus making more parsimonious assumptions—should be. This problem, while significantly amplified by the Up-Down model, is not exclusive to it. Indeed, any approach that solely focuses on identifying a start and an end (such as MACS or HMCan) will invariably make the same mistake. It is my opinion that, by articulating these assumptions through a mathematical model, we not only bring their limitations into sharp relief, but also open avenues for questioning and re-evaluating them. In essence, *always model and regularly doubt.*

4.2.3.2 Establishing a baseline model for peak calling

The standard changepoints model. In Liehrmann et al. [2021], we compared the recently developed methods to the standard changepoints model : a deterministic piecewise constant function with an additional homoscedastic Gaussian noise [Auger and Lawrence, 1989] [2]. After making minor adjustments as outlined below, we found that the standard changepoints model, inferred by minimizing a penalized Least Squares Criterion (LSC) [3], was equally or even more accurate than these methods (Figure 6 of [Liehrmann et al., 2021]).

The penalized LSC mentioned above can be swiftly optimized via the Functional Pruning Optimal Partitioning (FPOP) algorithm [Maidstone et al., 2016] [4]. FPOP's computation time is log-linear or linear, relative to the length of the signal, when there are few or many changepoints, respectively. This allows to segment 10^7 datapoints in less than 10 seconds, thereby matching the speed of linear heuristics like MACS. Furthermore, similar to GFPOP, the number of changepoints identified by FPOP is governed by one parameter, and is a decreasing function of this parameter value. It should be noted that, contrary to heuristics like MACS and HMCan,

2. see Equation (5.1) for a mathematical definition
3. see Equation (5.5) for a mathematical definition
4. see Section 5.4.2 for a review of the key elements of FPOP

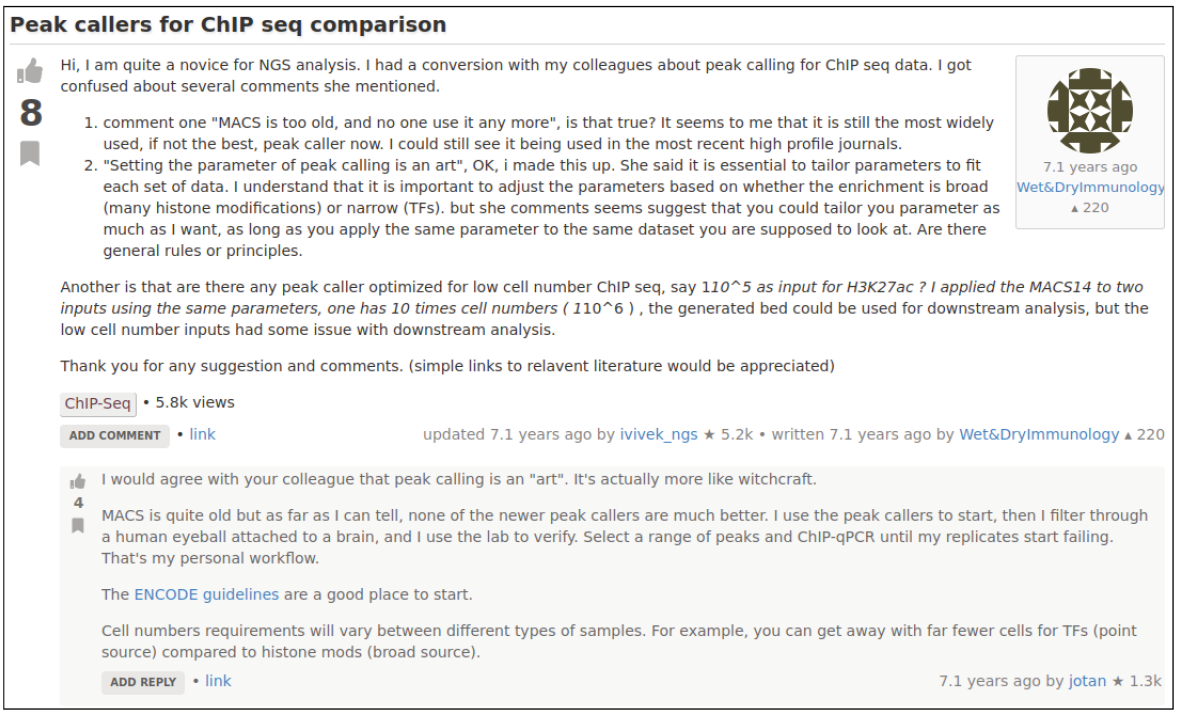


FIGURE 4.4 – Peak calling is an "art" https://www.biostars.org/p/190812/ (Biostars).

the standard changepoints model offers well studied statistical guarantees [Yao and Au, 1989, Garreau and Arlot, 2018], and has proven versatile, finding use in other applications like in the detection of DNA copy number variations [Picard et al., 2011].

Minor adjustments. Firstly, to approximately variance-stabilize the coverage profile, we used the Anscombe transformation as a preprocessing step [Anscombe, 1948] (Figure 4 of Liehrmann et al. [2021]). Secondly, the segmentation output from the standard changepoints model does not lend itself to a straightforward interpretation in terms of peaks (Figure 4.3.B). We proposed a rather natural post-processing rule, hereafter called max jump, to predict the start and end of peaks. Specifically, within successive increases and decreases, we select the upward and downward changes, respectively, that exhibit the largest mean-difference (Figure 2 of Liehrmann et al. [2021]).

Applying the Ockham's Razor. The two-stage peak calling procedures utilized in MACS and HMcan are influenced by a variety of parameters. Assessing the effects of tweaking these parameters—either individually or in combination—on the number and positions of detected peaks is not straightforward. However, as outlined in "ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia" [Landt et al., 2012], the composition of the final peak list is profoundly shaped by the specific parameter settings used and these parameters should be fine-tuned to suit each individual dataset.

Figure 4.4 illustrates this complexity through a snapshot of a discussion from the Biostars bioinformatics forum. The conversation occurs between a novice MACS tool user and a more

FIGURE 4.5 – Illustration from a manuscript featuring William of Ockham (*Summa totius Logicae*, 1341). William of Ockham (1287-1347), was an esteemed English Franciscan friar and a scholarly theologian. He made significant contributions as a philosopher during the medieval. His lasting fame, mainly as an eminent logician, rests largely on a philosophical principle widely attributed to him, known as *Ockham's Razor*. This razor is metaphorically employed to trim down superfluous assumptions or dissect similar conclusions when distinguishing between two hypotheses, thereby emphasizing simplicity and parsimony in reasoning. Today, the principle of *Occam's Razor* is frequently used across various fields as a heuristic guide to decision-making, problem-solving, and hypothesis testing [Anderson and Burnham, 2004, Gigerenzer and Gaissmaier, 2011].

seasoned user. The advanced user's response, with a hint of levity, aptly encapsulates the predicament : *"... peak calling is an "art". It's actually more like witchcraft."*

Somewhat at odds with this literature, the standard changepoints model, with minor adjustments, saves assumptions by proposing a single (interpretable) adjustable parameter. Additionally, it forgoes the directional rules set in the Up-Down model, instead suggesting a post-processing rule. Applying the max jump post-processing rule is arguably close to what a specialist's eye does when annotating peaks by hand. Despite fewer assumptions, the standard changepoints model demonstrates accuracy equal to, if not better than, its competitors.

In line with the principle of *Ockham's Razor*—the simplest sufficient assumptions should be preferred (Figure 4.5)—, I advocate using

Baseline model 1

☞ the standard changepoints model with minor adjustments

to identify peaks in ChIP-Seq data (Statistical question 1).

4.3 Detection of RNA regulations

4.3.1 Biological goal

As mentioned in the section discussing RNA metabolism, numerous studies have highlighted the occurrence of extensive alterations in the patterns of transcription initiation and RNA maturation processes during development, under stress conditions, or in diseases.

In this context, to study RNA regulations, biologists classically design an RNA-Seq experiment including several biological replicates from a condition of interest and a control (e.g. a diseased and healthy tissue). Through this experiment, a part of the biologists seeks to know :

Biological question 2

☞ What are the transcriptome differences between the two biological conditions ?

Indeed, in the results of a typical RNA-Seq, discarding any normalization issues due to different library sizes [Abbas-Aghababazadeh et al., 2018], the differences in RNA maturation or ATI lead to local variations in read density along the genome between the two biological conditions (Figure 4.6). Throughout this manuscript, I will refer to these local variations as Differentially Expressed Regions (DERs). Consequently, if one of these DERs overlaps with an annotation (e.g. the first intron), an event-level analysis allows us to detect it (Strategy 1). It is through this annotation that we can also formulate a hypothesis about the underlying regulatory mechanism (e.g., "we observe an accumulation of the first intron in biological condition A, thus the biological processes involved in the splicing of this intron seem to differ between the compared biological conditions").

Relying solely on annotations is a baseline which is biologically unsatisfactory for two reasons : (1) you do not look outside of these annotations, (2) the limits of these annotations can be ill-suited, making both detection and interpretation impossible (Figure 4.7). An alternative approach involves identifying DERs along the genome (Strategy 1) without relying on annotations (Strategy 2). This data-driven approach addresses the detection issue but does not readily provide an interpretation of the underlying regulatory mechanism. We can reformulate Biological question 2 as :

Statistical problem 2

☞ Which regions are differentially expressed along the genome ?

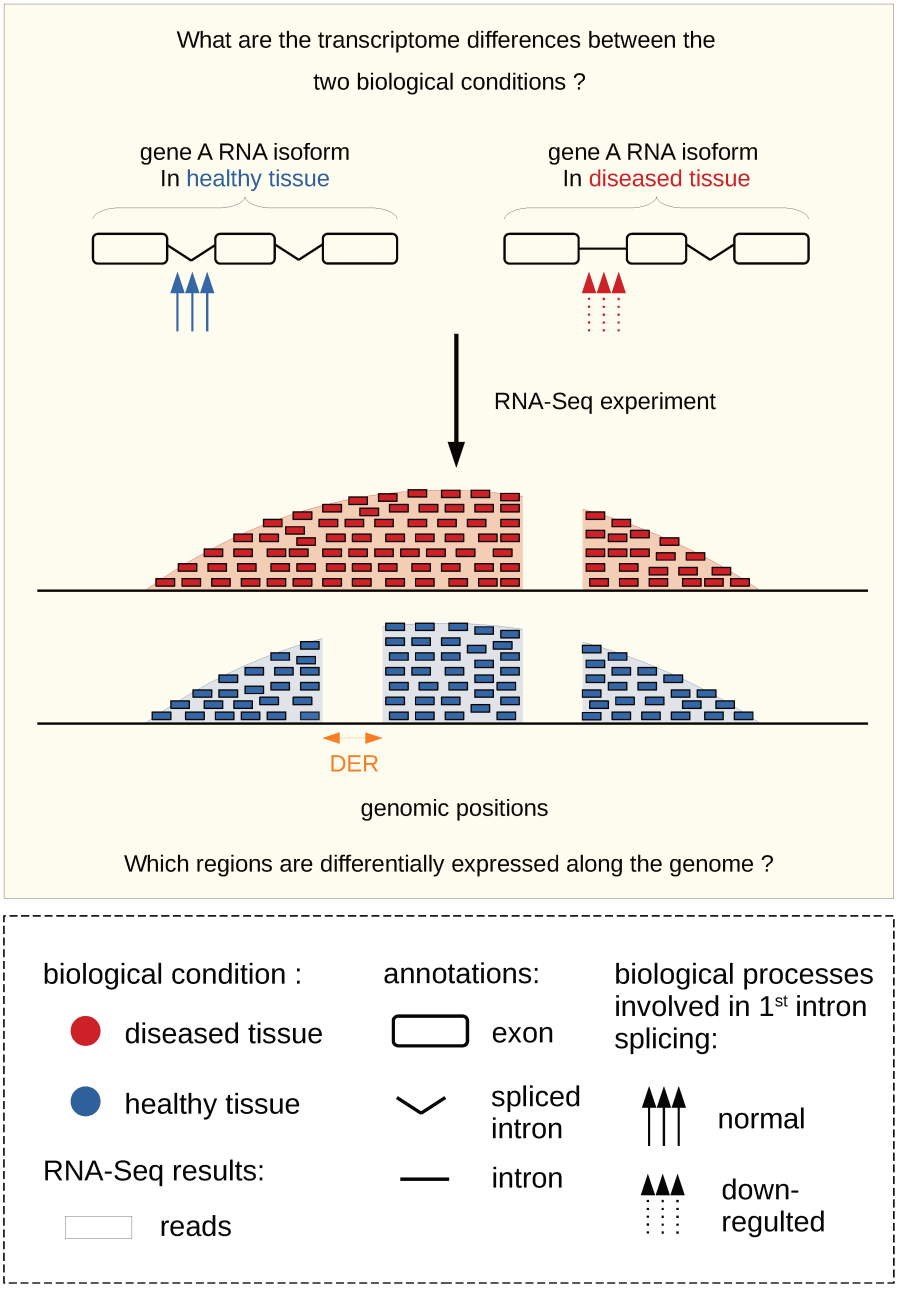


FIGURE 4.6 – Results of an RNA-Seq experiment on a diseased tissue compare to an healthy tissue. In the diseased tissue, the biological processes involved in the splicing of the first intron of gene A are down-regulated, resulting in an intron retention within the produced transcripts. In the healthy tissue, the splicing is done correctly. When an RNA-Seq experiment is done on both biological conditions, discarding any normalization issue, it results in a local variation of the density of reads at the level of the first intron. In practical scenarios, we can use results from an RNA-Seq experiment as viable proxy of RNA regulations.

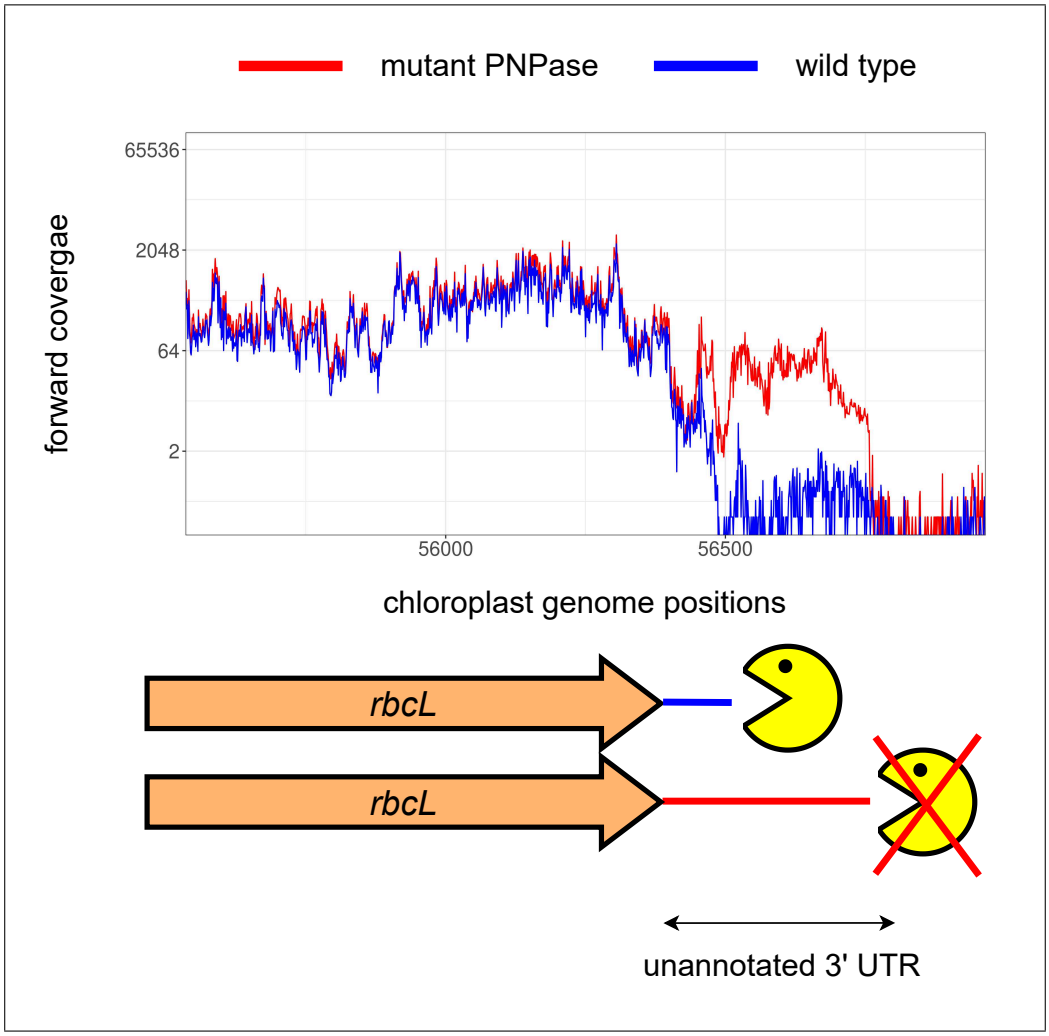


FIGURE 4.7 – The annotation of the *rbcL* gene does not capture the 3' extension observed in the *A. thaliana* mutant deficient in PNPase activity. The coverage profile overlapping the 3' region of the *rbcL* gene on the forward strand is presented for both a wild type and a PNPase-deficient mutant of *A. thaliana*. Examining the coverage within the gene annotation, represented by an orange arrow, reveals no significant expression difference between the mutant and the wild type. However, upon inspecting the area adjacent to the annotation, which corresponds to the 3' unannotated UTR, we observe a drop in coverage slightly before the mutant when compared to the wild type. This suggests an extension of *rbcL* transcripts in the mutant, consistent with the role of PNPase in 3' ends transcript trimming. This extension of transcripts cannot be evidenced within the scope of analyzing DERs within gene annotations.

4.3.2 Statistical model for transcriptome-wide detection of expression differences

4.3.2.1 Survey of methods for transcriptome-wide detection of expression differences

Similar to the detection of epigenetic marks problem, numerous research teams have advanced innovative and practical solutions to address Statistical Problem 2. These methods, occasionally referred to as "identify-then-annotate" tools [Frazee et al., 2014], approach the task of identifying DERs in two distinct steps. The first step involves summarizing the coverage profiles (one per replicate) from an RNA-Seq experiment into a single signal and using it to delineate the boundaries of candidate DERs along the genome. The primary differentiating factors among these methods lies in the specific type of signal they segment and their specific segmentation approach (Figure 1 of [Liehrmann et al., 2023]). The subsequent step involves a statistical evaluation of expression differences within the newly defined regions. Most of these methods utilize the negative binomial GLM of DESeq2, originally conceived for gene counts [Love et al., 2014] [5], but found to be reasonably effective for event counts as well [Anders et al., 2012]. I will now describe the candidate DERs identification stage for two identify-then-annotate methods that have opted for distinct modeling approaches.

derfinder RL. The core functionality of derfinder RL [Collado-Torres et al., 2016] relies on a threshold-based heuristic approach to detect candidate DERs in the coverage. This process unfolds in several stages. Initially, derfinder RL normalizes the coverage profiles with respect to sample-specific library size. Following this, for each genomic position, it calculates the mean of these normalized coverage profiles, thereby creating an average coverage profile. Subsequently, a user-determined cutoff, which likely influences both the position and number of DERs, is applied to this average coverage profile. Any contiguous sequence of bases that exhibits an expression exceeding this cutoff is designated as a candidate DERs.

srnadiff. srnadiff [Zytnicki and González, 2021] combines the candidate DERs identified by two distinct approaches : (srnadiff IR) a threshold-based heuristic applied to the per-base $\log_2$ fold-change ($\log_2$-FC)—the difference of expression on the logarithmic scale, and (srnadiff HMM) a two-state Hidden Markov Model used on the per-base p-values. To begin, like derfinder RL, srnadiff normalizes the coverage profiles with respect to sample-specific library size. In the IR procedure, srnadiff subsequently calculates the average coverage profile for each biological condition, leading to the derivation of the per-base $\log_2$-FC profile. From this profile, the IR procedure pinpoints any regions where the absolute $\log_2$-FC surpasses a user-defined threshold. Finally, srnadiff merges closely located candidate DERs that exhibit similar $\log_2$-FC. In the HMM procedure, srnadiff executes a DESeq2 analysis, extracting the Wald test p-value for each genomic

5. see Section 6.3 for a general introduction

position displaying an expression that exceeds a user-defined threshold. Following this, srnadiff constructs a two-state HMM (differentially expressed, not-differentially expressed) with user-determined parameters. Subsequently, the Viterbi algorithm is applied to the per-base p-values to infer the location of candidate DERs. Ultimately, the candidate regions of srnadiff IR and srnadiff HMM are merged using rules based on p-value and overlap. It is worth to note that, srnadiff provides at least six user-adjustable parameters that influence both the position and the number of DERs.

4.3.2.2 Establishing a baseline for transcriptome-wide detection of expression differences

The standard changepoints model (again). In Liehrmann et al. [2023], we assessed the recently developed methods mentioned above in contrast to the standard changepoints model, with changepoints observed in the per-base $\log_2$-FC [6]. After locating the changepoints with FPOP and assessing the resulting candidate DERs with DESeq2, we noted that, once again, the standard changepoints model was more accurate than state-of-the-art methods. For this evaluation, we incorporated biological labels that reflected molecularly validated accumulations of RNA fragments in two mutants of *A. thaliana* for chlorolastic ribonucleases [7].

Moreover, the standard changepoints model was better in accurately capturing the differential landscape. Specifically, it exhibited two strengths : (i) it demonstrated a reduced propensity to merge regions that likely arise from distinct regulatory mechanisms, and (ii) it showed a diminished tendency to fragment non-DERs, thus curbing the unnecessary inflation of regions for testing [8].

Finally, the outcome of the per-base $\log_2$-FC segmentation using FPOP is largely independent of coverage normalization [9]. Consequently, unlike in many other methods, normalization is not a compulsory pre-processing step to find candidate DERs.

Applying the Ockham's Razor (again). The derfinder RL method only identifies 4 out of the 23 biological labels used in [Liehrmann et al., 2023]. This low accuracy could arguably be attributed to the chosen approach of segmenting the mean of coverages using a threshold-based heuristic. Such an approach could potentially merge DERs and non-DERs, which may in turn mask DERs. For instance, this phenomenon is illustrated in Figure 5.C of Liehrmann et al. [2023]. Under these circumstances, defining derfinder RL as a baseline would likely be of limited interest.

On the other hand, srnadiff manages to detect 20 out of 23 labels, thus seemingly better at identifying expression differences than derfinder RL, albeit less so than the standard changepoints model (which finds all the labels). However, srnadiff operates under multiple assumptions

6. see *Differential transcription profile* section of Liehrmann et al. [2023] for a mathematical definition
7. see *DiffSegR improves the search for DERs* section of Liehrmann et al. [2023]
8. see *DiffSegR better captures the differential landscape* section of Liehrmann et al. [2023]
9. see *Normalization* section of Liehrmann et al. [2023]

that lead to a complex procedure for identifying candidate DERs, which involves several parameters that need calibration. Indeed, as previously detailed, srnadiff merges the candidate DERs identified by two distinct segmentation approaches, and the results are influenced by at least six user-specified parameters.

Contrary to the prevailing literature, the standard changepoint model provides a parsimonious solution without compromising accuracy. Moreover, this model is extensively justified and supported by a rich body of statistical and applied literature, as outlined in Chapter 5.

In this context, harking back to the principle of *Ockham's Razor*, I advocate using

Baseline model 2

☞ the standard changepoints model after transforming the coverage profiles in the per-base $\log_2$-FC

and subsequently testing the identified candidate DERs with DESeq2 (Statistical problem 2).

4.4 A few words on the coordination of RNA events

4.4.1 From a deterministic to a probabilistic view of dependence

A relationship in which one or more RNA events are absolutely required for another RNA event to occur (as mentioned in Section 3.3.3.2) is conceptually practical and experimentally easy to validate. However, this rather deterministic perspective of dependence is rarely corroborated by data. For instance, no such relationship was identified in our study on the coordination of chloroplast RNA maturation events [Guilcher et al., 2021]. Statistical dependence allows, in a set of random experimental data, to discern significant outcomes in a variety of scenarios including the case discussed below (Figure 4.8.C) and others that are less clear-cut (Figure 4.8.B).

4.4.2 Transcriptome-wide detection of co-maturations

Together with Marine Guilcher, Benoît Castandet, Guillem Rigaill, and Etienne Delannoy, as well as two students—Chloé Seyman, a bachelor's student, and Benjamin Vacus, a master's student, whom I had the pleasure of co-supervising for periods of three and six months respectively, we explored the interplay of biological processes involved in RNA events.

Intuitively, examining the statistical dependence of the K events that may occur along an RNA is similar to studying the 2^K isoforms. As we have seen in Section 3.2.2, this is a substantially complex task. To mitigate this complexity, we turned our attention on the dependence of the $\binom{K}{2}$ annotated event pairs—or what we refer to as co-maturations—as proposed in Strategy 1, and focusing for now on intron splicing and editing sites. Notably, we applied this strategy in Guilcher et al. [2021].

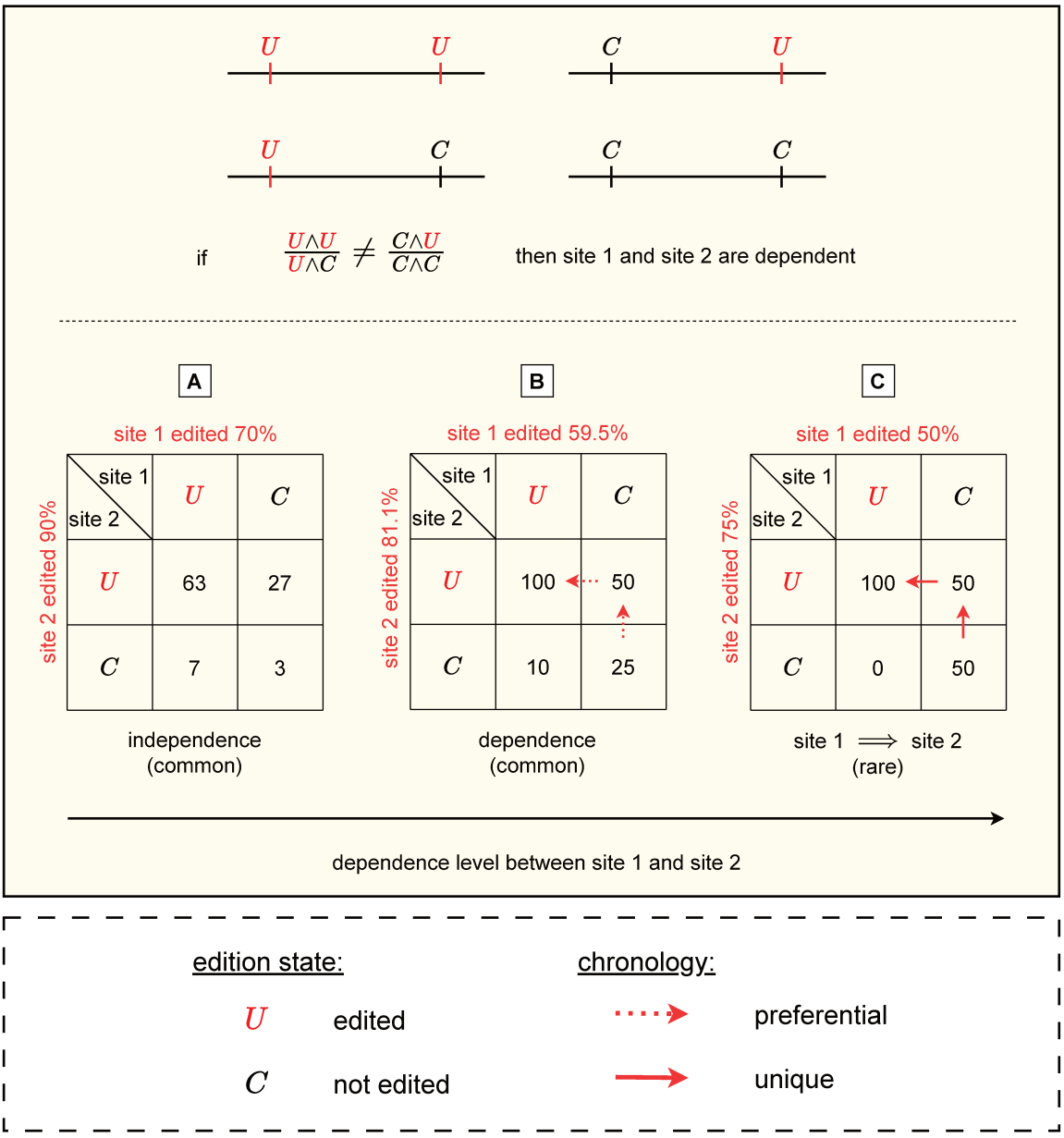


FIGURE 4.8 – Examining the statistical dependence between two editing sites. We can investigate the dependence between two editing sites by examining the difference in the fold change of one site's editing, conditional on the editing state of the other site. If we observe a significant difference, this allows us to reject the hypothesis of independence. In scenario **(A)**, we depict a hypothetical contingency table representing the number of transcripts for each combination of editing states for two sites. We notice that the fold change in editing of site 2 is 9 ($\frac{63}{7} = 9$ and $\frac{27}{3} = 9$), irrespective of the state of site 1. Under this condition, we cannot reject the hypothesis of independence. In scenarios **(B)** and **(C)**, we observe, following the same logic, that the pairs demonstrate dependency. In scenario **(B)**, site 2 seems to be preferentially edited before site 1 (indicating a preferred chronology), while in scenario **(C)**, site 2 is consistently edited prior to site 1 (indicating a unique chronology). Note that in all three scenarios, the marginal proportion of site 1 being edited is lower than that of site 2, which could potentially be explained by different reaction rates of editing on these two sites. Importantly, the observed dependency is independent of this difference.

In the same vein as the detection of epigenetic marks or RNA regulations, our most recent iteration of this project harnessed an established model—the DESeq2 model for RNA-Seq data—to assess the dependence between each annotated event along an RNA molecule. This testing is feasible provided that both events are covered by the same read.

Chapter 5

Multiple changepoint detection

Before starting this thesis, my succinct vision of a successful interdisciplinary project entailed the development of a novel statistical model or a new algorithm to handle each incoming biological project (the question and data). However, this vision swiftly evolved. As I have demonstrated in Chapter 4, it may be wise to economize on development by proposing or adapting an existing, proven model or algorithm. Yet, trusting in existing methodologies also entails continuing to develop interesting models and algorithms. Throughout my doctoral research, I have put this revised vision of interdisciplinary research into practice. In this chapter, I start by introducing a standard changepoints model that I employed in the detection of epigenetic marks [Liehrmann et al., 2021] and the detection of RNA regulations [Liehrmann et al., 2023], which yielded promising results. In the second part of this chapter, I introduce a new multiple changepoint detection algorithm, Ms.FPOP, that incorporates a multiscale penalty with better statistical properties than previously introduced penalties [Liehrmann and Rigaill, 2023].

5.1 Detecting changes in mean

Multiple changepoint detection, a regression problem, has been an area of active research since the 1950s [Page, 1954, 1957, Girshick and Rubin, 1952]. Initially sparked by a need for quality control within manufacturing operations, it has now risen to prominence as one of the "grand challenges of inference" in massive data analysis, as identified by the US National Research Council [Council et al., 2013]. Detecting changepoints is important in an extensive array of disciplines including genomics [Muggeo and Adelfio, 2010], neuroscience [Koepcke et al., 2016], econometrics [Bai, 1997], computer network security [Tartakovsky, 2014], and climate research [Reeves et al., 2007].

The prototypical and most prevalent changepoint detection problem is the identification of abrupt shifts in the mean of a univariate ordered signal, such as those manifested over time or along the genome. These sudden shifts, known as changepoints, delimit segments characterized by a homogeneous signal. In the context of my research, these changepoints might signify either the start/end of a peak in ChIP-Seq data, or the start/end of a DERs in RNA-Seq data drawn

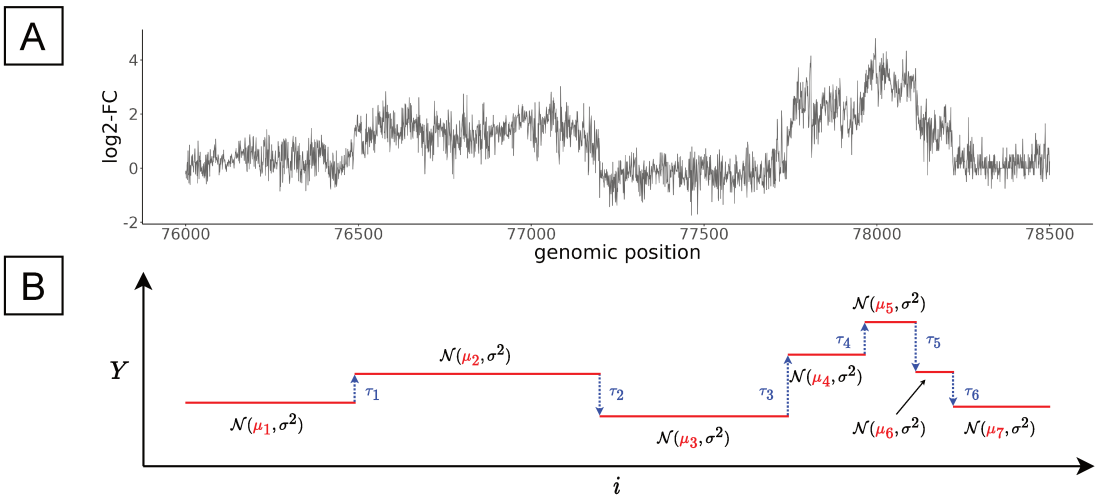


FIGURE 5.1 – The mean of the per-base $\log_2$-FC is affected by several noticeable changepoints. (A) The per-base $\log_2$-FC has been calculated from RNA-Seq data comparing two biological conditions, specifically for positions 76000 to 78500 of the chloroplast genome. Several changepoints can be visually identified, notably around positions 76500 and 77250. These changepoints mark a specific biological event : the accumulation of an intron in one of the two conditions, as detailed in Table 1 of Liehrmann et al. [2023]. (B) I present below a diagram of the standard Gaussian segmentation model applied to the same differential transcription profile. Each j^{th} segment is bound by two changepoints, τ_{j-1} and τ_j. Within this segment, datapoints Y_i are independent and follow a Gaussian distribution with mean μ_j and variance σ^2.

from two distinct biological conditions. In either scenario, these changepoints disclose biological events, such as genomic regions enriched with H3K4me3 epigenetic markers or disparities in RNA maturation processes.

The Figure 5.1.A illustrates an example of a differential transcription profile from an RNA seq experiment encompassing two distinct conditions. The signal displays significant variations in its mean. While detecting these changes may initially seems easy to spot with our eyes, it is actually a challenging problem. One way to grasp the challenge is by considering the number of potential segmentations of a profile with n datapoints. Each point, except the last one, can serve as a changepoint, resulting in a total of $n-1$ possible changepoints. Consequently, the number of segmentations reaches 2^{n-1}. For instance, for $n = 100$, the total number of segmentations exceeds 6×10^{29}. This raises numerous statistical and algorithmic problems.

In this context, during my thesis, I developed [Liehrmann and Rigaill, 2023] and applied to genomic data [Liehrmann et al., 2021, 2023] multiple changepoint detection algorithms that maximize a penalized likelihood. This approach, deeply anchored in traditional statistics, offers both asymptotic [Yao and Au, 1989, Boysen et al., 2009] and non-asymptotic [Lebarbier, 2005, Garreau and Arlot, 2018, Arlot et al., 2019] statistical guarantees for signal estimation and changepoint detection. Its computational efficiency is particularly suited for the intensive demands of genomic data analysis, where it is routine to handle profiles with millions of datapoints [Rigaill, 2015, Maidstone et al., 2016]. Finally, empirical evidence from both simulations [Fearnhead and Rigaill, 2020] and real-world applications frequently yields satisfactory outcomes, demonstrating

its effectiveness. Notably, it has already achieved state-of-the-art results in genomic applications [Lai et al., 2005, Hocking et al., 2013a, Cleynen et al., 2014b, Hocking et al., 2016].

5.2 Chapter summary at a glance

1. In Section 5.3, I introduce a standard model for multiple changepoint detection, along with the associated penalized likelihood problem. I applied this model on ChIP-Seq data in Liehrmann et al. [2021], and on RNA-Seq data in Liehrmann et al. [2023] as practical solution for the detection of peaks and candidate DERs, respectively. Various dynamic programming algorithms aimed at maximizing the penalized likelihood have been proposed over the years. I will introduce some of them in the second part of this first section.
2. In Section 5.5, I present a new multiscale penalty, introduced by Verzelen et al. [2020], that possesses superior statistical properties in terms of detection and localization compared to other penalties documented in the literature. Subsequently, I introduce a novel segmentation algorithm, Ms.FPOP, which leverages functional pruning techniques for efficient minimization of a least squares criterion with this multiscale penalty as elucitaded in Liehrmann and Rigaill [2023].

5.3 Model and penalized likelihood

5.3.1 The standard changepoints model

We consider the data $Y_1, Y_2, \cdots, Y_n$ and D changepoints $\tau_1 < \cdots < \tau_D$ within the range of 0 and n. We adopt the convention that $\tau_0 = 0$ and $\tau_{|\tau|} = n$. These changepoints define $|\tau| = D + 1$ distinct segments. The j^{th} segment includes the data $]\!]\tau_{j-1}, \tau_j]\!] = \{\tau_{j-1} + 1, \cdots \tau_j\}$.

Each segment is premised on the assumption that the Y_i therein are independent and follow the same Gaussian distribution, with a mean μ_j specific to that segment and a common variance σ^2. The model is illustrated in Figure 5.1.B. Expressed mathematically, we have :

$$\forall\ i\ \in]\!]\tau_{j-1}, \tau_j]\!] \quad Y_i \sim \mathcal{N}(\mu_j, \sigma^2) \quad iid. \tag{5.1}$$

5.3.2 Penalized likelihood

If the number of segments is known to be $|\tau|$, the model as described by Equation (5.1) is characterized by a parameter vector $\theta = (\mu_1, \ldots, \mu_{|\tau|}, \sigma^2, \tau_1, \ldots, \tau_{|\tau|})$. The log-likelihood function derived from this model, denoted as $\ell(y_1, \ldots, y_n; \theta)$, can be expressed as follows :

$$\ell(y_1, \ldots, y_n; \theta) = \sum_{j=1}^{|\tau|} f(y_{\tau_{j-1}+1}, \ldots, y_{\tau_j}; \mu_j, \sigma^2), \tag{5.2}$$

where $f(y_{\tau_{j-1}+1}, \cdots, y_{\tau_j}; \mu_j, \sigma^2)$ denotes the joint distribution of the data. Assuming Gaussian distribution and data independence, the log-likelihood can be expressed as follows :

$$-\frac{1}{2\sigma^2}\sum_{j=1}^{|\tau|}\sum_{i=\tau_{j-1}+1}^{\tau_j}(y_i-\mu_j)^2 \quad -\frac{n}{2}\log\left(2\pi\sigma^2\right). \tag{5.3}$$

By taking the derivative with respect to the parameter σ^2, which is assumed to be constant in (5.1), we find that to maximize the likelihood, we need to minimize the following quantity $C_{|\tau|,n}$, also known as least squares criterion (LSC) [Auger and Lawrence, 1989, Bellman and Kotkin, 1962, Fisher, 1958] :

$$\begin{aligned} C_{|\tau|,n} &= \min_{\substack{\tau_1,\dots,\tau_D \\ \mu_1,\dots,\mu_{|\tau|}}} \left\{ \sum_{j=1}^{|\tau|}\sum_{i=\tau_{j-1}+1}^{\tau_j}(y_i-\mu_j)^2 \right\} \\ &= \min_{\tau_1,\dots,\tau_D} \left\{ \sum_{j=1}^{|\tau|}\sum_{i=\tau_{j-1}+1}^{\tau_j}\left(y_i-\bar{y}_{\tau_{j-1}+1:\tau_j}\right)^2 \right\}, \end{aligned} \tag{5.4}$$

where $\bar{y}_{\tau_{j-1}+1:\tau_j}$ is the sample mean of the j^{th} segment :

$$\bar{y}_{\tau_{j-1}+1:\tau_j} = \sum_{i=\tau_{j-1}+1}^{\tau_j}\frac{y_i}{(\tau_j-\tau_{j-1})}.$$

In practice, the number of segments is usually unknown and needs to be determined from the data. In this context, without any form of penalization, the smallest value of $C_{|\tau|,n}$ will always be attained when $|\tau| = n$, leading to a segmentation cost of 0. As illustrated in the third panel of Figure 5.2, this result, while maximizing the likelihood, is clearly not meaningful from a practical standpoint, as it essentially overfits the data without revealing any underlying structure. In order to promote a more parsimonious, interpretable solution, as illustrated by the second panel of Figure 5.2, it is therefore conventional to introduce a penalty term in the likelihood, effectively discouraging models with excessive segmentation.

Numerous penalties have been proposed and examined in depth within the literature [Yao and Au, 1989, Birgé and Massart, 2001, Lebarbier, 2005, Zhang and Siegmund, 2006b, Davis et al., 2006, Baraud et al., 2009, Garreau and Arlot, 2018, Arlot et al., 2019, Verzelen et al., 2020]. The number of changepoints is typically a decreasing function of such penalties, which are commonly dependent on the parameters n and σ^2. For instance, one of the simplest and earliest penalties, proposed by Yao and Au [1989] and known as the Bayesian Information Criterion (BIC) or Schwarz Information Criterion (SIC) [Fryzlewicz, 2014], is linear in $|\tau|$ and can be expressed as $2\sigma^2\log(n)|\tau|$ (Figure 5.2).

The variance σ^2 is often required to be estimated empirically from the data. A commonly used method for this involves the unbiased estimator of the variance, denoted $\hat{\sigma}^2$, which is calculated as follows :

$$\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i-\bar{y}_{1:n})^2.$$

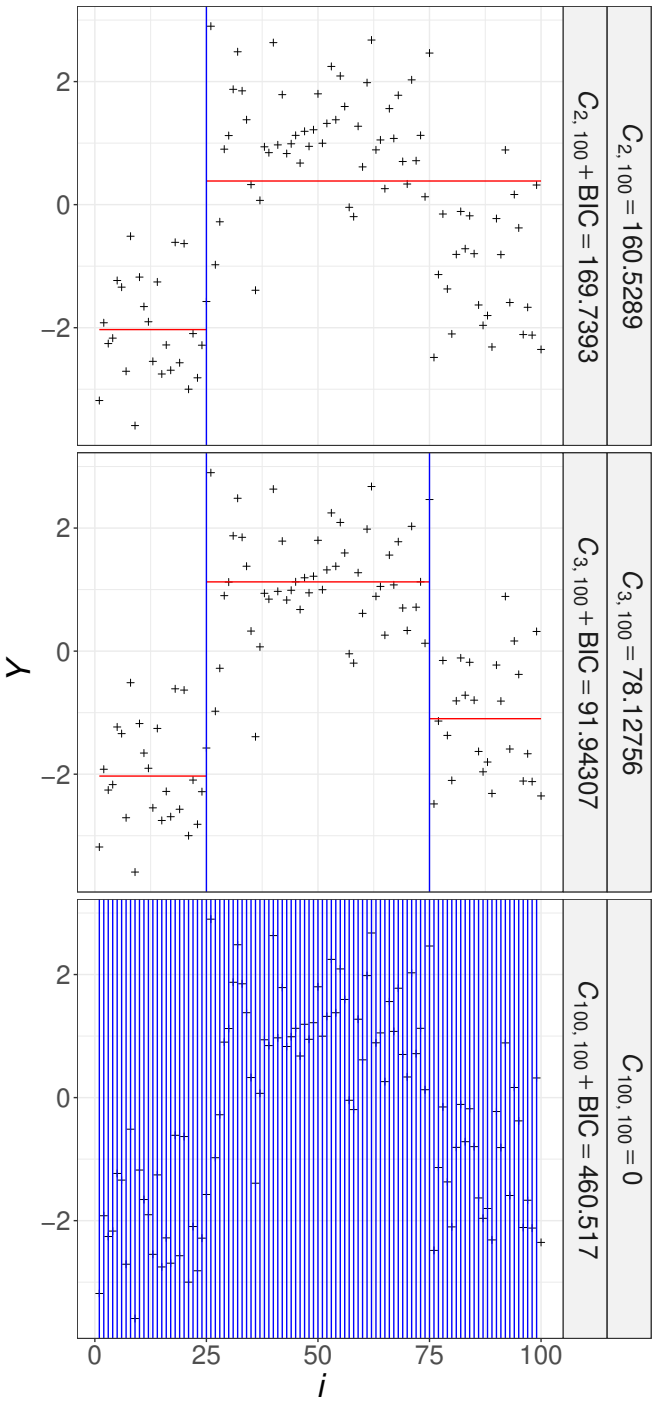


FIGURE 5.2 – Likelihood and penalized likelihood. An *iid* Gaussian signal of length $n = 100$ and variance $\sigma^2 = 1$ is affected by two changepoints at positions 25 and 75. The mean values corresponding to the initial, intermediate, and final segments are -2, 1, and -1, respectively. Different segmentation of the signal are depicted in a series of panels. From top to bottom, these segmentations represent the divisions of the signal into 2, 3, and 100 segments, respectively. With respect to the least squares criterion, the costs associated to these segmentations are $C_{2,100} = 160.530$, $C_{3,100} = 78.128$, and $C_{100,100} = 0$, in that order. In this scenario, an estimator aiming to minimize the cost would select the final segmentation into 100 segments. This estimator accurately detects the two true changepoints at positions 25 and 75, but unfortunately also falsely identifies 97 other positions as changepoints (the last datapoint can never be a changepoint). Incorporating a penalty term into the cost, like in the form of $2|\tau| \log(100)$, also known as BIC, changes the optimal segmentation. With this penalty, the (penalized) cost-minimizing segmentation would be a division into three segments, which accurately identifies the two real changepoints without any false positives.

I used this strategy for estimating σ^2 and calibrating the BIC penalty in [Liehrmann et al., 2023]. Nonetheless, it is important to note that the literature also presents more robust estimatiors for σ^2 [Hall et al., 1990].

In [Liehrmann et al., 2021], I employed an alternative strategy that aims to calibrate the penalty based on ChIP-Seq profiles annotated by biologists and bioinformaticians. The underlying concept is straightforward : identify the penalty values that minimize the annotation error [Hocking et al., 2013b].

Box 2: Section switch

☞ At this stage, non-specialist readers should possess a sufficient technical foundation on the standard changepoints model to engage with Liehrmann et al. [2021] (Appendix A). In this paper, I compare the accuracy of several models of multiple changepoint detection, as well as peak calling heuristics from the bioinformatics literature, in the context of the detection of epigenetic marks and the supervised learning of these methods' parameters. Notably, I demonstrate that the standard changepoints model has an accuracy at least as good as its competitors.

☞ For readers less interested in the algorithmic dimension of this thesis, and in particular the development of the Ms.FPOP algorithm, I would recommend proceeding directly to Chapter 6.

5.3.3 Definition of the penalized optimization problem

Considering a linear penalty expressed as $\alpha|\tau|$, where α denotes a constant, the algorithmic objective is to optimize the ensuing penalized optimization problem :

$$F_n = \min_{\substack{|\tau| \\ \tau_1,\ldots,\tau_D}} \left\{ \sum_{j=1}^{|\tau|} \left[\sum_{i=\tau_{j-1}+1}^{\tau_j} \left(y_i - \bar{y}_{\tau_{j-1}+1:\tau_j}\right)^2 \right] + \alpha|\tau| \right\}. \tag{5.5}$$

The number of segmentations that could be solution of the problem (5.5) is 2^{n-1}. As stated earlier, the search for the optimal solution by investigating all possible solutions independently becomes fastly computationally unfeasible.

5.4 Minimizing F_n through dynamic programming

Through the utilization of dynamic programming, F_n can be minimized efficiently. This computation hinges on a specific recurrence relation, and two primary forms of this relation are discussed in the literature :

"Recurrence on the last changepoint" the first approach considers all the possible positions of the last changepoint ;

"Recurrence on the last segment mean" the second approach considers all the possible means of the last segment.

5.4.1 Recurrence on the last changepoint

5.4.1.1 Optimal partioning

Because the cost of a segmentation in (5.5) is the sum of the cost of its segments, meaning that the cost $\sum_{i=\tau_{j-1}+1}^{\tau_j} \left(y_i - \bar{y}_{\tau_{j-1}+1:\tau_j}\right)^2$ only depends of data within the j^{th} segment, one can apply Bellman's dynamic programming principle to minimize (5.5) [Auger and Lawrence, 1989, Bellman and Kotkin, 1962].

The recurrence. The Optimal Partitioning (OP) methodology, depicted in Jackson et al. [2003], is specifically tailored to address the linear penalty present in (5.5). The optimal cost at iteration t, F_t, is derived by considering the costs of the best segmentation up to last changepoint candidate s, F_s, such that $0 \leq s < t$. To this, we add the cost of the last segment and the penalty. Mathematically, we obtain the following recurrence :

$$F_t = \min_{0 \leq s < t} \left\{ F_s + \sum_{i=s+1}^{t} \left(y_i - \bar{y}_{s+1:t}\right)^2 \right\} + \alpha, \quad \text{initialized at } F_0 = -\alpha. \tag{5.6}$$

Complexity. It can be proven that the time complexity of OP is $O(n^2)$, and the memory complexity is $O(n)$ [Jackson et al., 2003].

5.4.1.2 Inequalities based pruning

The recurrence (5.6) suggests that we have to go through all changepoint candidates s before t. Naively, reducing the numbers of comparisons to be performed at each iteration reduces the overall complexity. Killick et al. [2012] show that we can indeed, without resorting to an approximation, definitively eliminate all s such that :

$$F_s + \sum_{i=s+1}^{t} \left(y_i - \bar{y}_{s+1:t}\right)^2 > F_t. \tag{5.7}$$

The recurrence. This pruning idea is implemented in PELT [Killick et al., 2012]. PELT operates using two recurrences, one on F_t, another on the set of changepoints to consider at each iteration R_t :

$$\begin{aligned} F_t &= \min_{s \in R_{t-1}} \left\{ F_s + \sum_{i=s+1}^{t} \left(y_i - \bar{y}_{s+1:t}\right)^2 \right\} + \alpha \\ R_t &= \left\{ s \in R_{t-1} | F_s + \sum_{i=s+1}^{t} \left(y_i - \bar{y}_{s+1:t}\right)^2 \leq F_t \right\} \cup \{t\}. \end{aligned} \tag{5.8}$$

Complexity. If the number of changepoints increases linearly with n, the time complexity of PELT is $O(n)$. If there are few or no changepoints, it is still $O(n^2)$. The memory complexity is $O(n)$ [Killick et al., 2012].

5.4.2 Recurrence on the last segment mean

5.4.2.1 Functional based pruning

Building upon the prior works of Rigaill [2015] and Johnson [2013], Maidstone et al. [2016] proposed a recurrence on the last segment mean μ rather than the last changepoint candidate position s. In this context the pruning is said "functional". This idea is implemented within the FPOP algorithm [Maidstone et al., 2016].

Functionalization. FPOP introduces for every changepoint candidate s its best cost as function of the last segment mean μ at iteration t, $\widetilde{f}_{t,s}(\mu)$. Formally,

$$\widetilde{f}_{t,s}(\mu) = F_s + \sum_{i=s+1}^{t} (y_i - \mu)^2 + \alpha, \quad \text{with} \quad \widetilde{f}_{t,t}(\mu) = F_t + \alpha \quad \text{and} \quad F_0 = -\alpha. \tag{5.9}$$

Throughout the procedure, $\widetilde{f}_{t,s}(\mu)$ is maintained and updated with new datapoint y_t using the following formula :

$$\widetilde{f}_{t,s}(\mu) = \widetilde{f}_{t-1,s}(\mu) + (y_t - \mu)^2. \tag{5.10}$$

At each iteration t, the FPOP algorithm considers the minimum of the $\widetilde{f}_{t,s}(\mu)$, denoted as $\widetilde{F}_t(\mu)$, a piecewise quadratic function :

$$\widetilde{F}_t(\mu) = \min_{s \leq t} \left\{ \widetilde{f}_{t,s}(\mu) \right\}. \tag{5.11}$$

By definition, each interval of μ is associated with one last changepoint candidate s that achieves this optimal cost. Note that F_t, is obtained by minimizing (5.11) over μ. Formally, $F_t = \min_{\mu} \left[\widetilde{F}_t(\mu) \right]$.

The recurrence. Maidstone et al. [2016] have demonstrated that $\widetilde{F}_t(\mu)$ can be updated iteratively,

$$\widetilde{F}_t(\mu) = \min \left\{ \underbrace{\widetilde{F}_{t-1}(\mu)}_{\substack{\textit{best past} \\ \textit{changepoint candidates}}} , \underbrace{F_{t-1} + \alpha}_{\substack{\textit{last introduced} \\ \textit{changepoint candidate}}} \right\} + (y_t - \mu)^2 \tag{5.12}$$

The recursion (5.12) suggests that to update $\widetilde{F}_t(\mu)$ we need to compare the cost functions of changepoint candidates that achieve optimal cost (*best past changepoint candidates*) with the cost function of the most recently introduced changepoint candidate, i.e. $F_{t-1} + \alpha$. The other changepoint candidates can be pruned. More formally, for each changepoint candidate s, we define its "living set", $Z^*_{t,s}$, as the set of μ for which $\widetilde{f}_{t,s}(\mu)$ equals $\widetilde{F}_t(\mu)$,

$$Z^*_{t,s} = \left\{ \mu \mid \widetilde{f}_{t,s} = \widetilde{F}_t(\mu) \right\}. \tag{5.13}$$

Given (5.12), s is pruned as soon as its living set is empty, which is justified because

$$Z^*_{t,s} \supset Z^*_{t+1,s} \quad \text{and} \quad Z^*_{t,s} = \emptyset \implies Z^*_{t+1,s} = \emptyset \ . \tag{5.14}$$

Application. Below I propose a detailed example of the calculation of the recurrence (5.12) at the third iteration on the signal $y_1 = 1,\ y_2 = 0.5,\ y_3 = 0.5$ (Figure 5.3).

Initialization of the new changepoint candidate $s = 2$: In Figure 5.3.A, on the left panel, I have depicted, in bold, the piecewise quadratic function $\widetilde{F}_2(\mu)$, composed of cost functions of the best past changepoint candidates s=0 (no changepoints) and s=1 (divides the signal into two, with the first segment composed of point $\{y_1\}$ and the second segment composed of datapoints $\{y_2, y_3\}$). The changepoint candidate $s = 0$ has an optimal cost over $Z^*_{2,0} = [0.5, 1.7]$, while s=1 has an optimal cost over $Z^*_{2,1} =]1.7, 2]$ (the intervals are shown on the same figure below the curves). The minimum of $\widetilde{F}_2(\mu)$, which equals 0.125, is obtained through polynomial calculus. It is represented by the origin of the arrow on the right panel. After calculating this minimum, we initialize a new changepoint candidate whose cost function $\widetilde{f}_{2,2}(\mu)$ is equal to $F_2 + \alpha = 0.125 + 0.5 = 0.625$. By default, its living set is equal to the range of the data : $Z^*_{2,2} = [0.5, 2]$.

Recurrence interval-by-interval : The function $\min\{\widetilde{F}_2(\mu), F_2 + \alpha\}$ is then calculated interval by interval, once again utilizing polynomial calculus. In our example, over the first interval $\mu \in [0.5, 1.7]$ (left panel of Figure 5.3.B), we seek the roots of the polynomial

$$\widetilde{f}_{2,0}(\mu) - (F_2 + \alpha) = 0.625 - 3\mu + 2\mu^2.$$

The two roots equate to 0.25 and 1.25. Consequently, the changepoint candidate $s = 0$ is optimal over $Z^*_{2,0} = [0.5, 1.25]$, and the changepoint candidate $s = 2$ is optimal over $Z^*_{2,2} =]1.25, 1.7]$. Similarly, over the second interval $\mu \in [1.7, 2]$ (right panel of Figure 5.3.B), we seek the roots of the polynomial

$$\widetilde{f}_{2,1}(\mu) - (F_2 + \alpha) = 0.125 - \mu + \mu^2.$$

The two roots equate to 0.15 and 0.86. As a result, $s = 2$ is optimal over $Z^*_{2,2} = [1.25, 2]$, and the living set of $s = 1$ is empty. We can, therefore, safely prune $s = 1$.

Adding new datapoint y_3 : The third iteration ends by updating the cost functions of the remaining changepoint candidates with the last datapoint y_3 cost :

$$\widetilde{f}_{3,0}(\mu) = \widetilde{f}_{2,0}(\mu) + (y_3 - \mu)^2 = 2.25 - 5\mu + 3\mu^2,$$
$$\widetilde{f}_{3,2}(\mu) = \widetilde{f}_{2,2}(\mu) + (y_3 - \mu)^2 = 1.625 - 2\mu + \mu^2.$$

Complexity. Rigaill [2015] has shown that the number of intervals at iteration n is less than $2n - 1$. From this upper bound, we infer a worst-case complexity of $O(n^2)$, and a memory

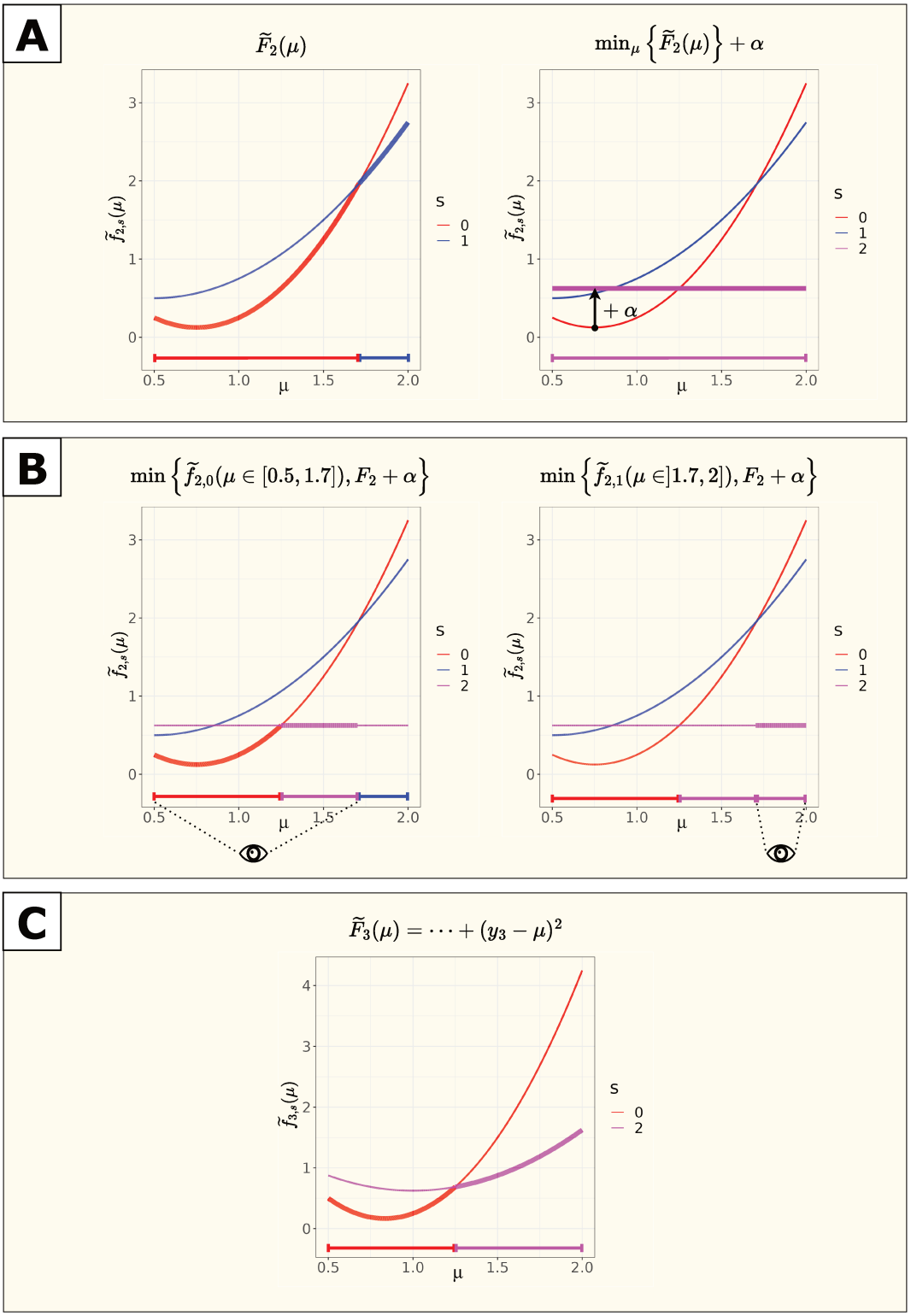


FIGURE 5.3 – Illustration of the recurrence on the last segment mean. Detailed example of the calculation of the recurrence (5.12) at the third iteration on the signal y_1, y_2, y_3. Step (A) illustrates the initialization of the new changepoint candidate $s = 2$. Step (B) shows the update interval-by-interval of the piecewise quadratic function $\widetilde{F}_2(\mu)$ by comparing best past changepoint candidates ($s = 0$ and $s = 1$) with the newly introduced one ($s = 2$). The recurrence ends at step (C), with the addition of the new datapoint y_3 cost. Each cost function (described above each panel) is prominently displayed as a bold line in the figure.

complexity of $O(n)$. However, for many signals, computation times are log-linear or linear in n when there are few changepoints or when the number of changepoints increases linearly with n, respectively. A theoretical proof supporting a log-linear complexity can be found in Romano et al. [2023].

5.4.2.2 FPOP vs PELT

Maidstone et al. [2016] has demonstrated that the complexity of FPOP is always less than that of PELT, meaning that FPOP prunes at least as well as PELT, regardless of the signal. The computation time of FPOP is also better than PELT (Figure 7 of Maidstone et al. [2016] and Figure 3 of Liehrmann and Rigaill [2023]). Both methods are implemented in C++/C.

5.5 Ms.FPOP : An exact and fast segmentation algorithm with a multiscale penalty

5.5.1 Key criteria for effective changepoint detection and localization

As detailed in Verzelen et al. [2020], an efficient changepoint detection and localization estimator should fulfill certain properties. These characteristics are formally articulated in Section 3.3 of the same study. My goal is to deliver below a succinct and understandable synopsis of these principles.

(NoSp) Spurious changepoints are avoided. This first principle states that the procedure should avoid estimating more than one changepoint in the proximity of a real one.

(Detec) Evident changepoints are detected. The second principle emphasizes that the procedure should identify an 'evident' changepoint—one whose height and spacing from neighboring changepoints are large enough. A formal lower bound can be found in the Proposition 5 of Verzelen et al. [2020].

(Loc) Localization hinges on height and spacing. The last principle states that the distance error between an evident changepoint and its estimation should only depend on the height of this changepoint and spacing from neighboring ones. Specifically, the procedure should localized this changepoint at optimal rate as define in Equation (30) of Verzelen et al. [2020].

Limitations of current estimators. The changepoint detection estimator builds on a LSC with a BIC penalty does not simultaneously satisfy the **(NoSp)**, **(Detec)**, and **(Loc)** properties. Likewise, the penalty proposed in Lebarbier [2005], despite possessing superior statistical properties compared to BIC, does not met concurrently all three properties either (Section 4.2.3 of Verzelen et al. [2020]). In particular both penalties fail to retrieve with high probability evident

changepoints with small jump heights and large adjacent segments. This typically occurs in scenarios where there are few well-spread changepoints. This is clearly illustrated in our simulations on hat-like and step-like profiles for the BIC penalty (see Section 4.3 of Liehrmann and Rigaill [2023]).

5.5.2 Optimization with a multiscale penalty

Definition. [Verzelen et al., 2020] put forth a LSC with a multiscale penalty adhering to **(NoSp)**, **(Detec)**, and **(Loc)** properties. This multiscale penalty is defined by the negative logarithm of segment lengths, which promotes the detection of well-spread changepoints. The penalty can be expressed mathematically as :

$$\sum_{j=1}^{|\tau|} \gamma + \beta \log(n) - \beta \log(\tau_j - \tau_{j-1}). \tag{5.15}$$

Here, $\gamma = qL$ and $\beta = 2L$ where q is a positive value and $L > 1$.

The penalized optimization problem given by (5.5) can be reformulated with the multiscale penalty as follows :

$$F_n = \min_{\substack{|\tau| \\ \tau_1,\ldots,\tau_D}} \left\{ \sum_{j=1}^{|\tau|} \left[\sum_{i=\tau_{j-1}+1}^{\tau_j} \left(y_i - \bar{y}_{\tau_{j-1}+1:\tau_j}\right)^2 - \beta \log(\tau_j - \tau_{j-1}) \right] + \alpha|\tau| \right\}. \tag{5.16}$$

Applying $\alpha = \gamma + \beta \log(n)$, we retrieve the multiscale penalty of (5.15), with γ and β as the constants requiring calibration.

Complexity. As mentioned in Verzelen et al. [2020] (5.16) demonstrates segment additivity. This attribute allows the application of dynamic programming algorithms that utilize a recurrence based on the last changepoint position or the last segment mean to optimize it. As detailed above, the latter recurrence exhibits better pruning capacity and computational efficiency, making it a more compelling choice for optimizing (5.15).

Ms.FPOP algorithm. In Liehrmann and Rigaill [2023], in collaboration with Guillem Rigaill I introduce Ms.FPOP, a dynamic programming algorithm designed to optimize a more general penalty, inclusive of (5.15). This algorithm extends the functional pruning techniques used by FPOP. This expansion is notably challenging due to the fact that (5.16) do not meet the point additive cost function criteria, as detailed in Section 2.2 of the same article. While FPOP maintains an optimal parameter set $(Z^*_{t,s})$ that progressively reduces with the addition of new datapoints, (5.16) does not guarantee such reduction, thereby complicating the update process. Ms.FPOP circumvents this complexity by managing a marginally larger set $(Z_{t,s})$ easier to update. As a reminder, $Z_{t,s}$ represents a set of intervals on μ, as depicted in Figure 5.4.

Sketch of the update rule. In this paragraph, I aim to elucidate the process of updating the best changepoint candidates in the Ms.FPOP algorithm (Equation (12) of Liehrmann and Rigaill [2023]). Importantly, the purpose here is not to justify why this update ensures the optimality of Ms.FPOP, but rather to present the essential operations comprising it. This explanation should serve as a valuable introductory step prior to engaging with Liehrmann and Rigaill [2023] in detail.

Figure 5.4.A illustrates the update of a recently initialized changepoint candidate t with the past changepoint candidates s. The cost function of t is compared to each s cost function. For each of these comparisons, the μ interval on which t does not have the lower cost is found using polynomial calculus and saved. The living set of t is then determined by taking the union followed by the complement in μ of these intervals.

Figure 5.4.B illustrates the update of a changepoint candidate s with another candidate s', which is initialized after s. The cost function of s is compared to that of s', and the μ interval where s exhibits a lower cost is determined using polynomial calculus and saved. The living set of s is then intersected with this interval, with the resulting intersection serving as the new living set of s. The interval on which s has a lower cost relative to s' diminishes at each iteration. This property suggests comparing s and s' at multiple iterations. In practice, at each iteration and for each s, we randomly draw one s' for comparison.

At the end of iteration t, the living set of s, $Z_{t,s}$—which, as a reminder, includes the true living set of s, $Z^*_{t,s}$—is empty. Consequently, s is pruned.

Box 3: Section switch

☞ At this stage, readers should have acquired sufficient knowledge of dynamic programming and its acceleration via functional pruning, or pruning based on inequality techniques, to engage with Liehrmann and Rigaill [2023] (Appendix B.1). In this paper, I present the Ms.FPOP algorithm in detail, and demonstrate that for large signals (with $n \geq 10^5$) containing relatively few real changepoints, Ms.FPOP is typically quasi-linear and an order of magnitude faster than PELT. Lastly, I illustrate through simple simulations that for sufficiently large profiles (n$\geq 10^4$), Ms.FPOP using the multiscale penalty is typically more powerful than FPOP using the BIC penalty.

5.5.3 Implementation of Ms.FPOP

5.5.3.1 Foreword

In the remainder of this section, I will shed light on the implementation of Ms.FPOP, which has not been elaborated upon extensively in the original paper.

The multifaceted concepts intrinsic to the Ms.FPOP algorithm, such as quadratic functions, last changepoint candidates, and intervals, suggested that an object-oriented programming approach would be suitable for its implementation. This approach provides a structured platform

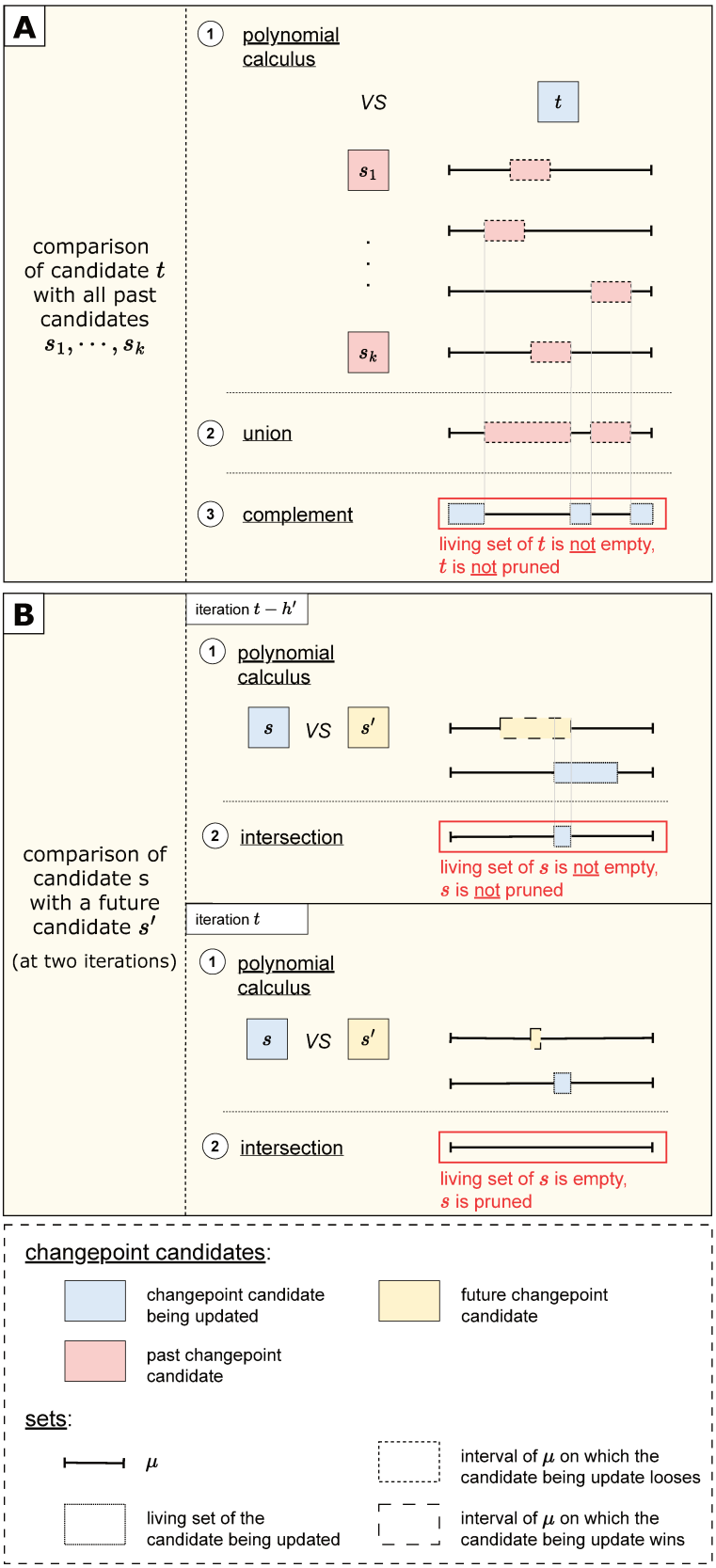


FIGURE 5.4 – Ms.FPOP : sketch of the update rule. Panel (A) shows the update of a new changepoint candidate t with past changepoint candidates s. Panel (B) depicts the update of a candidate s with another candidate (s'), initialized after s. Both updates involves comparing cost function of respective changepoint candidates by using polynomial calculus. This is followed by a series of set operations (union, complement, intersection).

where all information and related methods for manipulating an object are grouped, enhancing code readability and maintainability.

Ms.FPOP has been implemented in C++, a compiled, highly efficient language whose object-oriented nature aligns perfectly with our requirements. The extensive utilization of the C++ standard library in this implementation harnesses powerful features like containers, iterators, and a variety of functions for sorting, searching, counting, and object manipulation, thereby substantiating the decision to use the C++.

5.5.3.2 Overview of the classes.

Six classes were identified during the design of Ms.FPOP :

Candidate. The first class, Candidate, defines the concept of last changepoint candidate. Each changepoint candidate is characterized by its position. From a functional perspective, it is associated with a cost function and a living set. The cost function can be broken down into three parts : the cost of the best segmentation up to the changepoint candidate, the quadratic form, and the penalty which depends on the last segment length.

Interval. The second class, Interval, defines the concept of an interval. An interval is bounded by two real numbers. An empty interval is represented by an upper bound smaller than the lower bound. As justified in Section B.2, I have chosen to treat the specific case of singletons [1] as empty intervals.

Ordered_list_of_intervals. The third class, Ordered_list_of_intervals, defines a list of non-empty intervals ordered by their lower bound. The ordered property of this list is used to enhance the performance of updating the living set of changepoint candidates. In particular, this structure speeds up set operations such as union, complement and intersection (figure 5.4).

MsFPOP. The fourth class, MsFPOP, is the main class of this project. It facilitates instantiating a segmentation problem based on the data to be segmented and the penalty. Leveraging the other classes, it implements the procedure for estimation changepoints.

Quadratic. The fifth class, Quadratic, defines the quadratic form $a_0 + a_1\mu + a_2\mu^2$. This quadratic form is one of the components of the cost function associated with each changepoint candidate.

Sampling. The sixth class, Sampling, implements various strategies for sampling changepoint candidates, specifically those introduced after a defined point in time (future changepoint candidates).

In Section B.2 I elaborate on the six classes mentioned above, as well as the relationships between the objects they define. Notably, I explain in details a few implementation choices that enhance the overall execution time of Ms.FPOP.

1. A singleton is an interval of the form $[a, a]$.

5.5.3.3 Ms.FPOP R package

An R package, named after the method, was implemented using the Rcpp R package [Eddelbuettel and François, 2011], which allows calling C++ code within the R environment via a wrapper function. The Ms.FPOP package is available on GitHub : `https://aliehrmann.github.io/MsFPOP/index.html`. Importantly, this package includes a Vignette which shows on a minimal example how to use the main function. This Vignette should be considered as an extension of this manuscript.

Chapter 6

Applications for the multiscale analysis of the transcriptome

This chapter highlights the engineering facet of my thesis. Here below, I articulate my strategy for precise and rigorous analysis of expression differences and co-maturations. This strategy leverages the DESeq2 model and includes the control of evaluated differences, for instance, by employing a post-hoc procedure. Subsequently, I detail how I have incorporated this strategy into two R packages—DiffSegR and comaturationTrackeR. These tools exemplify the successful integration of complex analytical methodologies into practical, user-friendly software solutions.

6.1 Differential analysis

An important aspect of the transcriptome-wide detection of expression differences and co-maturations is the quantification of systematic changes between two groups, also known as differential analysis. In the first instance, the change pertains to the expression level of a site depending on the biological condition ; in the second instance, it relates to the maturation level of a site, contingent upon the maturation state of a second site. Quantifying these changes is challenging because the expression and maturation levels of a site can vary between samples. To account for this variability, both technical and biological, it is crucial to model the counts per event or per pair of events effectively. The GLM with a negative binomial distribution for RNA-Seq data, as implemented in the DESeq2 R package Love et al. [2014], performs this task reasonably well.

6.2 Chapter summary at a glance

1. Section 6.3 introduces key elements of the statistical model of gene counts implemented in DESeq2.
2. In Section 6.4, I unveil how to use the statistical model of DESeq2 to evaluate candidate DERs identified using FPOP. This is followed by a short presentation of DiffSegR, an R package that integrates the Baseline 2 and DESeq2 as shown in Liehrmann et al. [2023].

3. In Section 6.5, I provide a brief introduction of comaturationtrckeR, a method that exists in two forms : a published R pipeline [Guilcher et al., 2021] and a subsequent R package (still in development). The second version also leverages the statistical model of DESeq2 to assess co-maturations.

6.3 Generalized linear model for RNA-Seq data

6.3.1 Gene counts model

Let's revisit the DESeq2 model for gene counts. Here, K_{jd} is defined as the number of sequencing reads that align onto gene j in sample d, a concept diagrammatically depicted for one sample in Figure 3.7.A. In addition, we designate $q_{j,d}$ to be a quantity proportional to the expected concentration of cDNA fragments (Figure 3.5) for gene j in sample d.

To simplify matters, technical artifacts can be reduced to a multiplicative factor for each sample, denoted as s_d, or the "size factor". This factor adjusts variations in read counts across samples to account for differences in the total number of sequenced reads per sample. For instance, if sample A contains twice the total sequenced reads as sample B, it is reasonable to anticipate twice the reads mapping to each gene, suggesting that $s_A = 2s_B$.

Through systematic empirical analysis, it has been observed that the variance in gene counts obtained from multiple biological replicates tends to exceed their mean. In statistical terms, these counts display an "overdispersion" in comparison with a Poisson distribution. To account for overdispersion effectively, DESeq2 employs the gamma-Poisson distribution, also referred to as the negative binomial distribution. This approach introduces an extra gene-specific parameter, symbolized as φ_j, that establishes a relationship between the mean and variance. Mathematically, the DESeq2 model for gene counts is expressed as :

$$\begin{aligned} K_{j,d} &\sim \mathrm{NB}\left(\mu_{j,d}, \varphi_j\right), \\ \mathrm{Var}\left(K_{j,d}\right) &= \underbrace{\mu_{j,d}}_{\text{technical noise}} + \underbrace{\varphi_j \mu_{j,d}^2}_{\text{biological noise}}, \end{aligned} \tag{6.1}$$

where $\mu_{j,d} = s_j q_{j,d}$ designates the un-normalized mean expression of gene j in sample d. The variance can be decomposed in two components : (technical noise) the variability in the measurements, and (biological noise) the variability in the biology of the samples. Additionally, one can observe a characteristic pattern in the relationship between gene dispersion and mean values in RNA-Seq data (Figure 6.1). The trend in dispersion smoothly decreases as gene expression increases, and eventually reaches an asymptote.

6.3.2 Generalized linear model

The underlying proportion $q_{j,d}$ can be effectively represented using the notation of a GLM. This involves the use of a design matrix, X, and gene-specific regression parameters, symbolized

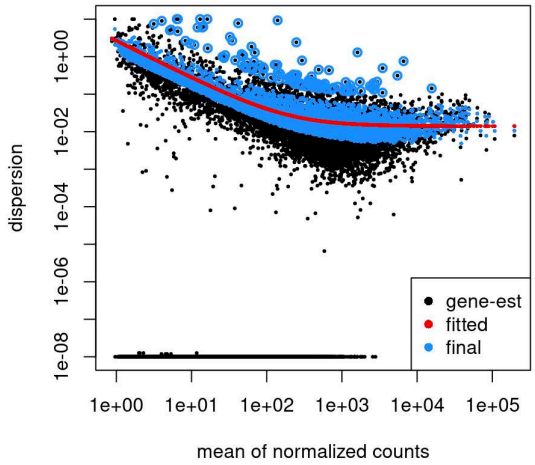


FIGURE 6.1 – Visualization of the gene dispersion trend in RNA-Seq data. The figure is derived from the DESeq2 R package Vignette.

as β_j. To make this model intuitive, I describe X and β_j below in the simple scenario where we are comparing two biological conditions (♣).

The GLM is characterized as "linear" since we apply the regression parameters, β_j, to form a linear combination of the columns in the design matrix, X. This is expressed as a matrix multiplication, $X\beta_j$, that aims to minimize the error (or log-likelihood) when approximating normalized gene counts—gene counts divided by multiplicative size factors— denoted as $\tilde{K} = s_d^{-1} K_{j,d}$.

The term "generalized" in GLM refers to the utilization of a link function, which establishes the relationship between the linear predictor, $X\beta_j$, and the underlying proportions, $q_{j,d}$. In the case of DESeq2, a $\log_2$ link function is used. Mathematically, this relationship is expressed as :

$$\log_2(q_{j,d}) = x_d \beta_j \tag{6.2}$$

where x_d denotes the d^{th} row of X.

After carrying out the non-trivial task of estimating s_d and φ_j parameters, as detailed in Love et al. [2014], we can proceed to estimate the coefficients β_j. This task can be accomplished utilizing standard GLM algorithms, as thoroughly elaborated in the works of Park and Hastie [2007] and Friedman et al. [2010].

♣ **Difference between two biological conditions.** Let's consider a simple scenario[1] that includes two distinct biological conditions, each with two samples. In this case, the design matrix

1. Freely inspired by the differential expression analysis courses taught by Christophe Ambroise, Professor of Statistics at the University of Évry Val d'Essonne.

X can be formulated as :

$$X = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

In the context of this two-condition design matrix, the gene-specific regression parameters β_j can be expressed as :

$$\beta_j = \begin{pmatrix} \beta_{j,0} \\ \beta_{j,1} \end{pmatrix}.$$

Here, $\beta_{j,0}$ symbolizes the $\log_2$ of the mean of normalized counts for the j^{th} gene in the first two samples, which belong to the first biological condition. In addition, $\beta_{j,0} + \beta_{j,1}$ represents the $\log_2$ of the mean of normalized counts for the j^{th} gene in the last two samples, associated with the second biological condition. To illustrate, let's assume that $\beta_{j,0} = 3$ and $\beta_{j,1} = 1$. In this case, the underlying proportions $q_{j,d}$ for samples 1, 2, 3, and 4 are :

$$\begin{aligned} q_{j,1} = q_{j,2} &= 2^{\beta_{j,0}} = 8, \\ q_{j,3} = q_{j,4} &= 2^{\beta_{j,0}+\beta_{j,1}} = 16. \end{aligned}$$

The $\log_2$ fold-change ($\log_2$-FC) between the mean of normalized counts of the two conditions under comparison is then $\beta_{j,1}$:

$$\log_2\left(\frac{q_{j,3}}{q_{j,1}}\right) = \log_2(q_{j,3}) - \log_2(q_{j,1}) = (\beta_{j,0} + \beta_{j,1}) - \beta_{j,0} = \beta_{j,1}.$$

6.3.3 Contrast

Following the estimation of the GLM parameters to individual genes, the subsequent statistical inference typically involves scrutiny of either a singular estimated regression parameter's nullity or that of a linear combination of such parameters, often referred to as a "contrast". Mathematically, the null hypothesis, denoted as H_0, is characterized as follows :

$$H_0 : \langle c, \beta_j \rangle = 0, \tag{6.3}$$

where $c \in \mathbb{R}^p$ symbolizes the contrast vector, with p the number of parameters. In the earlier described scenario (♣), the process of assessing differences in gene expression across the two compared conditions aligns with testing the nullity of the coefficient $\beta_{j,1}$, a.k.a the estimated $\log_2$-FC between the mean of normalized counts :

$$H_0 : \beta_{j,1} = 0. \tag{6.4}$$

To do so, both the Wald test and likelihood ratio test are available for GLMs with known (asymptotic) distribution under the null hypothesis (6.3). In theory, the distribution of associated

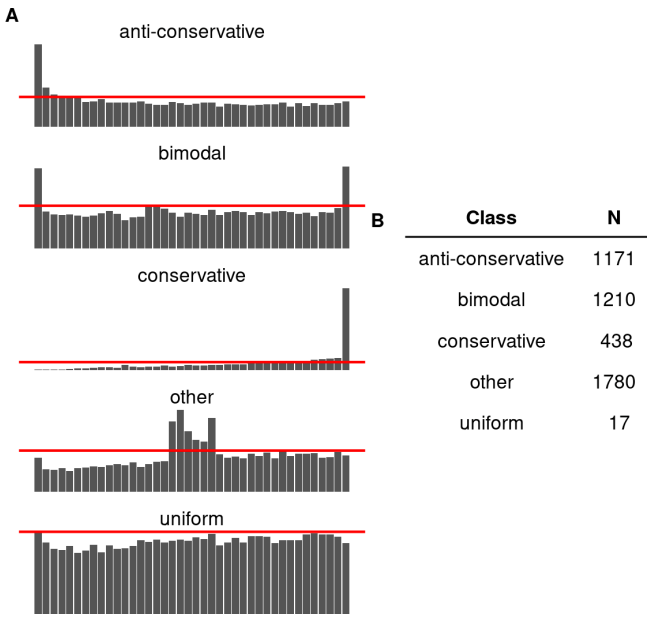


FIGURE 6.2 – Different p-value histogram classes. (A) This figure depicts various classes of p-value histograms. The algorithmic thresholds demarcating distinct classes of p-value histograms are depicted by red lines. Unique histogram types such as bimodal, conservative, and others stand out as anomalies. The anti-conservative histogram is the expected in an experiment with high statistical power and differentially expressed genes, denoted by a peak at the lower p-values (uniform otherwise). In a low statistical power experiment or one lacking differentially expressed genes, a uniform histogram is anticipated. (B) Summary of p-value histograms identified from 4,616 Gene Expression Omnibus datasets. The figure is taken from Päll et al. [2023].

raw p-values is dominated by a uniform distribution. Therefore, any deviations from this pattern in the raw p-value histogram, as illustrated by the *bimodal*, *conservative* and *other* histograms in Figure 6.2, can reveal inadequacies in the statistical model's fit to the data [Rigaill et al., 2016].

6.3.4 Multiple testing

In the process of evaluating a large number of genes (or regions as discussed in the subsequent section) for expression differences, it is often deemed acceptable to allow for a certain fraction of false positives (genes incorrectly identified as differentially expressed) in order to yield a higher count of true ones. The prevalent approach in dealing with large-scale multiple testing is through controlling the False Discovery Rate (FDR) [Benjamini and Hochberg, 1995], which refers to expected proportion of false positives amongst all selected genes, known as the False Discovery Proportion (FDP). The Benjamini-Hochberg procedure, typically adopted to control the FDR, is effective when the null hypotheses are independent or show a specific kind of positive dependence called Positive Regression Dependency on a Subset (PRDS) [Benjamini and Yekutieli, 2001]. PRDS is generally accepted as a reasonable assumption within differential gene expression studies [Goeman and Solari, 2014].

However, If the user is not satisfy with the results, it may snoop into the data, possibly selecting a subset of gene of interest. One typical approach involves setting a threshold on the

absolute $\log_2$-FC, facilitating the selection of genes that manifest the most significant expression differences between the two conditions under comparison. Importantly, recent comprehensive simulation studies by Ebrahimpoor and Goeman [2021] have demonstrated that this picking strategy often leads to inflated FDRs. In such circumstances, a post-hoc inference procedure can be used to provide confidence bounds for the FDP in arbitrary, and potentially data-driven, subsets of genes [Goeman and Solari, 2011, Ebrahimpoor and Goeman, 2021]. This tool is quite practical and well-suited to biologists' needs, despite its current underutilization. For these reasons, I have chosen to implement this feature in the DiffSegR R package, building upon the sanssouci R package Neuvial et al. [2022]. A description of how to use the post-hoc procedure in DiffSegR is outlined in the *Advanced tutorials* section of the associated Vignette (Section 6.4.2.2). I also plan to incorporate it into the comaturationTrackeR R package.

I will not delve further into the concept of post-hoc inference in this manuscript, but an excellent introduction can be found in [Enjalbert-Courrech and Neuvial, 2022].

6.4 Transcriptome-wide detection of expression differences

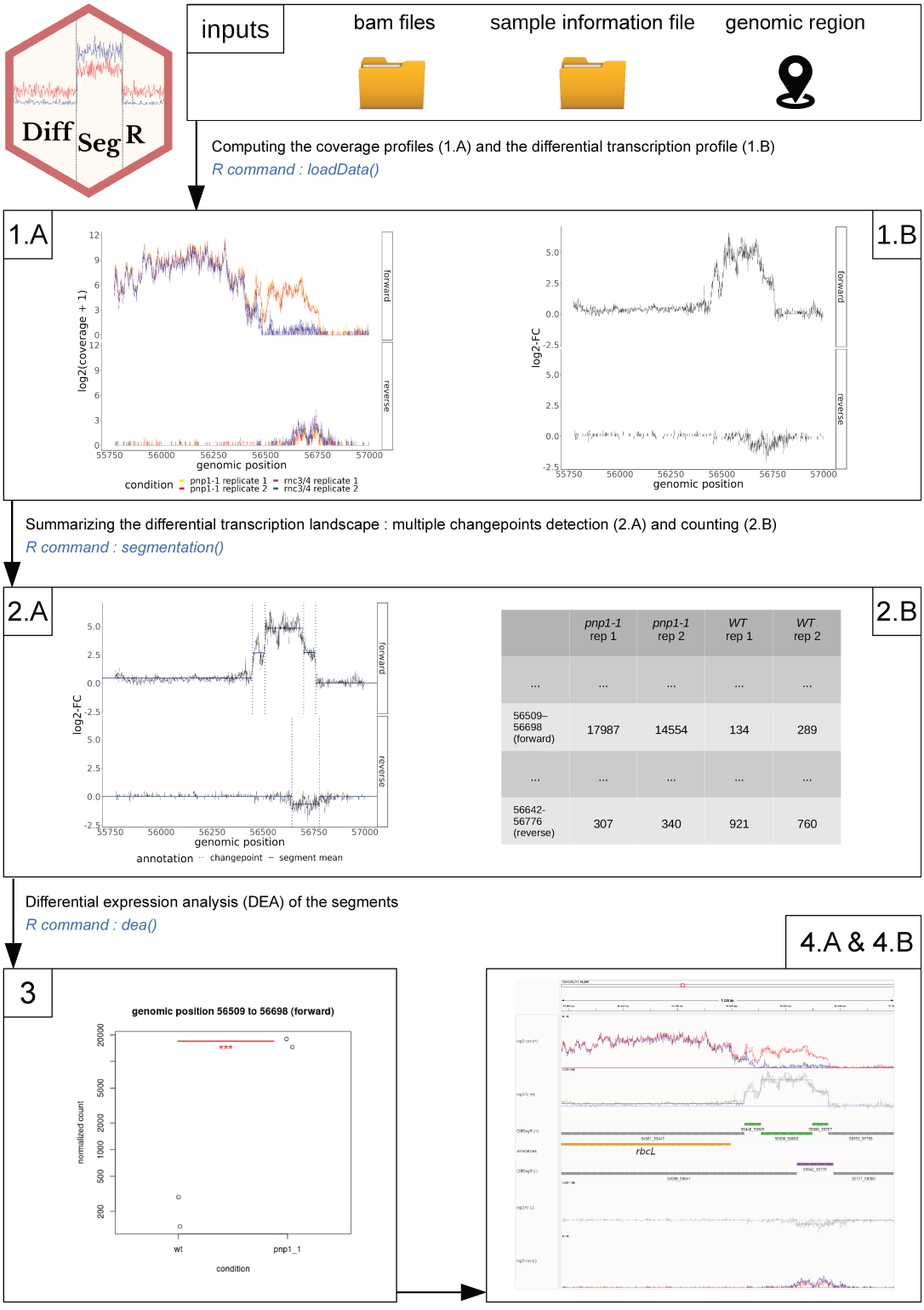
As previously mentioned in section 4.3, during my thesis I worked on the detection of DERs across the genome. In [Liehrmann et al., 2023], we introduced DiffSegR, a method that delineates candidates DERs within the $\log_2$-FC using FPOP (Baseline 2) and subsequently evaluates these regions using DESeq2. In the following two sections, I will first outline the statistical contrast that is tested using DESeq2 in DiffSegR, and then describe the four main stages of the DiffSegR method.

6.4.1 Contrast

Assume $\hat{\tau}$ is the set of changepoints estimated by FPOP on the per-base $\log_2$-FC calculated on an RNA-Seq experiment involving two biological conditions. Here, the j^{th} segment starts at position $\hat{\tau}_j + 1$ and ends at position $\hat{\tau}_{j+1}$. In the context of transcriptome-wide expression difference detection, each such segment is considered as a candidate DER.

We subsequently redefine $K_{j,d}$ as the number of sequencing reads overlapping the j^{th} candidate DER. Note that a single read may be assigned to multiple candidate DERs. The counts of candidate DERs can be modeled with (6.1) and (6.2). Candidate DERs can subsequently be assessed by testing the contrast (6.4).

Significantly, in three separate analyses conducted as detailed in Liehrmann et al. [2023], the dispersion trends observed in candidate DERs (Figures S2, S5 and S8) as well as the p-value histograms (Figures S3, S6 and S9) appeared regulars.



(4.A) Annotating the DERs with nearest using using user specified annotations or
(4.B) visualizing the DERs in IGV
R commands : annotateNearest() & exportResults()

FIGURE 6.3 – Schematic representation of the DiffSegR pipeline (figure from Liehrmann et al. [2023])

6.4.2 DiffSegR : An RNA-Seq data driven method for differential expression analysis using changepoint detection

6.4.2.1 DiffSegR in a nutshell

As illustrated in Figure 6.3, a classical differential expression analysis conducted using DiffSegR along the genome involves :

Computing the coverage profiles and the differential transcription profile. (1.A) Firstly, coverage profiles are generated from specified *BAM* files, which contain the aligned reads, and a user-determined genomic region. Individual coverage profiles for each strand are produced for every replicate of both biological conditions. **(1.B)** Following this, the per-base $\log_2$-FC for each strand is computed based on these coverage profiles.

Summarizing the differential transcription landscape. (2.A) FPOP is employed on the per-base log2-FC of each strand in order to identify segment boundaries. **(2.B)** Then, the featurecounts program [Liao et al., 2013] is utilized to assign mapped reads to these identified segments, leading to the creation of a count matrix.

Differential expression analysis. (3) DESeq2 is used to test the difference in average expression of each segment (candidate DERs) under the two compared biological conditions.

Annotating and visualizing the DERs. (4.A) The DERs are annotated based on user-specified annotations file (in *gff3* or *gtf* format). **(4.B)** Concurrently, data for DERs, non-DERs, segmentations, the mean of coverage profiles from both biological conditions, and per-base $\log_2$-FC are saved in formats that are compatible with genome viewers such as the Integrative Genomics Viewe (IGV). An IGV session in *XML* format is also created, which allows all tracks to be loaded simultaneously, thereby providing a user-friendly way to visualize and interpret DiffSegR results.

6.4.2.2 DiffSegR R package

I encapsulated the implementation of DiffSegR in an R package named after the method itself. The DiffSegR package is available on GitHub : `https://aliehrmann.github.io/DiffSegR/index.html`. Importantly, this package includes a Vignette which shows a minimal example on how to use the main functions, and then delves into a more advanced uses of DiffSegR. This Vignette should be considered as an extension of this manuscript.

Box 4: Section switch

By this stage, readers who have read the introduction of Chapter 5 should have a good understanding of how DiffSegR identifies homogeneous segments of the $\log_2$-FC along the genome using FPOP. After reading the first sections of the current chapter, they should now have a solid theoretical knowledge on the differential expression analysis of segments with DESeq2. Interested readers can now proceed to read [Liehrmann et al., 2023] (Appendix C). In addition to other technical details on DiffSegR, they will find a benchmark of DiffSegR with other methods from bioinformatics literature, on two plant RNA-Seq datasets that were previously used in conjunction with molecular biology techniques to decipher the roles of the chloroplast ribonucleases PNPase and Mini-III. Notably, I demonstrated that DiffSegR is the only method capable of retrieving all segments known to differentially accumulate outside of the annotated genic regions (3' and 5' extensions, antisense). I also present encouraging results from the application of DiffSegR to the *Bacillus subtilis* transcriptome.

6.5 Transcriptome-wide detection of co-maturations

The co-maturations can also be assessed using DESeq2.

6.5.0.1 A few words on comaturationTrackeR

In 2021, along with Chloé Seyman and Guillem Rigaill, I developed a novel approach, comaturationTrackeR. This tool utilizes RNA-seq data to detect co-maturations, provided that both events are cover by the same read. The initial version of comaturationTrackeR was completed and published as a pipeline [Guilcher et al., 2021]. The following year, I collaborated with Benjamin Vacus and Guillem Rigaill on a second version, which is presently under development and presented as an R package. Without going into exhaustive details, both iterations of comaturationTrackeR rely on a homemade function which annotates the reads by registering the maturation state of each user-provided event they cover. Following this, the statistical dependence of each pair of event is evaluated based on a Fisher's exact test for the first version of the method, and by testing the nullity of the classical "difference in differences" [2] estimator that we estimate using the GLM model of DESeq2 in the second version.

2. https://en.wikipedia.org/wiki/Difference_in_differences

Box 5: Section switch

At this stage, readers can delve into Guilcher et al. [2021], where we have highlighted the co-maturation of 42 pairs of splicing and editing sites in the chloroplast of *A. thaliana* (wild type), along with a preferred chronology where splicing typically occurs post editing at most sites. The analyses undertaken in this study rely on the comaturationtrackeR method. Comprehensive details regarding the initial and subsequent versions of comaturationtrackeR are provided in the bachelor's thesis of Chloé Seyman (Appendix D.2) and the master's thesis of Benjamin Vacus (Appendix D.3), respectively. Together with Benjamin, we further explored co-maturations in a PNPase mutant of *A. thaliana* and the dependencies among triple event sets.

Chapter 7

Perspectives

7.1 Ms.FPOP

7.1.1 Implementation of a more efficient update rule

An effective reduction of the living set of a candidate changepoint s presupposes comparing it with one or more candidate changepoints s' introduced after it (future candidates). Indeed, as intuitively shown in Figure 5.4.B and more formally in inclusion (15) of Liehrmann and Rigaill [2023], the interval over which s has a lower cost solution than s' decreases with t. However, it is evident that this operation, carried out at each iteration, comes at a cost. Alternatively, when comparing changepoint candidates s' to s, on top of computing the current bound of the intervals, one could computes and stores the iteration t_{empty} at which s' would lead to an empty intersection with s. Assuming this value is stored we can discard the interval as soon as the current iteration is larger than t_{empty}.

7.1.2 Further simulations

By employing signals simulated under Gaussian noise without changepoints, we have demonstrated that it is feasible to calibrate the multiscale penalty such that Ms.FPOP does not yield an excessive number of false positives (below 5% in our calibration). Under this framework, we evaluated Ms.FPOP against FPOP (which implements the LSC with the BIC penalty) in various scenarios. Ms.FPOP outperformed FPOP in segmentations with well-spread changepoints. In addition, Ms.FPOP was at least on par with FPOP for smaller segments within large enough profiles ($n \geq 10^4$). It is anticipated that a comparison with the penalty proposed in Lebarbier [2005] (known to have better statistical properties than the BIC penalty) will yield similar results [Verzelen et al., 2020]. However, this hypothesis remains to be confirmed using similar simulations. Additionally, I could compare Ms.FPOP with further proposed approaches to detecting changes in mean : R-FPOP Fearnhead and Rigaill [2018], WBS Fryzlewicz [2014], IDetect Anastasiou and Fryzlewicz [2021].

7.1.3 Applying Ms.FPOP to genomic series

As mentioned above, by using the multiscale penalty implemented in Ms.FPOP, we can enhance the detection power for fairly large segments, in comparison to the BIC penalty. Hence, when this methodology is applied to empirical data, it allows for the identification of these segments in the results of a more noisy RNA-Seq experiment (or in any other experiments where results can be aligned along the genome). Moreover, by leveraging Ms.FPOP in methods like DiffSegR, which segment the average coverage profile from multiple biological replicates, we could reasonably anticipate achieving equally good segmentation with fewer replicates.

However, the applicability of Ms.FPOP, with its current calibration, to real data is yet to be substantiated. Moreover, all our simulations have been conducted on signals with known variance, which is not typically the case in most real-world applications, like in genomics. As indicated in Section 5.3.2, it becomes necessary to derive the variance directly from the data. Several estimators and heuristic approaches for this purpose are available [Hall et al., 1990, Arlot et al., 2019].

I intend to compare the results of FPOP (currently implemented in DiffSegR) and Ms.FPOP in identifying candidate DERs based on the chloroplast RNA-Seq data, and the associated biological labels, that I used in Liehrmann et al. [2023]. With the help of biologists of the Organellar Gene Expression team we will scan the segmented profiles in IGV to assess the goodness of each segmentation.

Finally, I plan to compare Ms.FPOP with FPOP and other multiple changepoint detection methods on annotated datasets of DNA copy number variation [Hocking et al., 2013b] and ChIP-Seq [Hocking et al., 2016]. For the latter I will reuse the simulations from Liehrmann et al. [2021].

7.2 DiffSegR

7.2.1 Challenge in analyzing larger genomes with increased zeroes

Results from Liehrmann et al. [2023] (Section *DiffSegR can be used on sparser genomes*) suggest that DiffSegR is effective and powerful at detecting DERs in bacteria RNA-Seq datasets. Compared to the chloroplast, the coverage profiles computed on this bacterial dataset contain many more genomic positions with 0 counts. The assumption of a constant per-base $\log_2$-FC variance is less likely to hold in these case, thereby challenging the assumption of the standard changepoints model. As a result, the per-base $\log_2$-FC may be over-segmented and the resulting DERs may be less interpretable (Figure S38 of Liehrmann et al. [2023]). This problem is likely to be more severe on larger genomes, such as nuclear genomes.

A rather straightforward solution to the issue of low coverage is to apply DiffSegR to smaller chunks of the genome that have sufficient coverage. This is not as easy as it might seem. Indeed, (i) identifying those chunks is a segmentation problem itself, (ii) one ends up with multiple chunk and thus several multiple changepoint detection problems complexifying the model selection, and

(iii) we get a triple-dipping a problem as the data is used three times to recover the chunks, detect changes within the chunks, and tests segments within the chunks.

An alternative route would be to integrate more advanced segmentation methods, available in the statistical literature, in DiffSegR. In particular, it might make sense (i) to weight the base pair according to its coverage (using a weighted version of FPOP [Rigaill, 2022]), (ii) to consider full length reads [1] at the prize of modeling auto-correlation [Romano et al., 2021], and (iii) to model the discrete nature of the data using a negative binomial model [Cleynen and Lebarbier, 2014a].

7.2.2 Complex designs

In DiffSegR we only consider a simple RNA-seq experimental design with two conditions. In that case it is rather natural to segment the per-base $\log_2$-FC. For more complex design one could consider various contrasts. For example, consider a two-way anova design with two factors : lineage (wild type or mutant) and stress (standard or heat). In this experiment, one can be interested for example in :

1. the effect of the lineage irrespective of the stress condition ;
2. the effect of the stress irrespective of the lineage ;
3. the effect of the stress in the wild type lineage ;
4. the effect of the stress in the mutant ;
5. the effect of the lineage in the standard condition ;
6. the effect of the lineage in the heat condition ;
7. the interaction between the two factors Lambert et al. [2020].

If someone has a specific interest in a particular contrast, it make sense to define the signal to segment based on this contrast and then use DESeq2 on the resulting segments.

In reality, it is probable that one's interest extends to multiple contrasts, not just a single one. A straightforward solution is to run the DiffSegR analysis on each of these contrasts of interest, following by the correction of all the contrasts tested. Alternatively, an option could be to segment the signals corresponding to these multiple contrasts jointly. GeomFPOP [Pishchagina et al., 2023], a segmentation algorithm for multidimensional signals, allows to solve this problem exactly within a reasonable timeframe (a few minutes) for four contrasts and signals of size 10^5, which is approximately the size of the chloroplast genome.

7.2.3 Applying the diffsegR strategy to other genomic series

The DiffSegR strategy, which involves segmenting with FPOP, testing with DESeq2's negative binomial GLM, and then controlling multiple tests with a post-hoc bound, is relatively versatile. Furthermore, the tools employed at each step are statistically rigorous and robust.

1. see Note S1-2 of Liehrmann et al. [2023]

Ultimately, from an application perspective, the primary decision lies in selecting the signal that FPOP should segment, and eventually the contrast evaluated in DESeq2. As explained below results on RNA-Seq (chloroplast, bacteria) and RNA Immunoprecipitation Sequencing (RIP-Seq) (mitochondria) data lead me to believe that this versatility does not compromise the relevance of the biological events identified.

In [Liehrmann et al., 2023], we demonstrated that DiffSegR is proficient in accurately identifying 3' and 5' extensions of transcripts, as well as the accumulation of antisense RNAs and introns in two *A. thaliana* mutants for chloroplast ribonucleases—Mini-III [Hotto et al., 2015] and PNPase [Castandet et al., 2013]. As previously mentioned, we also showed that it could successfully find all potential candidates for direct degradation by Rae1 in *B. subtilis*. The candidates and two confirmed sites were previously identified by Leroy et al. [2017] and Deves et al. [2023].

Furthermore, in collaboration with Huy Cuong Tran (PhD student, Lund University, Sweden) and Olivier Van Aken (Associate Professor, Lund University, Sweden), we utilized DiffSegR on RIP-Seq data to establish that a protein under study has a binding affinity towards 5' untranslated regions [Tran et al., 2023]. This collaboration strengthened my conviction that DiffSegR holds potential for application across a wide range of RNA-Seq based strategies aimed at capturing specific biological events [Han et al., 2015].

For instance, it could be used to detect newly transcribed RNAs compared to mature RNA controls in nascent RNA analysis [Wissink et al., 2019], discern differences in ribosome-bound RNA in translatome analysis [Calviello and Ohler, 2017], or to distinguish structured (double-stranded RNA) from unstructured RNAs in structurome analysis [Kertesz et al., 2010], to mention just a few possibilities.

7.3 Coordination of chloroplast RNA maturation events

Leveraging a dedicated Nanopore long-read protocol [Guilcher et al., 2021], we sequenced the chloroplast transcripts of *A. thaliana* under normal growth conditions. This sequencing data was subsequently analyzed using the initial version of comaturationTrackeR, revealing dependencies between 42 pairs of editing and intron splicing sites. Some of these dependencies had been previously documented in scientific literature. Furthermore, our findings elucidated a preferential sequence of maturation events, wherein splicing generally transpired subsequent to the editing of most sites [Guilcher et al., 2021]. This investigation represents a pioneering study exploring the coordination of chloroplast RNA maturation events at transcriptome-scale.

However, in its current form, comaturationTrackeR is not equipped to analyze dependencies between the 5'/3' ends of transcripts and other maturation events. Without committing to a specific methodology, this feature is eagerly anticipated by the biologists, including those from the Organellar Gene Expression team.

Long term perspectives. In the context of co-maturation predictions by comaturationTrackeR, questions arise of how to validate the list of identified co-maturations and decipher the molecular mechanisms underlying the observed co-maturations. The Organellar Gene Expression team has proposed a potential validation strategy. This strategy involves conducting experiments with a mutant specific to a particular event, to investigate its impact on other dependent events.

Bibliography

F. Abbas-Aghababazadeh, Q. Li, and B. L. Fridley. Comparison of normalization approaches for gene expression studies completed with high-throughput sequencing. *PLOS ONE*, 13(10) : e0206312, Oct. 2018. doi : 10.1371/journal.pone.0206312. URL https://doi.org/10.1371/journal.pone.0206312.

V. Agarwal, S. Lopez-Darwin, D. R. Kelley, and J. Shendure. The landscape of alternative polyadenylation in single cells of the developing mouse embryo. *Nature Communications*, 12(1), Aug. 2021. doi : 10.1038/s41467-021-25388-8. URL https://doi.org/10.1038/s41467-021-25388-8.

Y. Aloni, R. Dhar, O. Laub, M. Horowitz, and G. Khoury. Novel mechanism for RNA maturation : the leader sequences of simian virus 40 mRNA are not transcribed adjacent to the coding sequences. *Proceedings of the National Academy of Sciences*, 74(9) :3686–3690, Sept. 1977. doi : 10.1073/pnas.74.9.3686. URL https://doi.org/10.1073/pnas.74.9.3686.

A. Anastasiou and P. Fryzlewicz. Detecting multiple generalized change-points by isolating single ones. *Metrika*, 85(2) :141–174, May 2021. doi : 10.1007/s00184-021-00821-6. URL https://doi.org/10.1007/s00184-021-00821-6.

S. Anders, A. Reyes, and W. Huber. Detecting differential usage of exons from RNA-seq data. *Genome Research*, 22(10) :2008–2017, June 2012. doi : 10.1101/gr.133744.111. URL https://doi.org/10.1101/gr.133744.111.

D. Anderson and K. Burnham. Model selection and multi-model inference. *Second. NY : Springer-Verlag*, 63(2020) :10, 2004.

C. Angelini, D. D. Canditiis, and I. D. Feis. Computational approaches for isoform detection and estimation : good and bad news. *BMC Bioinformatics*, 15(1), May 2014. doi : 10.1186/1471-2105-15-135. URL https://doi.org/10.1186/1471-2105-15-135.

F. J. Anscombe. The transformation of poisson, binomial and negative-binomial data. *Biometrika*, 35(3/4) :246–254, 1948.

J. M. Archibald. The puzzle of plastid evolution. *Current Biology*, 19(2) :R81–R88, Jan. 2009. doi : 10.1016/j.cub.2008.11.067. URL https://doi.org/10.1016/j.cub.2008.11.067.

S. Arlot, A. Celisse, and Z. Harchaoui. A kernel multiple change-point algorithm via model selection. *Journal of machine learning research*, 20(162), 2019.

H. Ashoor, A. Hérault, A. Kamoun, F. Radvanyi, V. B. Bajic, E. Barillot, and V. Boeva. HMCan : a method for detecting chromatin modifications in cancer samples using ChIP-seq data. *Bioinformatics*, 29(23) :2979–2986, Sept. 2013. doi : 10.1093/bioinformatics/btt524. URL https://doi.org/10.1093/bioinformatics/btt524.

I. E. Auger and C. E. Lawrence. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology*, 51(1) :39–54, Jan. 1989. doi : 10.1007/bf02458835. URL https://doi.org/10.1007/bf02458835.

J. Bai. Estimation of a change point in multiple regression models. *Review of Economics and Statistics*, 79(4) :551–563, 1997.

F. E. Baralle and J. Giudice. Alternative splicing as a regulator of development and tissue identity. *Nature Reviews Molecular Cell Biology*, 18(7) :437–451, May 2017. doi : 10.1038/nrm.2017.27. URL https://doi.org/10.1038/nrm.2017.27.

Y. Baraud, C. Giraud, and S. Huet. Gaussian model selection with an unknown variance. *The Annals of Statistics*, 37(2), Apr. 2009. doi : 10.1214/07-aos573. URL https://doi.org/10.1214/07-aos573.

A. C. Barbrook, C. J. Howe, and S. Purton. Why are plastid genomes retained in non-photosynthetic organisms ? *Trends in Plant Science*, 11(2) :101–108, Feb. 2006. doi : 10.1016/j.tplants.2005.12.004. URL https://doi.org/10.1016/j.tplants.2005.12.004.

A. Barkan and I. Small. Pentatricopeptide repeat proteins in plants. *Annual Review of Plant Biology*, 65(1) :415–442, Apr. 2014. doi : 10.1146/annurev-arplant-050213-040159. URL https://doi.org/10.1146/annurev-arplant-050213-040159.

P. Batut, A. Dobin, C. Plessy, P. Carninci, and T. R. Gingeras. High-fidelity promoter profiling reveals widespread alternative promoter usage and transposon-driven developmental gene expression. *Genome Research*, 23(1) :169–180, Aug. 2012. doi : 10.1101/gr.139618.112. URL https://doi.org/10.1101/gr.139618.112.

K. Baudry. *L'éditosome du chloroplaste : questions, éléments de réponses et digressions*. PhD thesis, Université Paris Saclay (COmUE), 2019. NNT : 2019SACLE036. ⟨tel-03207729⟩.

M. Behm and M. Öhman. RNA editing : A contributor to neuronal dynamics in the mammalian brain. *Trends in Genetics*, 32(3) :165–175, Mar. 2016. doi : 10.1016/j.tig.2015.12.005. URL https://doi.org/10.1016/j.tig.2015.12.005.

R. E. Bellman and B. Kotkin. *On the Approximation of Curves by Line Segments Using Dynamic Programming II*. RAND Corporation, Santa Monica, CA, 1962.

S. Ben-Aroya and E. Y. Levanon. A-to-i RNA editing : An overlooked source of cancer mutations. *Cancer Cell*, 33(5) :789–790, May 2018. doi : 10.1016/j.ccell.2018.04.006. URL https://doi.org/10.1016/j.ccell.2018.04.006.

B. A. Benayoun, E. A. Pollina, D. Ucar, S. Mahmoudi, K. Karra, E. D. Wong, K. Devarajan, A. C. Daugherty, A. B. Kundaje, E. Mancini, B. C. Hitz, R. Gupta, T. A. Rando, J. C. Baker, M. P. Snyder, J. M. Cherry, and A. Brunet. H3k4me3 breadth is linked to cell identity and transcriptional consistency. *Cell*, 158(3) :673–688, July 2014. doi : 10.1016/j.cell.2014.06.027. URL https://doi.org/10.1016/j.cell.2014.06.027.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate : a practical and powerful approach to multiple testing. *Journal of the Royal statistical society : series B (Methodological)*, 57(1) :289–300, 1995.

Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188, 2001.

K. Berge, K. M. Hembach, C. Soneson, S. Tiberi, L. Clement, M. I. Love, R. Patro, and M. D. Robinson. RNA sequencing data : Hitchhiker's guide to expression analysis. *Annual Review of Biomedical Data Science*, 2(1) :139–173, July 2019. doi : 10.1146/annurev-biodatasci-072018-021255. URL https://doi.org/10.1146/annurev-biodatasci-072018-021255.

S. M. Berget, C. Moore, and P. A. Sharp. Spliced segments at the 5′ terminus of adenovirus 2 late mRNA. *Proceedings of the National Academy of Sciences*, 74(8) :3171–3175, Aug. 1977. doi : 10.1073/pnas.74.8.3171. URL https://doi.org/10.1073/pnas.74.8.3171.

E. Bernard, L. Jacob, J. Mairal, and J.-P. Vert. Efficient RNA isoform identification and quantification from RNA-seq data with network flows. *Bioinformatics*, 30(17) :2447–2455, May 2014. doi : 10.1093/bioinformatics/btu317. URL https://doi.org/10.1093/bioinformatics/btu317.

S. A. Bhuiyan, S. Ly, M. Phan, B. Huntington, E. Hogan, C. C. Liu, J. Liu, and P. Pavlidis. Systematic evaluation of isoform function in literature reports of alternative splicing. *BMC Genomics*, 19(1), Aug. 2018. doi : 10.1186/s12864-018-5013-2. URL https://doi.org/10.1186/s12864-018-5013-2.

L. Birgé and P. Massart. Gaussian model selection. *Journal of the European Mathematical Society*, 3(3) :203–268, Aug. 2001. doi : 10.1007/s100970100031. URL https://doi.org/10.1007/s100970100031.

D. L. Black. Mechanisms of alternative pre-messenger RNA splicing. *Annual Review of Biochemistry*, 72(1) :291–336, June 2003. doi : 10.1146/annurev.biochem.72.121801.161720. URL https://doi.org/10.1146/annurev.biochem.72.121801.161720.

S. C. Bonnal, I. López-Oreja, and J. Valcárcel. Roles and mechanisms of alternative splicing in cancer implications for care. *Nature Reviews Clinical Oncology*, 17(8) :457–474, Apr. 2020. doi : 10.1038/s41571-020-0350-x. URL https://doi.org/10.1038/s41571-020-0350-x.

L. Boysen, A. Kempe, V. Liebscher, A. Munk, and O. Wittich. Consistencies and rates of convergence of jump-penalized least squares estimators. 2009.

D. Branton, D. W. Deamer, A. Marziali, H. Bayley, S. A. Benner, T. Butler, M. Di Ventra, S. Garaj, A. Hibbs, X. Huang, et al. The potential and challenges of nanopore sequencing. *Nature biotechnology*, 26(10) :1146–1153, 2008.

N. L. Bray, H. Pimentel, P. Melsted, and L. Pachter. Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology*, 34(5) :525–527, Apr. 2016. doi : 10.1038/nbt.3519. URL https://doi.org/10.1038/nbt.3519.

R. Breathnach, J. L. Mandel, and P. Chambon. Ovalbumin gene is split in chicken DNA. *Nature*, 270(5635) :314–319, Nov. 1977. doi : 10.1038/270314a0. URL https://doi.org/10.1038/270314a0.

B. B. Buchanan, W. Gruissem, and R. L. Jones. *Biochemistry and molecular biology of plants*. John wiley & sons, 2015.

I. Buchumenski, K. Holler, L. Appelbaum, E. Eisenberg, J. P. Junker, and E. Y. Levanon. Systematic identification of a-to-i RNA editing in zebrafish development and adult organs. *Nucleic Acids Research*, 49(8) :4325–4337, Apr. 2021. doi : 10.1093/nar/gkab247. URL https://doi.org/10.1093/nar/gkab247.

A. Byrne, A. E. Beaudin, H. E. Olsen, M. Jain, C. Cole, T. Palmer, R. M. DuBois, E. C. Forsberg, M. Akeson, and C. Vollmers. Nanopore long-read RNAseq reveals widespread transcriptional variation among the surface receptors of individual b cells. *Nature Communications*, 8(1), July 2017. doi : 10.1038/ncomms16027. URL https://doi.org/10.1038/ncomms16027.

L. Calviello and U. Ohler. Beyond read-counts : Ribo-seq data analysis to understand the functions of the transcriptome. *Trends in Genetics*, 33(10) :728–744, Oct. 2017. doi : 10.1016/j.tig.2017.08.003. URL https://doi.org/10.1016/j.tig.2017.08.003.

B. Castandet, A. M. Hotto, Z. Fei, and D. B. Stern. Strand-specific RNA sequencing uncovers chloroplast ribonuclease functions. *FEBS Letters*, 587(18) :3096–3101, Aug. 2013. doi : 10.1016/j.febslet.2013.08.004. URL https://doi.org/10.1016/j.febslet.2013.08.004.

B. Castandet, A. Germain, A. M. Hotto, and D. B. Stern. Systematic sequencing of chloroplast transcript termini from arabidopsis thaliana reveals >200 transcription initiation sites and the extensive imprints of RNA-binding proteins and secondary structures. *Nucleic Acids Research*, Nov. 2019. doi : 10.1093/nar/gkz1059. URL https://doi.org/10.1093/nar/gkz1059.

Y. Choquet, M. Goldschmidt-Clermont, J. Girard-Bascou, U. Kück, P. Bennoun, and J.-D. Rochaix. Mutant phenotypes support a trans-splicing mechanism for the expression of the tripartite psaA gene in the c. reinhardtii chloroplast. *Cell*, 52(6) :903–913, Mar. 1988. doi : 10.1016/0092-8674(88)90432-1. URL https://doi.org/10.1016/0092-8674(88)90432-1.

A. Cleynen and E. Lebarbier. Segmentation of the poisson and negative binomial rate models : a penalized estimator. *ESAIM : Probability and Statistics*, 18 :750–769, 2014a. doi : 10.1051/ps/2014005. URL https://doi.org/10.1051/ps/2014005.

A. Cleynen, S. Dudoit, and S. Robin. Comparing segmentation methods for genome annotation based on rna-seq data. *Journal of Agricultural, Biological, and Environmental Statistics*, 19 : 101–118, 2014b.

B. Clouet-d'Orval, M. Batista, M. Bouvier, Y. Quentin, G. Fichant, A. Marchfelder, and L.-K. Maier. Insights into RNA-processing pathways and associated RNA-degrading enzymes in archaea. *FEMS Microbiology Reviews*, 42(5) :579–613, Apr. 2018. doi : 10.1093/femsre/fuy016. URL https://doi.org/10.1093/femsre/fuy016.

L. Collado-Torres, A. Nellore, A. C. Frazee, C. Wilks, M. I. Love, B. Langmead, R. A. Irizarry, J. T. Leek, and A. E. Jaffe. Flexible expressed region analysis for RNA-seq with derfinder. *Nucleic Acids Research*, 45(2) :e9–e9, Sept. 2016. doi : 10.1093/nar/gkw852. URL https://doi.org/10.1093/nar/gkw852.

C. Condon. RNA processing and degradation in bacillus subtilis. *Microbiology and Molecular Biology Reviews*, 67(2) :157–174, June 2003. doi : 10.1128/mmbr.67.2.157-174.2003. URL https://doi.org/10.1128/mmbr.67.2.157-174.2003.

N. R. Council et al. *Frontiers in massive data analysis*. National Academies Press, 2013.

P. Cramer, C. G. Pesce, F. E. Baralle, and A. R. Kornblihtt. Functional association between promoter structure and transcript alternative splicing. *Proceedings of the National Academy of Sciences*, 94(21) :11456–11460, Oct. 1997. doi : 10.1073/pnas.94.21.11456. URL https://doi.org/10.1073/pnas.94.21.11456.

F. Crick. On protein synthesis. *Symp Soc Exp Biol*, 12 :138–63, 1958.

F. Crick. Central dogma of molecular biology. *Nature*, 227(5258) :561–563, Aug. 1970. doi : 10.1038/227561a0. URL https://doi.org/10.1038/227561a0.

N. Cvetesic, H. G. Leitch, M. Borkowska, F. Müller, P. Carninci, P. Hajkova, and B. Lenhard. SLIC-CAGE : high-resolution transcription start site mapping using nanogram-levels of total RNA. *Genome Research*, 28(12) :1943–1956, Nov. 2018. doi : 10.1101/gr.235937.118. URL https://doi.org/10.1101/gr.235937.118.

G. B. Danks, P. Navratilova, B. Lenhard, and E. M. Thompson. Distinct core promoter codes drive transcription initiation at key developmental transitions in a marine chordate. *BMC Genomics*, 19(1), Feb. 2018. doi : 10.1186/s12864-018-4504-5. URL https://doi.org/10.1186/s12864-018-4504-5.

R. A. Davis, T. C. M. Lee, and G. A. Rodriguez-Yam. Structural break estimation for nonstationary time series models. *Journal of the American Statistical Association*, 101(473) : 223–239, Mar. 2006. doi : 10.1198/016214505000000745. URL https://doi.org/10.1198/016214505000000745.

R. V. Davuluri, Y. Suzuki, S. Sugano, C. Plass, and T. H.-M. Huang. The functional consequences of alternative promoter use in mammalian genomes. *Trends in Genetics*, 24(4) :167–177, Apr. 2008. doi : 10.1016/j.tig.2008.01.008. URL https://doi.org/10.1016/j.tig.2008.01.008.

D. Demircioğlu, E. Cukuroglu, M. Kindermans, T. Nandi, C. Calabrese, N. A. Fonseca, A. Kahles, K.-V. Lehmann, O. Stegle, A. Brazma, A. N. Brooks, G. Rätsch, P. Tan, and J. Göke. A pan-cancer transcriptome analysis reveals pervasive regulation through alternative promoters. *Cell*, 178(6) :1465–1477.e17, Sept. 2019. doi : 10.1016/j.cell.2019.08.018. URL https://doi.org/10.1016/j.cell.2019.08.018.

D. Deshpande, K. Chhugani, Y. Chang, A. Karlsberg, C. Loeffler, J. Zhang, A. Muszyńska, V. Munteanu, H. Yang, J. Rotman, L. Tao, B. Balliu, E. Tseng, E. Eskin, F. Zhao, P. Mohammadi, P. P. Łabaj, and S. Mangul. RNA-seq data science : From raw data to effective interpretation. *Frontiers in Genetics*, 14, Mar. 2023. doi : 10.3389/fgene.2023.997383. URL https://doi.org/10.3389/fgene.2023.997383.

V. Deves, A. Trinquier, L. Gilet, J. Alharake, M. Leroy, C. Condon, and F. Braun. Identification of a new substrate for the ribosome associated endoribonuclease rae1 reveals a link to the b. subtilis response and sensitivity to chloramphenicol. Feb. 2023. doi : 10.1101/2023.02.16.528812. URL https://doi.org/10.1101/2023.02.16.528812.

I. W. Deveson, M. E. Brunck, J. Blackburn, E. Tseng, T. Hon, T. A. Clark, M. B. Clark, J. Crawford, M. E. Dinger, L. K. Nielsen, et al. Universal alternative splicing of noncoding exons. *Cell Systems*, 6(2) :245–255, 2018.

A. Diaz, K. Park, D. A. Lim, and J. S. Song. Normalization, bias correction, and peak calling for ChIP-seq. *Statistical Applications in Genetics and Molecular Biology*, 11(3), Jan. 2012. doi : 10.1515/1544-6115.1750. URL https://doi.org/10.1515/1544-6115.1750.

F. Dick, G. S. Nido, G. W. Alves, O.-B. Tysnes, G. H. Nilsen, C. Dölle, and C. Tzoulis. Differential transcript usage in the parkinson's disease brain. *PLOS Genetics*, 16(11) :e1009182, Nov. 2020. doi : 10.1371/journal.pgen.1009182. URL https://doi.org/10.1371/journal.pgen.1009182.

A. Dobin, C. A. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson, and T. R. Gingeras. STAR : ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1) : 15–21, Oct. 2012. doi : 10.1093/bioinformatics/bts635. URL https://doi.org/10.1093/bioinformatics/bts635.

M. T. Doel, M. Houghton, E. A. Cook, and N. H. Carey. The presence of ovalbumin mRNA coding sequences in multiple restriction fragments of chicken DNA. *Nucleic Acids Research*, 4(11) :3701–3714, 1977. doi : 10.1093/nar/4.11.3701. URL https://doi.org/10.1093/nar/4.11.3701.

I. Donkin and R. Barrès. Sperm epigenetics and influence of environmental factors. *Molecular Metabolism*, 14 :1–11, Aug. 2018. doi : 10.1016/j.molmet.2018.02.006. URL https://doi.org/10.1016/j.molmet.2018.02.006.

M. Ebrahimpoor and J. J. Goeman. Inflated false discovery rate due to volcano plots : problem and solutions. *Briefings in bioinformatics*, 22(5) :bbab053, 2021.

D. Eddelbuettel and R. François. Rcpp : Seamless r and c++ integration. *Journal of Statistical Software*, 40(8), 2011. doi : 10.18637/jss.v040.i08. URL https://doi.org/10.18637/jss.v040.i08.

E. Eisenberg and E. Y. Levanon. A-to-i RNA editing immune protector and transcriptome diversifier. *Nature Reviews Genetics*, 19(8) :473–490, Apr. 2018. doi : 10.1038/s41576-018-0006-1. URL https://doi.org/10.1038/s41576-018-0006-1.

N. Enjalbert-Courrech and P. Neuvial. Powerful and interpretable control of false discoveries in two-group differential expression studies. *Bioinformatics*, 38(23) :5214–5221, Oct. 2022. doi : 10.1093/bioinformatics/btac693. URL https://doi.org/10.1093/bioinformatics/btac693.

P. Fearnhead and G. Rigaill. Changepoint detection in the presence of outliers. *Journal of the American Statistical Association*, 114(525) :169–183, June 2018. doi : 10.1080/01621459.2017.1385466. URL https://doi.org/10.1080/01621459.2017.1385466.

P. Fearnhead and G. Rigaill. Relating and comparing methods for detecting changes in mean. *Stat*, 9(1), Jan. 2020. doi : 10.1002/sta4.291. URL https://doi.org/10.1002/sta4.291.

J. P. Fededa, E. Petrillo, M. S. Gelfand, A. D. Neverov, S. Kadener, G. Nogués, F. Pelisch, F. E. Baralle, A. F. Muro, and A. R. Kornblihtt. A polar mechanism coordinates different regions of alternative splicing within a single gene. *Molecular Cell*, 19(3) :393–404, Aug. 2005. doi : 10.1016/j.molcel.2005.06.035. URL https://doi.org/10.1016/j.molcel.2005.06.035.

A. P. Fejes, G. Robertson, M. Bilenky, R. Varhol, M. Bainbridge, and S. J. M. Jones. FindPeaks 3.1 : a tool for identifying areas of enrichment from massively parallel short-read sequencing technology. *Bioinformatics*, 24(15) :1729–1730, July 2008. doi : 10.1093/bioinformatics/btn305. URL https://doi.org/10.1093/bioinformatics/btn305.

V. Fey, R. Wagner, K. Braütigam, M. Wirtz, R. Hell, A. Dietzmann, D. Leister, R. Oelmüller, and T. Pfannschmidt. Retrograde plastid redox signals in the expression of nuclear genes for chloroplast proteins of arabidopsis thaliana. *Journal of Biological Chemistry*, 280(7) : 5318–5328, Feb. 2005. doi : 10.1074/jbc.m406358200. URL https://doi.org/10.1074/jbc.m406358200.

W. D. Fisher. On grouping for maximum homogeneity. *Journal of the American Statistical Association*, 53(284) :789–798, Dec. 1958. doi : 10.1080/01621459.1958.10501479. URL https://doi.org/10.1080/01621459.1958.10501479.

S. W. Flavell, T.-K. Kim, J. M. Gray, D. A. Harmin, M. Hemberg, E. J. Hong, E. Markenscoff-Papadimitriou, D. M. Bear, and M. E. Greenberg. Genome-wide analysis of MEF2 transcriptional program reveals synaptic target genes and neuronal activity-dependent polyadenylation site selection. *Neuron*, 60(6) :1022–1038, Dec. 2008. doi : 10.1016/j.neuron.2008.11.029. URL https://doi.org/10.1016/j.neuron.2008.11.029.

A. C. Frazee, S. Sabunciyan, K. D. Hansen, R. A. Irizarry, and J. T. Leek. Differential expression analysis of RNA-seq data at single-base resolution. *Biostatistics*, 15(3) :413–426, Jan. 2014. doi : 10.1093/biostatistics/kxt053. URL https://doi.org/10.1093/biostatistics/kxt053.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1) :1, 2010.

P. Fryzlewicz. Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6), Dec. 2014. doi : 10.1214/14-aos1245. URL https://doi.org/10.1214/14-aos1245.

T. S. Furey. ChIP–seq and beyond : new and improved methodologies to detect and characterize protein–DNA interactions. *Nature Reviews Genetics*, 13(12) :840–852, Oct. 2012. doi : 10.1038/nrg3306. URL https://doi.org/10.1038/nrg3306.

L. Gao, Y. Geng, B. Li, J. Chen, and J. Yang. Genome-wide DNA methylation alterations of alternanthera philoxeroides in natural and manipulated habitats : implications for epigenetic regulation of rapid responses to environmental fluctuation and phenotypic variation. *Plant, Cell Environment*, 33(11) :1820–1827, June 2010. doi : 10.1111/j.1365-3040.2010.02186.x. URL https://doi.org/10.1111/j.1365-3040.2010.02186.x.

D. Garreau and S. Arlot. Consistent change-point detection with kernels. *Electronic Journal of Statistics*, 12(2), Jan. 2018. doi : 10.1214/18-ejs1513. URL https://doi.org/10.1214/18-ejs1513.

N. H. Gehring and J.-Y. Roignant. Anything but ordinary – emerging splicing mechanisms in eukaryotic gene regulation. *Trends in Genetics*, 37(4) :355–372, Apr. 2021. doi : 10.1016/j.tig.2020.10.008. URL https://doi.org/10.1016/j.tig.2020.10.008.

A. Germain, S. Herlich, S. Larom, S. H. Kim, G. Schuster, and D. B. Stern. Mutational analysis of arabidopsis chloroplast polynucleotide phosphorylase reveals roles for both RNase PH core domains in polyadenylation, RNA 3ι-end maturation and intron degradation. *The Plant Journal*, 67(3) :381–394, May 2011. doi : 10.1111/j.1365-313x.2011.04601.x. URL https://doi.org/10.1111/j.1365-313x.2011.04601.x.

A. Germain, A. M. Hotto, A. Barkan, and D. B. Stern. RNA processing and decay in plastids. *Wiley Interdisciplinary Reviews : RNA*, 4(3) :295–316, Mar. 2013. doi : 10.1002/wrna.1161. URL https://doi.org/10.1002/wrna.1161.

D. C. D. Giammartino, K. Nishida, and J. L. Manley. Mechanisms and consequences of alternative polyadenylation. *Molecular Cell*, 43(6) :853–866, Sept. 2011. doi : 10.1016/j.molcel.2011.08.017. URL https://doi.org/10.1016/j.molcel.2011.08.017.

G. Gigerenzer and W. Gaissmaier. Heuristic decision making. *Annual Review of Psychology*, 62(1) :451–482, Jan. 2011. doi : 10.1146/annurev-psych-120709-145346. URL https://doi.org/10.1146/annurev-psych-120709-145346.

W. Gilbert. Why genes in pieces ? *Nature*, 271(5645) :501–501, Feb. 1978. doi : 10.1038/271501a0. URL https://doi.org/10.1038/271501a0.

M. A. Girshick and H. Rubin. A bayes approach to a quality control model. *The Annals of mathematical statistics*, 23(1) :114–125, 1952.

P. Glaus, A. Honkela, and M. Rattray. Identifying differentially expressed transcripts from RNA-seq data with biological variation. *Bioinformatics*, 28(13) :1721–1728, May 2012. doi : 10.1093/bioinformatics/bts260. URL https://doi.org/10.1093/bioinformatics/bts260.

J. Gleeson, A. Leger, Y. D. J. Prawer, T. A. Lane, P. J. Harrison, W. Haerty, and M. B. Clark. Accurate expression quantification from nanopore direct RNA sequencing with NanoCount. *Nucleic Acids Research*, 50(4) :e19–e19, Nov. 2021. doi : 10.1093/nar/gkab1129. URL https://doi.org/10.1093/nar/gkab1129.

J. J. Goeman and A. Solari. Multiple testing for exploratory research. *Statistical Science*, 26(4), Nov. 2011. doi : 10.1214/11-sts356. URL https://doi.org/10.1214/11-sts356.

J. J. Goeman and A. Solari. Multiple hypothesis testing in genomics. *Statistics in Medicine*, 33 (11) :1946–1978, Jan. 2014. doi : 10.1002/sim.6082. URL https://doi.org/10.1002/sim.6082.

A. Gohr, L. P. Iñiguez, A. Torres-Méndez, S. Bonnal, and M. Irimia. Insplico : effective computational tool for studying splicing order of adjacent introns genome-wide with short and long RNA-seq reads. *Nucleic Acids Research*, Apr. 2023. doi : 10.1093/nar/gkad244. URL https://doi.org/10.1093/nar/gkad244.

B. R. Graveley, A. N. Brooks, J. W. Carlson, M. O. Duff, J. M. Landolin, L. Yang, C. G. Artieri, M. J. van Baren, N. Boley, B. W. Booth, J. B. Brown, L. Cherbas, C. A. Davis, A. Dobin, R. Li, W. Lin, J. H. Malone, N. R. Mattiuzzo, D. Miller, D. Sturgill, B. B. Tuch, C. Zaleski, D. Zhang, M. Blanchette, S. Dudoit, B. Eads, R. E. Green, A. Hammonds, L. Jiang, P. Kapranov, L. Langton, N. Perrimon, J. E. Sandler, K. H. Wan, A. Willingham, Y. Zhang, Y. Zou, J. Andrews, P. J. Bickel, S. E. Brenner, M. R. Brent, P. Cherbas, T. R. Gingeras, R. A. Hoskins, T. C. Kaufman, B. Oliver, and S. E. Celniker. The developmental transcriptome of drosophila melanogaster. *Nature*, 471(7339) :473–479, Dec. 2010. doi : 10.1038/nature09715. URL https://doi.org/10.1038/nature09715.

B. D. Gregory, R. C. O'Malley, R. Lister, M. A. Urich, J. Tonti-Filippini, H. Chen, A. H. Millar, and J. R. Ecker. A link between RNA metabolism and silencing affecting arabidopsis development. *Developmental Cell*, 14(6) :854–866, June 2008. doi : 10.1016/j.devcel.2008.04.005. URL https://doi.org/10.1016/j.devcel.2008.04.005.

D. Griesemer, J. R. Xue, S. K. Reilly, J. C. Ulirsch, K. Kukreja, J. R. Davis, M. Kanai, D. K. Yang, J. C. Butts, M. H. Guney, J. Luban, S. B. Montgomery, H. K. Finucane, C. D. Novina, R. Tewhey, and P. C. Sabeti. Genome-wide functional screen of 3′UTR variants uncovers causal variants for human disease and evolution. *Cell*, 184(20) :5247–5260.e19, Sept. 2021. doi : 10.1016/j.cell.2021.08.025. URL https://doi.org/10.1016/j.cell.2021.08.025.

M. Guilcher, A. Liehrmann, C. Seyman, T. Blein, G. Rigaill, B. Castandet, and E. Delannoy. Full length transcriptome highlights the coordination of plastid transcript processing. *International Journal of Molecular Sciences*, 22(20) :11297, Oct. 2021. doi : 10.3390/ijms222011297. URL https://doi.org/10.3390/ijms222011297.

R. Guo, L. Zheng, J. W. Park, R. Lv, H. Chen, F. Jiao, W. Xu, S. Mu, H. Wen, J. Qiu, Z. Wang, P. Yang, F. Wu, J. Hui, X. Fu, X. Shi, Y. G. Shi, Y. Xing, F. Lan, and Y. Shi. BS69/ZMYND11 reads and connects histone h3.3 lysine 36 trimethylation-decorated chromatin to regulated pre-mRNA processing. *Molecular Cell*, 56(2) :298–310, Oct. 2014. doi : 10.1016/j.molcel.2014.08.022. URL https://doi.org/10.1016/j.molcel.2014.08.022.

V. S. Hahn, H. Knutsdottir, X. Luo, K. Bedi, K. B. Margulies, S. M. Haldar, M. Stolina, J. Yin, A. Y. Khakoo, J. Vaishnav, J. S. Bader, D. A. Kass, and K. Sharma. Myo-

cardial gene expression signatures in human heart failure with preserved ejection fraction. *Circulation*, 143(2) :120–134, Jan. 2021. doi : 10.1161/circulationaha.120.050498. URL https://doi.org/10.1161/circulationaha.120.050498.

P. Hall, J. Kay, and D. Titterington. Asymptotically optimal difference-based estimation of variance in nonparametric regression. *Biometrika*, 77(3) :521–528, 1990.

Y. Han, S. Gao, K. Muegge, W. Zhang, and B. Zhou. Advanced applications of RNA sequencing and challenges. *Bioinformatics and Biology Insights*, 9s1 :BBI.S28991, Jan. 2015. doi : 10.4137/bbi.s28991. URL https://doi.org/10.4137/bbi.s28991.

T. J. Hardcastle and K. A. Kelly. baySeq : Empirical bayesian methods for identifying differential expression in sequence count data. *BMC Bioinformatics*, 11(1), Aug. 2010. doi : 10.1186/1471-2105-11-422. URL https://doi.org/10.1186/1471-2105-11-422.

S. A. Hardwick, W. Hu, A. Joglekar, L. Fan, P. G. Collier, C. Foord, J. Balacco, S. Lanjewar, M. M. Sampson, F. Koopmans, A. D. Prjibelski, A. Mikheenko, N. Belchikov, J. Jarroux, A. B. Lucas, M. Palkovits, W. Luo, T. A. Milner, L. C. Ndhlovu, A. B. Smit, J. Q. Trojanowski, V. M. Y. Lee, O. Fedrigo, S. A. Sloan, D. Tombácz, M. E. Ross, E. Jarvis, Z. Boldogkői, L. Gan, and H. U. Tilgner. Single-nuclei isoform RNA sequencing unlocks barcoded exon connectivity in frozen brain tissue. *Nature Biotechnology*, 40(7) :1082–1092, Mar. 2022. doi : 10.1038/s41587-022-01231-3. URL https://doi.org/10.1038/s41587-022-01231-3.

A. Harmanci, J. Rozowsky, and M. Gerstein. MUSIC : identification of enriched regions in ChIP-seq experiments using a mappability-corrected multiscale signal processing framework. *Genome Biology*, 15(10), Oct. 2014. doi : 10.1186/s13059-014-0474-3. URL https://doi.org/10.1186/s13059-014-0474-3.

L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray. From molecular to modular cell biology. *Nature*, 402(S6761) :C47–C52, Dec. 1999. doi : 10.1038/35011540. URL https://doi.org/10.1038/35011540.

T. Hocking, G. Rigaill, J.-P. Vert, and F. Bach. Learning sparse penalties for change-point detection using max margin interval regression. In *International conference on machine learning*, pages 172–180. PMLR, 2013a.

T. Hocking, G. Rigaill, and G. Bourque. Peakseg : constrained optimal segmentation and supervised penalty learning for peak detection in count data. In *International Conference on Machine Learning*, pages 324–332. PMLR, 2015.

T. D. Hocking, G. Schleiermacher, I. Janoueix-Lerosey, V. Boeva, J. Cappo, O. Delattre, F. Bach, and J.-P. Vert. Learning smoothing models of copy number profiles using breakpoint annotations. *BMC Bioinformatics*, 14(1), May 2013b. doi : 10.1186/1471-2105-14-164. URL https://doi.org/10.1186/1471-2105-14-164.

T. D. Hocking, P. Goerner-Potvin, A. Morin, X. Shao, T. Pastinen, and G. Bourque. Optimizing ChIP-seq peak detectors using visual labels and supervised machine learning. *Bioinformatics*, 33(4) :491–499, Nov. 2016. doi : 10.1093/bioinformatics/btw672. URL https://doi.org/10.1093/bioinformatics/btw672.

T. D. Hocking, G. Rigaill, P. Fearnhead, and G. Bourque. Constrained dynamic programming and supervised penalty learning algorithms for peak detection in genomic data. *The Journal of Machine Learning Research*, 21(1) :3388–3427, 2020.

T. D. Hocking, G. Rigaill, P. Fearnhead, and G. Bourque. Generalized functional pruning optimal partitioning (GFPOP) for constrained changepoint detection in genomic data. *Journal of Statistical Software*, 101(10), 2022. doi : 10.18637/jss.v101.i10. URL https://doi.org/10.18637/jss.v101.i10.

D. Hong and S. Jeong. 3'UTR diversity : Expanding repertoire of RNA alterations in human mRNAs. *Molecules and Cells*, 46(1) :48–56, Jan. 2023. doi : 10.14348/molcells.2023.0003. URL https://doi.org/10.14348/molcells.2023.0003.

A. M. Hotto, R. J. Schmitz, Z. Fei, J. R. Ecker, and D. B. Stern. Unexpected diversity of chloroplast noncoding RNAs as revealed by deep sequencing of the arabidopsis transcriptome. *G3 Genes|Genomes|Genetics*, 1(7) :559–570, Dec. 2011. doi : 10.1534/g3.111.000752. URL https://doi.org/10.1534/g3.111.000752.

A. M. Hotto, B. Castandet, L. Gilet, A. Higdon, C. Condon, and D. B. Stern. Arabidopsis chloroplast mini-ribonuclease III participates in rRNA maturation and intron recycling. *The Plant Cell*, 27(3) :724–740, Feb. 2015. doi : 10.1105/tpc.114.134452. URL https://doi.org/10.1105/tpc.114.134452.

Y. Hu, L. Fang, X. Chen, J. F. Zhong, M. Li, and K. Wang. LIQA : long-read isoform quantification and analysis. *Genome Biology*, 22(1), June 2021. doi : 10.1186/s13059-021-02399-8. URL https://doi.org/10.1186/s13059-021-02399-8.

Y. Hu, A. Gouru, and K. Wang. DELongSeq for efficient detection of differential isoform expression from long-read RNA-seq data. *NAR Genomics and Bioinformatics*, 5(1), jan 2023. doi : 10.1093/nargab/lqad019. URL https://doi.org/10.1093%2Fnargab%2Flqad019.

A. Ishihama. Functional modulation of escherichia coli RNA polymerase. *Annual Review of Microbiology*, 54(1) :499–518, Oct. 2000. doi : 10.1146/annurev.micro.54.1.499. URL https://doi.org/10.1146/annurev.micro.54.1.499.

T. Iwagawa and S. Watanabe. Molecular mechanisms of h3k27me3 and h3k4me3 in retinal development. *Neuroscience Research*, 138 :43–48, Jan. 2019. doi : 10.1016/j.neures.2018.09.010. URL https://doi.org/10.1016/j.neures.2018.09.010.

B. Jackson, J. D. Scargle, D. Barnes, S. Arabhi, A. Alt, P. Gioumousis, E. Gwin, P. Sangtrakulcharoen, L. Tan, and T. T. Tsai. An algorithm for optimal partitioning of data on an interval. 2003. doi : 10.48550/ARXIV.MATH/0309285. URL https://arxiv.org/abs/math/0309285.

A. Johnson, J. Lewis, and B. ALBERTS. Molecular biology of the cell. 2002.

J. M. Johnson, S. Edwards, D. Shoemaker, and E. E. Schadt. Dark matter in the genome : evidence of widespread transcription detected by microarray tiling experiments. *Trends in Genetics*, 21(2) :93–102, Feb. 2005. doi : 10.1016/j.tig.2004.12.009. URL https://doi.org/10.1016/j.tig.2004.12.009.

N. A. Johnson. A dynamic programming algorithm for the fused lasso and l 0-segmentation. *Journal of Computational and Graphical Statistics*, 22(2) :246–260, 2013.

T. Juven-Gershon, J.-Y. Hsu, J. W. Theisen, and J. T. Kadonaga. The RNA polymerase II core promoter — the gateway to transcription. *Current Opinion in Cell Biology*, 20(3) :253–259, June 2008. doi : 10.1016/j.ceb.2008.03.003. URL https://doi.org/10.1016/j.ceb.2008.03.003.

A. Kalsotra and T. A. Cooper. Functional consequences of developmentally regulated alternative splicing. *Nature Reviews Genetics*, 12(10) :715–729, Sept. 2011. doi : 10.1038/nrg3052. URL https://doi.org/10.1038/nrg3052.

G. Karlebach, L. Carmody, J. C. Sundaramurthi, E. Casiraghi, P. Hansen, J. Reese, C. J. Mungall, G. Valentini, and P. N. Robinson. An algorithmic framework for isoform-specific functional analysis. May 2022. doi : 10.1101/2022.05.13.491897. URL https://doi.org/10.1101/2022.05.13.491897.

P. J. Keeling. The endosymbiotic origin, diversification and fate of plastids. *Philosophical Transactions of the Royal Society B : Biological Sciences*, 365(1541) :729–748, Mar. 2010. doi : 10.1098/rstb.2009.0103. URL https://doi.org/10.1098/rstb.2009.0103.

O. Kelemen, P. Convertini, Z. Zhang, Y. Wen, M. Shen, M. Falaleeva, and S. Stamm. Function of alternative splicing. *Gene*, 514(1) :1–30, Feb. 2013. doi : 10.1016/j.gene.2012.07.083. URL https://doi.org/10.1016/j.gene.2012.07.083.

M. Kertesz, Y. Wan, E. Mazor, J. L. Rinn, R. C. Nutter, H. Y. Chang, and E. Segal. Genome-wide measurement of RNA secondary structure in yeast. *Nature*, 467(7311) :103–107, Sept. 2010. doi : 10.1038/nature09322. URL https://doi.org/10.1038/nature09322.

M. R. Khan, R. J. Wellinger, and B. Laurent. Exploring the alternative splicing of long noncoding RNAs. *Trends in Genetics*, 37(8) :695–698, Aug. 2021. doi : 10.1016/j.tig.2021.03.010. URL https://doi.org/10.1016/j.tig.2021.03.010.

R. Killick, P. Fearnhead, and I. A. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500) :1590–1598, 2012.

D. Kim, B. Langmead, and S. L. Salzberg. HISAT : a fast spliced aligner with low memory requirements. *Nature Methods*, 12(4) :357–360, Mar. 2015. doi : 10.1038/nmeth.3317. URL https://doi.org/10.1038/nmeth.3317.

M. Kim, N. J. Krogan, L. Vasiljeva, O. J. Rando, E. Nedea, J. F. Greenblatt, and S. Buratowski. The yeast rat1 exonuclease promotes transcription termination by RNA polymerase II. *Nature*, 432(7016) :517–522, Nov. 2004. doi : 10.1038/nature03041. URL https://doi.org/10.1038/nature03041.

S. Kim, H. Kim, N. Fong, B. Erickson, and D. L. Bentley. Pre-mRNA splicing is a determinant of histone h3k36 methylation. *Proceedings of the National Academy of Sciences*, 108(33) : 13564–13569, Aug. 2011. doi : 10.1073/pnas.1109475108. URL https://doi.org/10.1073/pnas.1109475108.

S. K. Kim, J. Lund, M. Kiraly, K. Duke, M. Jiang, J. M. Stuart, A. Eizinger, B. N. Wylie, and G. S. Davidson. A gene expression map for caenorhabditis elegans. *Science*, 293(5537) :2087–2092, Sept. 2001. doi : 10.1126/science.1061603. URL https://doi.org/10.1126/science.1061603.

M. Kisielow, S. Kleiner, M. Nagasawa, A. Faisal, and Y. Nagamine. Isoform-specific knockdown and expression of adaptor protein shca using small interfering rna. *Biochemical Journal*, 363 (1) :1–5, 2002.

V. Knoop. When you can't trust the DNA : RNA editing changes transcript sequences. *Cellular and Molecular Life Sciences*, 68(4) :567–586, Oct. 2010. doi : 10.1007/s00018-010-0538-9. URL https://doi.org/10.1007/s00018-010-0538-9.

L. Koepcke, G. Ashida, and J. Kretzberg. Single and multiple change point detection in spike trains : Comparison of different CUSUM methods. *Frontiers in Systems Neuroscience*, 10, June 2016. doi : 10.3389/fnsys.2016.00051. URL https://doi.org/10.3389/fnsys.2016.00051.

S. Kovaka, S. Ou, K. M. Jenike, and M. C. Schatz. Approaching complete genomes, transcriptomes and epi-omes with accurate long-read sequencing. *Nature Methods*, 20(1) : 12–16, Jan. 2023. doi : 10.1038/s41592-022-01716-8. URL https://doi.org/10.1038/s41592-022-01716-8.

K. R. Kukurba and S. B. Montgomery. RNA sequencing and analysis. *Cold Spring Harbor Protocols*, 2015(11) :pdb.top084970, Apr. 2015. doi : 10.1101/pdb.top084970. URL https://doi.org/10.1101/pdb.top084970.

Y. Kurihara, Y. Makita, M. Kawashima, T. Fujita, S. Iwasaki, and M. Matsui. Transcripts from downstream alternative transcription start sites evade uORF-mediated inhibition of gene expression in arabidopsis. *Proceedings of the National Academy of Sciences*, 115(30) :7831–7836, June 2018. doi : 10.1073/pnas.1804971115. URL https://doi.org/10.1073/pnas.1804971115.

W. R. Lai, M. D. Johnson, R. Kucherlapati, and P. J. Park. Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics*, 21 (19) :3763–3770, Aug. 2005. doi : 10.1093/bioinformatics/bti611. URL https://doi.org/10.1093/bioinformatics/bti611.

I. Lambert, C. P.-L. Roux, S. Colella, and M.-L. Martin-Magniette. DiCoExpress : a tool to process multifactorial RNAseq experiments from quality controls to co-expression analysis through differential analysis based on contrasts inside GLM models. *Plant Methods*, 16(1), May 2020. doi : 10.1186/s13007-020-00611-7. URL https://doi.org/10.1186/s13007-020-00611-7.

S. G. Landt, G. K. Marinov, A. Kundaje, P. Kheradpour, F. Pauli, S. Batzoglou, B. E. Bernstein, P. Bickel, J. B. Brown, P. Cayting, Y. Chen, G. DeSalvo, C. Epstein, K. I. Fisher-Aylor, G. Euskirchen, M. Gerstein, J. Gertz, A. J. Hartemink, M. M. Hoffman, V. R. Iyer, Y. L. Jung, S. Karmakar, M. Kellis, P. V. Kharchenko, Q. Li, T. Liu, X. S. Liu, L. Ma, A. Milosavljevic, R. M. Myers, P. J. Park, M. J. Pazin, M. D. Perry, D. Raha, T. E. Reddy, J. Rozowsky, N. Shoresh, A. Sidow, M. Slattery, J. A. Stamatoyannopoulos, M. Y. Tolstorukov, K. P. White, S. Xi, P. J. Farnham, J. D. Lieb, B. J. Wold, and M. Snyder. ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. *Genome Research*, 22(9) :1813–1831, Sept. 2012. doi : 10.1101/gr.136184.111. URL https://doi.org/10.1101/gr.136184.111.

E. Lebarbier. Detecting multiple change-points in the mean of gaussian process by model selection. *Signal Processing*, 85(4) :717–736, Apr. 2005. doi : 10.1016/j.sigpro.2004.11.012. URL https://doi.org/10.1016/j.sigpro.2004.11.012.

B. Lenhard, A. Sandelin, and P. Carninci. Metazoan promoters : emerging characteristics and insights into transcriptional regulation. *Nature Reviews Genetics*, 13(4) :233–245, Mar. 2012. doi : 10.1038/nrg3163. URL https://doi.org/10.1038/nrg3163.

K. Leppek, R. Das, and M. Barna. Functional 5′ UTR mRNA structures in eukaryotic translation regulation and how to find them. *Nature Reviews Molecular Cell Biology*, 19(3) :158–174, Nov. 2017. doi : 10.1038/nrm.2017.103. URL https://doi.org/10.1038/nrm.2017.103.

J. K. Lerch, F. Kuo, D. Motti, R. Morris, J. L. Bixby, and V. P. Lemmon. Isoform diversity and regulation in peripheral and central neurons revealed through RNA-seq. *PLoS ONE*, 7 (1) :e30417, Jan. 2012. doi : 10.1371/journal.pone.0030417. URL https://doi.org/10.1371/journal.pone.0030417.

M. Leroy, J. Piton, L. Gilet, O. Pellegrini, C. Proux, J.-Y. Coppée, S. Figaro, and C. Condon. Rae1/YacP, a new endoribonuclease involved in ribosome-dependent mRNA decay in bacillus subtilis. *The EMBO Journal*, 36(9) :1167–1181, Mar. 2017. doi : 10.15252/embj.201796540. URL https://doi.org/10.15252/embj.201796540.

S. K. Leung, A. R. Jeffries, I. Castanho, B. T. Jordan, K. Moore, J. P. Davies, E. L. Dempster, N. J. Bray, P. O'Neill, E. Tseng, Z. Ahmed, D. A. Collier, E. D. Jeffery, S. Prabhakar, L. Schalkwyk, C. Jops, M. J. Gandal, G. M. Sheynkman, E. Hannon, and J. Mill. Full-length transcript sequencing of human and mouse cerebral cortex identifies widespread isoform diversity and alternative splicing. *Cell Reports*, 37(7) :110022, Nov. 2021. doi : 10.1016/j.celrep.2021.110022. URL https://doi.org/10.1016/j.celrep.2021.110022.

B. Li and C. N. Dewey. RSEM : accurate transcript quantification from RNA-seq data with or without a reference genome. *BMC Bioinformatics*, 12(1), Aug. 2011. doi : 10.1186/1471-2105-12-323. URL https://doi.org/10.1186/1471-2105-12-323.

X. Li, A. Nair, S. Wang, and L. Wang. Quality control of RNA-seq experiments. In *Methods in Molecular Biology*, pages 137–146. Springer New York, Dec. 2014. doi : 10.1007/978-1-4939-2291-8_8. URL https://doi.org/10.1007/978-1-4939-2291-8_8.

Y. I. Li, D. A. Knowles, J. Humphrey, A. N. Barbeira, S. P. Dickinson, H. K. Im, and J. K. Pritchard. Annotation-free quantification of RNA splicing using LeafCutter. *Nature Genetics*, 50(1) :151–158, Dec. 2017. doi : 10.1038/s41588-017-0004-9. URL https://doi.org/10.1038/s41588-017-0004-9.

Y. Liao, G. K. Smyth, and W. Shi. featureCounts : an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics*, 30(7) :923–930, Nov. 2013. doi : 10.1093/bioinformatics/btt656. URL https://doi.org/10.1093/bioinformatics/btt656.

B. J. Liddicoat, R. Piskol, A. M. Chalk, G. Ramaswami, M. Higuchi, J. C. Hartner, J. B. Li, P. H. Seeburg, and C. R. Walkley. RNA editing by ADAR1 prevents MDA5 sensing of endogenous dsRNA as nonself. *Science*, 349(6252) :1115–1120, Sept. 2015. doi : 10.1126/science.aac7049. URL https://doi.org/10.1126/science.aac7049.

A. Liehrmann and G. Rigaill. Ms.fpop : An exact and fast segmentation algorithm with a multiscale penalty, 2023. URL https://arxiv.org/abs/2303.08723.

A. Liehrmann, G. Rigaill, and T. D. Hocking. Increased peak detection accuracy in over-dispersed ChIP-seq data with supervised segmentation models. *BMC Bioinformatics*, 22(1), June 2021. doi : 10.1186/s12859-021-04221-5. URL https://doi.org/10.1186/s12859-021-04221-5.

A. Liehrmann, E. Delannoy, A. Launay-Avon, E. Gilbault, O. Loudet, B. Castandet, and G. Rigaill. DiffSegR : an RNA-seq data driven method for differential expression analysis using

changepoint detection. *NAR Genomics and Bioinformatics*, 5(4) :lqad098, 11 2023. ISSN 2631-9268. doi : 10.1093/nargab/lqad098. URL https://doi.org/10.1093/nargab/lqad098.

M. S. Lima and D. R. Smith. Pervasive transcription of mitochondrial, plastid, and nucleomorph genomes across diverse plastid-bearing species. *Genome Biology and Evolution*, 9(10) :2650–2657, Sept. 2017. doi : 10.1093/gbe/evx207. URL https://doi.org/10.1093/gbe/evx207.

X. Liu and P. D. Wulf. Probing the ArcA-p modulon of escherichia coli by whole genome transcriptional analysis and sequence recognition profiling. *Journal of Biological Chemistry*, 279(13) :12588–12597, Mar. 2004. doi : 10.1074/jbc.m313454200. URL https://doi.org/10.1074/jbc.m313454200.

M. Lloret-Llinares, S. Pérez-Lluch, D. Rossell, T. Morán, J. Ponsa-Cobas, H. Auer, M. Corominas, and F. Azorín. dKDM5/LID regulates h3k4me3 dynamics at the transcription-start site (TSS) of actively transcribed developmental genes. *Nucleic Acids Research*, 40(19) :9493–9505, Aug. 2012. doi : 10.1093/nar/gks773. URL https://doi.org/10.1093/nar/gks773.

D. J. Lockhart, H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Norton, and E. L. Brown. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, 14(13) :1675–1680, Dec. 1996. doi : 10.1038/nbt1296-1675. URL https://doi.org/10.1038/nbt1296-1675.

I. Lopes, G. Altab, P. Raina, and J. P. de Magalhães. Gene size matters : An analysis of gene length in the human genome. *Frontiers in Genetics*, 12, Feb. 2021. doi : 10.3389/fgene.2021.559998. URL https://doi.org/10.3389/fgene.2021.559998.

J. E. Love, E. J. Hayden, and T. T. Rohn. Alternative splicing in alzheimer's disease. *Journal of Parkinson's disease and Alzheimer's disease*, 2(2), 2015. doi : 10.13188/2376-922x.1000010. URL https://doi.org/10.13188/2376-922x.1000010.

M. I. Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12), Dec. 2014. doi : 10.1186/s13059-014-0550-8. URL https://doi.org/10.1186/s13059-014-0550-8.

C. Lurin, C. Andreés, S. Aubourg, M. Bellaoui, F. Bitton, C. Bruyere, M. Caboche, C. Debast, J. Gualberto, B. Hoffmann, A. Lecharny, M. L. Ret, M.-L. Martin-Magniette, H. Mireau, N. Peeters, J.-P. Renou, B. Szurek, L. Taconnat, and I. Small. Genome-wide analysis of arabidopsis pentatricopeptide repeat proteins reveals their essential role in organelle biogenesis[w]. *The Plant Cell*, 16(8) :2089–2103, Aug. 2004. doi : 10.1105/tpc.104.022236. URL https://doi.org/10.1105/tpc.104.022236.

G. C. MacIntosh and B. Castandet. Organellar and secretory ribonucleases : Major players in plant RNA homeostasis. *Plant Physiology*, 183(4) :1438–1452, June 2020. doi : 10.1104/pp.20.00076. URL https://doi.org/10.1104/pp.20.00076.

R. Maidstone, T. Hocking, G. Rigaill, and P. Fearnhead. On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, 27(2) :519–533, Feb. 2016. doi : 10.1007/s11222-016-9636-3. URL https://doi.org/10.1007/s11222-016-9636-3.

P. A. Makhnovskii, O. A. Gusev, R. O. Bokov, G. R. Gazizova, T. F. Vepkhvadze, E. A. Lysenko, O. L. Vinogradova, F. A. Kolpakov, and D. V. Popov. Alternative transcription start sites contribute to acute-stress-induced transcriptome response in human skeletal muscle. *Human Genomics*, 16(1), July 2022. doi : 10.1186/s40246-022-00399-8. URL https://doi.org/10.1186/s40246-022-00399-8.

N. M. Mannion, S. M. Greenwood, R. Young, S. Cox, J. Brindle, D. Read, C. Nellåker, C. Vesely, C. P. Ponting, P. J. McLaughlin, M. F. Jantsch, J. Dorin, I. R. Adams, A. Scadden, M. Öhman, L. P. Keegan, and M. A. O'Connell. The RNA-editing enzyme ADAR1 controls innate immune responses to RNA. *Cell Reports*, 9(4) :1482–1494, Nov. 2014. doi : 10.1016/j.celrep.2014.10.041. URL https://doi.org/10.1016/j.celrep.2014.10.041.

D. Marques-Coelho, L. da Cruz Carvalho Iohan, A. R. M. de Farias, A. Flaig, F. Letournel, M.-L. Martin-Négrier, F. Chapon, M. Faisant, C. Godfraind, C.-A. Maurage, V. Deramecourt, M. Duchesne, D. Meyronnet, N. Streichenberger, A. M. de Paula, V. Rigau, F. Vandenbos-Burel, C. Duyckaerts, D. Seilhean, S. Milin, D. C. Chiforeanu, A. Laquerrière, F. Marguet, B. Lannes, J.-C. Lambert, and M. R. C. and. Differential transcript usage unravels gene expression alterations in alzheimer's disease human brains. *npj Aging and Mechanisms of Disease*, 7(1), Jan. 2021. doi : 10.1038/s41514-020-00052-5. URL https://doi.org/10.1038/s41514-020-00052-5.

C. Mayr and D. P. Bartel. Widespread shortening of 3′UTRs by alternative cleavage and polyadenylation activates oncogenes in cancer cells. *Cell*, 138(4) :673–684, Aug. 2009. doi : 10.1016/j.cell.2009.06.016. URL https://doi.org/10.1016/j.cell.2009.06.016.

A. Mehmood, A. Laiho, M. S. Venäläinen, A. J. McGlinchey, N. Wang, and L. L. Elo. Systematic evaluation of differential splicing tools for RNA-seq studies. *Briefings in Bioinformatics*, 21 (6) :2052–2065, Dec. 2019. doi : 10.1093/bib/bbz126. URL https://doi.org/10.1093/bib/bbz126.

M. K. Mejía-Guerra, W. Li, N. F. Galeano, M. Vidal, J. Gray, A. I. Doseff, and E. Grotewold. Core promoter plasticity between maize tissues and genotypes contrasts with predominance of sharp transcription initiation sites. *The Plant Cell*, 27(12) :3309–3320, Dec. 2015. doi : 10.1105/tpc.15.00630. URL https://doi.org/10.1105/tpc.15.00630.

A. T. Merryweather-Clarke, A. Atzberger, S. Soneji, N. Gray, K. Clark, C. Waugh, S. J. McGowan, S. Taylor, A. K. Nandi, W. G. Wood, D. J. Roberts, D. R. Higgs, V. J. Buckle, and K. J. H. Robson. Global gene expression analysis of human erythroid proge-

nitors. *Blood*, 117(13) :e96–e108, Mar. 2011. doi : 10.1182/blood-2010-07-290825. URL https://doi.org/10.1182/blood-2010-07-290825.

P. Miura, S. Shenker, C. Andreu-Agullo, J. O. Westholm, and E. C. Lai. Widespread and extensive lengthening of 3′ UTRs in the mammalian brain. *Genome Research*, 23(5) :812–825, Mar. 2013. doi : 10.1101/gr.146886.112. URL https://doi.org/10.1101/gr.146886.112.

A. Morillon and D. Gautheret. Bridging the gap between reference and real transcriptomes. *Genome Biology*, 20(1), June 2019. doi : 10.1186/s13059-019-1710-7. URL https://doi.org/10.1186/s13059-019-1710-7.

H. Moriya. Quantitative nature of overexpression experiments. *Molecular Biology of the Cell*, 26(22) :3932–3939, Nov. 2015. doi : 10.1091/mbc.e15-07-0512. URL https://doi.org/10.1091/mbc.e15-07-0512.

A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by RNA-seq. *Nature Methods*, 5(7) :621–628, May 2008. doi : 10.1038/nmeth.1226. URL https://doi.org/10.1038/nmeth.1226.

M. Movassat, T. L. Crabb, A. Busch, C. Yao, D. J. Reynolds, Y. Shi, and K. J. Hertel. Coupling between alternative polyadenylation and alternative splicing is limited to terminal introns. *RNA Biology*, 13(7) :646–655, July 2016. doi : 10.1080/15476286.2016.1191727. URL https://doi.org/10.1080/15476286.2016.1191727.

V. M. R. Muggeo and G. Adelfio. Efficient change point detection for genomic sequences of continuous measurements. *Bioinformatics*, 27(2) :161–166, Nov. 2010. doi : 10.1093/bioinformatics/btq647. URL https://doi.org/10.1093/bioinformatics/btq647.

A. Nellore, A. E. Jaffe, J.-P. Fortin, J. Alquicira-Hernández, L. Collado-Torres, S. Wang, R. A. P. III, N. Karbhari, K. D. Hansen, B. Langmead, and J. T. Leek. Human splicing diversity and the extent of unannotated splice junctions across human RNA-seq samples on the sequence read archive. *Genome Biology*, 17(1), Dec. 2016. doi : 10.1186/s13059-016-1118-6. URL https://doi.org/10.1186/s13059-016-1118-6.

P. Neuvial, G. Blanchard, G. Durand, N. Enjalbert-Courrech, and E. Roquain. *sanssouci : Post Hoc Multiple Testing Inference*, 2022. URL https://sanssouci-org.github.io/sanssouci. R package version 0.12.8.

J. Neve, R. Patel, Z. Wang, A. Louey, and A. M. Furger. Cleavage and polyadenylation : Ending the message expands gene regulation. *RNA Biology*, 14(7) :865–890, May 2017. doi : 10.1080/15476286.2017.1306171. URL https://doi.org/10.1080/15476286.2017.1306171.

T. W. Nilsen and B. R. Graveley. Expansion of the eukaryotic proteome by alternative splicing. *Nature*, 463(7280) :457–463, Jan. 2010. doi : 10.1038/nature08909. URL https://doi.org/10.1038/nature08909.

K. Nishikura. Editor meets silencer : crosstalk between RNA editing and RNA interference. *Nature Reviews Molecular Cell Biology*, 7(12) :919–931, Dec. 2006. doi : 10.1038/nrm2061. URL https://doi.org/10.1038/nrm2061.

K. Nishikura. Functions and regulation of RNA editing by ADAR deaminases. *Annual Review of Biochemistry*, 79(1) :321–349, June 2010. doi : 10.1146/annurev-biochem-060208-105251. URL https://doi.org/10.1146/annurev-biochem-060208-105251.

K. Nishikura. A-to-i editing of coding and non-coding RNAs by ADARs. *Nature Reviews Molecular Cell Biology*, 17(2) :83–96, Dec. 2015. doi : 10.1038/nrm.2015.4. URL https://doi.org/10.1038/nrm.2015.4.

Y. Noda, S. Okada, and T. Suzuki. Regulation of a-to-i RNA editing and stop codon recoding to control selenoprotein expression during skeletal myogenesis. *Nature Communications*, 13(1), May 2022. doi : 10.1038/s41467-022-30181-2. URL https://doi.org/10.1038/s41467-022-30181-2.

R. Nussinov, C.-J. Tsai, and H. Jang. Protein ensembles link genotype to phenotype. *PLOS Computational Biology*, 15(6) :e1006648, June 2019. doi : 10.1371/journal.pcbi.1006648. URL https://doi.org/10.1371/journal.pcbi.1006648.

K. Okuda, T. Nakamura, M. Sugita, T. Shimizu, and T. Shikanai. A pentatricopeptide repeat protein is a site recognition factor in chloroplast rna editing. *Journal of Biological Chemistry*, 281(49) :37661–37667, 2006.

E. Page. On problems in which a change in a parameter occurs at an unknown point. *Biometrika*, 44(1/2) :248–252, 1957.

E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2) :100–115, 1954.

T. Päll, H. Luidalepp, T. Tenson, and Ülo Maiväli. A field-wide assessment of differential expression profiling by high-throughput sequencing reveals widespread bias. *PLOS Biology*, 21(3) : e3002007, Mar. 2023. doi : 10.1371/journal.pbio.3002007. URL https://doi.org/10.1371/journal.pbio.3002007.

Q. Pan, O. Shai, L. J. Lee, B. J. Frey, and B. J. Blencowe. Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nature Genetics*, 40 (12) :1413–1415, Nov. 2008. doi : 10.1038/ng.259. URL https://doi.org/10.1038/ng.259.

M. Y. Park and T. Hastie. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 69(4) :659–677, 2007.

P. J. Park. ChIP–seq : advantages and challenges of a maturing technology. *Nature Reviews Genetics*, 10(10) :669–680, Sept. 2009. doi : 10.1038/nrg2641. URL https://doi.org/10.1038/nrg2641.

R. Patro, G. Duggal, M. I. Love, R. A. Irizarry, and C. Kingsford. Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods*, 14(4) :417–419, Mar. 2017. doi : 10.1038/nmeth.4197. URL https://doi.org/10.1038/nmeth.4197.

C. M. Perou, T. Sørlie, M. B. Eisen, M. van de Rijn, S. S. Jeffrey, C. A. Rees, J. R. Pollack, D. T. Ross, H. Johnsen, L. A. Akslen, Ø. Fluge, A. Pergamenschikov, C. Williams, S. X. Zhu, P. E. Lønning, A.-L. Børresen-Dale, P. O. Brown, and D. Botstein. Molecular portraits of human breast tumours. *Nature*, 406(6797) :747–752, Aug. 2000. doi : 10.1038/35021093. URL https://doi.org/10.1038/35021093.

M. Pertea, A. Shumate, G. Pertea, A. Varabyou, F. P. Breitwieser, Y.-C. Chang, A. K. Madugundu, A. Pandey, and S. L. Salzberg. CHESS : a new human gene catalog curated from thousands of large-scale RNA sequencing experiments reveals extensive transcriptional noise. *Genome Biology*, 19(1), Nov. 2018. doi : 10.1186/s13059-018-1590-2. URL https://doi.org/10.1186/s13059-018-1590-2.

F. Picard, E. Lebarbier, M. Hoebeke, G. Rigaill, B. Thiam, and S. Robin. Joint segmentation, calling, and normalization of multiple CGH profiles. *Biostatistics*, 12(3) :413–428, Jan. 2011. doi : 10.1093/biostatistics/kxq076. URL https://doi.org/10.1093/biostatistics/kxq076.

H. Pimentel, N. L. Bray, S. Puente, P. Melsted, and L. Pachter. Differential analysis of RNA-seq incorporating quantification uncertainty. *Nature Methods*, 14(7) :687–690, June 2017. doi : 10.1038/nmeth.4324. URL https://doi.org/10.1038/nmeth.4324.

L. Pishchagina, G. Rigaill, and V. Runge. Geometric-based pruning rules for change point detection in multiple independent time series, 2023. URL https://arxiv.org/abs/2306.09555.

M. Pohl, R. H. Bortfeldt, K. Grützmann, and S. Schuster. Alternative splicing of mutually exclusive exons—a review. *Biosystems*, 114(1) :31–38, Oct. 2013. doi : 10.1016/j.biosystems.2013.07.003. URL https://doi.org/10.1016/j.biosystems.2013.07.003.

R. A. Policastro and G. E. Zentner. Global approaches for profiling transcription initiation. *Cell Reports Methods*, 1(5) :100081, Sept. 2021. doi : 10.1016/j.crmeth.2021.100081. URL https://doi.org/10.1016/j.crmeth.2021.100081.

R. I. Ponce-Toledo, P. López-García, and D. Moreira. Horizontal and endosymbiotic gene transfer in early plastid evolution. *New Phytologist*, 224(2) :618–624, July 2019. doi : 10.1111/nph.15965. URL https://doi.org/10.1111/nph.15965.

G. Prelich. Gene overexpression : Uses, mechanisms, and interpretation. *Genetics*, 190(3) : 841–854, Mar. 2012. doi : 10.1534/genetics.111.136911. URL https://doi.org/10.1534/genetics.111.136911.

A. D. Prjibelski, A. Mikheenko, A. Joglekar, A. Smetanin, J. Jarroux, A. L. Lapidus, and H. U. Tilgner. Accurate isoform discovery with IsoQuant using long reads. *Nature Biotechnology*, Jan. 2023. doi : 10.1038/s41587-022-01565-y. URL https://doi.org/10.1038/s41587-022-01565-y.

N. U. Rashid, P. G. Giresi, J. G. Ibrahim, W. Sun, and J. D. Lieb. ZINBA integrates local covariates with DNA-seq data to identify broad and narrow regions of enrichment, even within amplified genomic regions. *Genome Biology*, 12(7) :R67, 2011. doi : 10.1186/gb-2011-12-7-r67. URL https://doi.org/10.1186/gb-2011-12-7-r67.

J. Reeves, J. Chen, X. L. Wang, R. Lund, and Q. Q. Lu. A review and comparison of changepoint detection techniques for climate data. *Journal of Applied Meteorology and Climatology*, 46(6) : 900–915, June 2007. doi : 10.1175/jam2493.1. URL https://doi.org/10.1175/jam2493.1.

H. Rhinn, L. Qiang, T. Yamashita, D. Rhee, A. Zolin, W. Vanti, and A. Abeliovich. Alternative α-synuclein transcript usage as a convergent mechanism in parkinson's disease pathology. *Nature Communications*, 3(1), Sept. 2012. doi : 10.1038/ncomms2032. URL https://doi.org/10.1038/ncomms2032.

G. Rigaill. A pruned dynamic programming algorithm to recover the best segmentations with 1 to k_{max} change-points. *Journal de la société française de statistique*, 156(4) :180–205, 2015. URL http://www.numdam.org/item/JSFS_2015__156_4_180_0/.

G. Rigaill. *fpopw : Weighted Segmentation using Functional Pruning and Optimal Partioning*, 2022. URL https://CRAN.R-project.org/package=fpopw. R package version 1.1.

G. Rigaill, S. Balzergue, V. Brunaud, E. Blondet, A. Rau, O. Rogier, J. Caius, C. Maugis-Rabusseau, L. Soubigou-Taconnat, S. Aubourg, C. Lurin, M.-L. Martin-Magniette, and E. Delannoy. Synthetic data sets for the identification of key ingredients for RNA-seq differential analysis. *Briefings in Bioinformatics*, page bbw092, Oct. 2016. doi : 10.1093/bib/bbw092. URL https://doi.org/10.1093/bib/bbw092.

M. E. Ritchie, B. Phipson, D. Wu, Y. Hu, C. W. Law, W. Shi, and G. K. Smyth. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, 43(7) :e47–e47, Jan. 2015. doi : 10.1093/nar/gkv007. URL https://doi.org/10.1093/nar/gkv007.

M. D. Robinson, D. J. McCarthy, and G. K. Smyth. edger : a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1) :139–

140, Nov. 2009. doi : 10.1093/bioinformatics/btp616. URL https://doi.org/10.1093/bioinformatics/btp616.

G. Romano, G. Rigaill, V. Runge, and P. Fearnhead. Detecting abrupt changes in the presence of local fluctuations and autocorrelated noise. *Journal of the American Statistical Association*, 117(540) :2147–2162, May 2021. doi : 10.1080/01621459.2021.1909598. URL https://doi.org/10.1080/01621459.2021.1909598.

G. Romano, I. A. Eckley, P. Fearnhead, and G. Rigaill. Fast online changepoint detection via functional pruning cusum statistics. *Journal of Machine Learning Research*, 24(81) :1–36, 2023. URL http://jmlr.org/papers/v24/21-1230.html.

J. J. Rosenthal and P. H. Seeburg. A-to-i RNA editing : Effects on proteins key to neural excitability. *Neuron*, 74(3) :432–439, May 2012. doi : 10.1016/j.neuron.2012.04.010. URL https://doi.org/10.1016/j.neuron.2012.04.010.

J. Rozowsky, G. Euskirchen, R. K. Auerbach, Z. D. Zhang, T. Gibson, R. Bjornson, N. Carriero, M. Snyder, and M. B. Gerstein. PeakSeq enables systematic scoring of ChIP-seq experiments relative to controls. *Nature Biotechnology*, 27(1) :66–75, Jan. 2009. doi : 10.1038/nbt.1518. URL https://doi.org/10.1038/nbt.1518.

V. Runge, T. D. Hocking, G. Romano, F. Afghah, P. Fearnhead, and G. Rigaill. gfpop : An r package for univariate graph-constrained change-point detection. *Journal of Statistical Software*, 106(6), 2023. doi : 10.18637/jss.v106.i06. URL https://doi.org/10.18637/jss.v106.i06.

H. Ruwe, B. Castandet, C. Schmitz-Linneweber, and D. B. Stern. Arabidopsis chloroplast quantitative editotype. *FEBS Letters*, 587(9) :1429–1433, Mar. 2013. doi : 10.1016/j.febslet.2013.03.022. URL https://doi.org/10.1016/j.febslet.2013.03.022.

R. Sandberg, J. R. Neilson, A. Sarma, P. A. Sharp, and C. B. Burge. Proliferating cells express mRNAs with shortened 3' untranslated regions and fewer MicroRNA target sites. *Science*, 320(5883) :1643–1647, June 2008. doi : 10.1126/science.1155390. URL https://doi.org/10.1126/science.1155390.

A. Sandelin, P. Carninci, B. Lenhard, J. Ponjavic, Y. Hayashizaki, and D. A. Hume. Mammalian RNA polymerase II core promoters : insights from genome-wide studies. *Nature Reviews Genetics*, 8(6) :424–436, May 2007. doi : 10.1038/nrg2026. URL https://doi.org/10.1038/nrg2026.

D. Sarantopoulou, T. G. Brooks, S. Nayak, A. Mrčela, N. F. Lahens, and G. R. Grant. Comparative evaluation of full-length isoform quantification from RNA-seq. *BMC Bioinformatics*, 22(1), May 2021. doi : 10.1186/s12859-021-04198-1. URL https://doi.org/10.1186/s12859-021-04198-1.

M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235) :467–470, Oct. 1995. doi : 10.1126/science.270.5235.467. URL https://doi.org/10.1126/science.270.5235.467.

C. Schmitz-Linneweber. Heterologous, splicing-dependent RNA editing in chloroplasts : allotetraploidy provides trans-factors. *The EMBO Journal*, 20(17) :4874–4883, Sept. 2001. doi : 10.1093/emboj/20.17.4874. URL https://doi.org/10.1093/emboj/20.17.4874.

M. M. Scotti and M. S. Swanson. RNA mis-splicing in disease. *Nature Reviews Genetics*, 17(1) : 19–32, Nov. 2015. doi : 10.1038/nrg.2015.3. URL https://doi.org/10.1038/nrg.2015.3.

C. Sessegolo, C. Cruaud, C. D. Silva, A. Cologne, M. Dubarry, T. Derrien, V. Lacroix, and J.-M. Aury. Transcriptome profiling of mouse samples using nanopore sequencing of cDNA and RNA molecules. *Scientific Reports*, 9(1), Oct. 2019. doi : 10.1038/s41598-019-51470-9. URL https://doi.org/10.1038/s41598-019-51470-9.

S. Shen, J. W. Park, Z. xiang Lu, L. Lin, M. D. Henry, Y. N. Wu, Q. Zhou, and Y. Xing. rMATS : Robust and flexible detection of differential alternative splicing from replicate RNA-seq data. *Proceedings of the National Academy of Sciences*, 111(51), Dec. 2014. doi : 10.1073/pnas.1419161111. URL https://doi.org/10.1073/pnas.1419161111.

P. J. Shepard, E.-A. Choi, J. Lu, L. A. Flanagan, K. J. Hertel, and Y. Shi. Complex and dynamic landscape of RNA polyadenylation revealed by PAS-seq. *RNA*, 17(4) :761–772, Feb. 2011. doi : 10.1261/rna.2581711. URL https://doi.org/10.1261/rna.2581711.

H. Shi, Y. Zhou, E. Jia, M. Pan, Y. Bai, and Q. Ge. Bias in RNA-seq library preparation : Current challenges and solutions. *BioMed Research International*, 2021 :1–11, Apr. 2021. doi : 10.1155/2021/6647597. URL https://doi.org/10.1155/2021/6647597.

D. Singh, C. K. Singh, J. Taunk, R. S. S. Tomar, A. K. Chaturvedi, K. Gaikwad, and M. Pal. Transcriptome analysis of lentil (lens culinaris medikus) in response to seedling drought stress. *BMC Genomics*, 18(1), Feb. 2017. doi : 10.1186/s12864-017-3596-7. URL https://doi.org/10.1186/s12864-017-3596-7.

I. Small, J. Melonek, A.-V. Bohne, J. Nickelsen, and C. Schmitz-Linneweber. Plant organellar RNA maturation. *The Plant Cell*, 35(6) :1727–1751, Feb. 2023. doi : 10.1093/plcell/koad049. URL https://doi.org/10.1093/plcell/koad049.

D. R. Smith. Haematococcus lacustris : the makings of a giant-sized chloroplast genome. *AoB Plants*, 10(5) :ply058, 2018.

C. Soneson, K. L. Matthes, M. Nowicka, C. W. Law, and M. D. Robinson. Isoform prefiltering improves performance of count-based methods for analysis of differential transcript usage.

Genome Biology, 17(1), Jan. 2016. doi : 10.1186/s13059-015-0862-3. URL https://doi.org/10.1186/s13059-015-0862-3.

C. Soneson, Y. Yao, A. Bratus-Neuenschwander, A. Patrignani, M. D. Robinson, and S. Hussain. A comprehensive examination of nanopore native RNA sequencing for characterization of complex transcriptomes. *Nature Communications*, 10(1), July 2019. doi : 10.1038/s41467-019-11272-z. URL https://doi.org/10.1038/s41467-019-11272-z.

C. Spyrou, R. Stark, A. G. Lynch, and S. Tavaré. BayesPeak : Bayesian analysis of ChIP-seq data. *BMC Bioinformatics*, 10(1), Sept. 2009. doi : 10.1186/1471-2105-10-299. URL https://doi.org/10.1186/1471-2105-10-299.

A. Srivastava, L. Malik, H. Sarkar, M. Zakeri, F. Almodaresi, C. Soneson, M. I. Love, C. Kingsford, and R. Patro. Alignment and mapping methodology influence transcript abundance estimation. *Genome Biology*, 21(1), Sept. 2020. doi : 10.1186/s13059-020-02151-8. URL https://doi.org/10.1186/s13059-020-02151-8.

R. Stark, M. Grzelak, and J. Hadfield. RNA sequencing : the teenage years. *Nature Reviews Genetics*, 20(11) :631–656, July 2019. doi : 10.1038/s41576-019-0150-2. URL https://doi.org/10.1038/s41576-019-0150-2.

T. Steijger, , J. F. Abril, P. G. Engström, F. Kokocinski, T. J. Hubbard, R. Guigó, J. Harrow, and P. Bertone. Assessment of transcript reconstruction methods for RNA-seq. *Nature Methods*, 10(12) :1177–1184, Nov. 2013. doi : 10.1038/nmeth.2714. URL https://doi.org/10.1038/nmeth.2714.

D. B. Stern and W. Gruissem. Control of plastid gene expression : 3′ inverted repeats act as mRNA processing and stabilizing elements, but do not terminate transcription. *Cell*, 51(6) : 1145–1157, Dec. 1987. doi : 10.1016/0092-8674(87)90600-3. URL https://doi.org/10.1016/0092-8674(87)90600-3.

D. B. Stern, M. Goldschmidt-Clermont, and M. R. Hanson. Chloroplast RNA metabolism. *Annual Review of Plant Biology*, 61(1) :125–155, June 2010. doi : 10.1146/annurev-arplant-042809-112242. URL https://doi.org/10.1146/annurev-arplant-042809-112242.

W. Sun, X. You, A. Gogol-Döring, H. He, Y. Kise, M. Sohn, T. Chen, A. Klebes, D. Schmucker, and W. Chen. Ultra-deep profiling of alternatively spliced drosophila dscam isoforms by circularization-assisted multi-segment sequencing. *The EMBO Journal*, 32(14) :2029–2038, June 2013. doi : 10.1038/emboj.2013.144. URL https://doi.org/10.1038/emboj.2013.144.

A. D. Tang, C. M. Soulette, M. J. van Baren, K. Hart, E. Hrabeta-Robinson, C. J. Wu, and A. N. Brooks. Full-length transcript characterization of SF3b1 mutation in chronic lymphocytic leu-

kemia reveals downregulation of retained introns. *Nature Communications*, 11(1), Mar. 2020. doi : 10.1038/s41467-020-15171-6. URL https://doi.org/10.1038/s41467-020-15171-6.

S. Tarazona, F. García, A. Ferrer, J. Dopazo, and A. Conesa. NOIseq : a RNA-seq differential expression method robust for sequencing depth biases. *EMBnet.journal*, 17(B) :18, Feb. 2012. doi : 10.14806/ej.17.b.265. URL https://doi.org/10.14806/ej.17.b.265.

A. G. Tartakovsky. Rapid detection of attacks in computer networks by quickest changepoint detection methods. In *Data Analysis for Network Cyber-Security*, pages 33–70. IMPERIAL COLLEGE PRESS, Apr. 2014. doi : 10.1142/9781783263752_0002. URL https://doi.org/10.1142/9781783263752_0002.

M. K. Tello-Ruiz, J. Stein, S. Wei, J. Preece, A. Olson, S. Naithani, V. Amarasinghe, P. Dharmawardhana, Y. Jiao, J. Mulvaney, S. Kumari, K. Chougule, J. Elser, B. Wang, J. Thomason, D. M. Bolser, A. Kerhornou, B. Walts, N. A. Fonseca, L. Huerta, M. Keays, Y. A. Tang, H. Parkinson, A. Fabregat, S. McKay, J. Weiser, P. D'Eustachio, L. Stein, R. Petryszak, P. J. Kersey, P. Jaiswal, and D. Ware. Gramene 2016 : comparative plant genomics and pathway resources. *Nucleic Acids Research*, 44(D1) :D1133–D1140, Nov. 2015. doi : 10.1093/nar/gkv1179. URL https://doi.org/10.1093/nar/gkv1179.

H. Thorvaldsdottir, J. T. Robinson, and J. P. Mesirov. Integrative genomics viewer (IGV) : high-performance genomics data visualization and exploration. *Briefings in Bioinformatics*, 14(2) : 178–192, Apr. 2012. doi : 10.1093/bib/bbs017. URL https://doi.org/10.1093/bib/bbs017.

B. Tian and J. L. Manley. Alternative polyadenylation of mRNA precursors. *Nature Reviews Molecular Cell Biology*, 18(1) :18–30, Sept. 2016. doi : 10.1038/nrm.2016.116. URL https://doi.org/10.1038/nrm.2016.116.

J. N. Timmis, M. A. Ayliffe, C. Y. Huang, and W. Martin. Endosymbiotic gene transfer : organelle genomes forge eukaryotic chromosomes. *Nature Reviews Genetics*, 5(2) :123–135, Feb. 2004. doi : 10.1038/nrg1271. URL https://doi.org/10.1038/nrg1271.

H. C. Tran, V. Schmitt, S. Lama, C. Wang, A. Launay-Avon, K. Bernfur, K. Sultan, K. Khan, V. Brunaud, A. Liehrmann, et al. An mtran-mrna interaction mediates mitochondrial translation initiation in plants. *Science*, 381(6661) :eadg0995, 2023.

V. D. T. Tran, O. Souiai, N. Romero-Barrios, M. Crespi, and D. Gautheret. Detection of generic differential RNA processing events from RNA-seq data. *RNA Biology*, 13(1) :59–67, Jan. 2016. doi : 10.1080/15476286.2015.1118604. URL https://doi.org/10.1080/15476286.2015.1118604.

C. Trapnell, D. G. Hendrickson, M. Sauvageau, L. Goff, J. L. Rinn, and L. Pachter. Differential analysis of gene regulation at transcript resolution with RNA-seq. *Nature Biotechnology*, 31 (1) :46–53, Dec. 2012. doi : 10.1038/nbt.2450. URL https://doi.org/10.1038/nbt.2450.

G. Tushev, C. Glock, M. Heumüller, A. Biever, M. Jovanovic, and E. M. Schuman. Alternative 3′ UTRs modify the localization, regulatory potential, stability, and plasticity of mRNAs in neuronal compartments. *Neuron*, 98(3) :495–511.e6, May 2018. doi : 10.1016/j.neuron.2018.03.030. URL https://doi.org/10.1016/j.neuron.2018.03.030.

A. Typas, G. Becker, and R. Hengge. The molecular basis of selective promoter activation by the subunit of RNA polymerase. *Molecular Microbiology*, 63(5) :1296–1306, Mar. 2007. doi : 10.1111/j.1365-2958.2007.05601.x. URL https://doi.org/10.1111/j.1365-2958.2007.05601.x.

T. Ushijima, K. Hanada, E. Gotoh, W. Yamori, Y. Kodama, H. Tanaka, M. Kusano, A. Fukushima, M. Tokizawa, Y. Y. Yamamoto, Y. Tada, Y. Suzuki, and T. Matsushita. Light controls protein localization through phytochrome-mediated alternative promoter selection. *Cell*, 171(6) :1316–1325.e12, Nov. 2017. doi : 10.1016/j.cell.2017.10.018. URL https://doi.org/10.1016/j.cell.2017.10.018.

N. Verzelen, M. Fromont, M. Lerasle, and P. Reynaud-Bouret. Optimal change-point detection and localization. *arXiv preprint arXiv :2010.11470*, 2020.

E. T. Wang, R. Sandberg, S. Luo, I. Khrebtukova, L. Zhang, C. Mayr, S. F. Kingsmore, G. P. Schroth, and C. B. Burge. Alternative isoform regulation in human tissue transcriptomes. *Nature*, 456(7221) :470–476, Nov. 2008. doi : 10.1038/nature07509. URL https://doi.org/10.1038/nature07509.

Y. Wang, J. Liu, B. huang, Y. mai Xu, J. Li, L.-F. Huang, J. Lin, J. Zhang, Q.-H. Min, W.-M. Yang, and X.-Z. Wang. Mechanism of alternative splicing and its regulation. *Biomedical Reports*, 3(2) :152–158, Dec. 2014. doi : 10.3892/br.2014.407. URL https://doi.org/10.3892/br.2014.407.

Y. Wang, Y. Zhao, A. Bollas, Y. Wang, and K. F. Au. Nanopore sequencing technology, bioinformatics and applications. *Nature Biotechnology*, 39(11) :1348–1365, Nov. 2021. doi : 10.1038/s41587-021-01108-x. URL https://doi.org/10.1038/s41587-021-01108-x.

Z. Wang, M. Gerstein, and M. Snyder. RNA-seq : a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1) :57–63, Jan. 2009. doi : 10.1038/nrg2484. URL https://doi.org/10.1038/nrg2484.

N. Whiffin, K. J. Karczewski, X. Zhang, S. Chothani, M. J. Smith, D. G. Evans, A. M. Roberts, N. M. Quaife, S. Schafer, O. Rackham, J. Alföldi, A. H. O'Donnell-Luria, L. C. Francioli, I. M. Armean, E. Banks, L. Bergelson, K. Cibulskis, R. L. Collins, K. M. Connolly, M. Covarrubias, B. Cummings, M. J. Daly, S. Donnelly, Y. Farjoun, S. Ferriera, S. Gabriel, L. D. Gauthier, J. Gentry, N. Gupta, T. Jeandet, D. Kaplan, K. M. Laricchia, C. Llanwarne, E. V.

Minikel, R. Munshi, B. M. Neale, S. Novod, N. Petrillo, T. Poterba, D. Roazen, V. Ruano-Rubio, A. Saltzman, K. E. Samocha, M. Schleicher, C. Seed, M. Solomonson, J. Soto, G. Tiao, K. Tibbetts, C. Tolonen, C. Vittal, G. Wade, A. Wang, Q. Wang, N. A. Watts, B. Weisburd, C. A. A. Salinas, T. Ahmad, C. M. Albert, D. Ardissino, G. Atzmon, J. Barnard, L. Beaugerie, E. J. Benjamin, M. Boehnke, L. L. Bonnycastle, E. P. Bottinger, D. W. Bowden, M. J. Bown, J. C. Chambers, J. C. Chan, D. Chasman, J. Cho, M. K. Chung, B. Cohen, A. Correa, D. Dabelea, M. J. Daly, D. Darbar, R. Duggirala, J. Dupuis, P. T. Ellinor, R. Elosua, J. Erdmann, T. Esko, M. Färkkilä, J. Florez, A. Franke, G. Getz, B. Glaser, S. J. Glatt, D. Goldstein, C. Gonzalez, L. Groop, C. Haiman, C. Hanis, M. Harms, M. Hiltunen, M. M. Holi, C. M. Hultman, M. Kallela, J. Kaprio, S. Kathiresan, B.-J. Kim, Y. J. Kim, G. Kirov, J. Kooner, S. Koskinen, H. M. Krumholz, S. Kugathasan, S. H. Kwak, M. Laakso, T. Lehtimäki, R. J. F. Loos, S. A. Lubitz, R. C. W. Ma, J. Marrugat, K. M. Mattila, S. McCarroll, M. I. McCarthy, D. McGovern, R. McPherson, J. B. Meigs, O. Melander, A. Metspalu, B. M. Neale, P. M. Nilsson, M. C. O'Donovan, D. Ongur, L. Orozco, M. J. Owen, C. N. A. Palmer, A. Palotie, K. S. Park, C. Pato, A. E. Pulver, N. Rahman, A. M. Remes, J. D. Rioux, S. Ripatti, D. M. Roden, D. Saleheen, V. Salomaa, N. J. Samani, J. Scharf, H. Schunkert, M. B. Shoemaker, P. Sklar, H. Soininen, H. Sokol, T. Spector, P. F. Sullivan, J. Suvisaari, E. S. Tai, Y. Y. Teo, T. Tiinamaija, M. Tsuang, D. Turner, T. Tusie-Luna, E. Vartiainen, H. Watkins, R. K. Weersma, M. Wessman, J. G. Wilson, R. J. Xavier, M. P. Vawter, S. A. Cook, P. J. R. Barton, D. G. MacArthur, J. S. Ware, and and. Characterising the loss-of-function impact of 5' untranslated region variants in 15, 708 individuals. *Nature Communications*, 11(1), May 2020. doi : 10.1038/s41467-019-10717-9. URL https://doi.org/10.1038/s41467-019-10717-9.

T. Widiez, A. Symeonidi, C. Luo, E. Lam, M. Lawton, and S. A. Rensing. The chromatin landscape of the moss physcomitrella patens and its dynamics during development and drought stress. *The Plant Journal*, 79(1) :67–81, June 2014. doi : 10.1111/tpj.12542. URL https://doi.org/10.1111/tpj.12542.

E. M. Wissink, A. Vihervaara, N. D. Tippens, and J. T. Lis. Nascent RNA analyses : tracking transcription and its regulation. *Nature Reviews Genetics*, 20(12) :705–723, Aug. 2019. doi : 10.1038/s41576-019-0159-6. URL https://doi.org/10.1038/s41576-019-0159-6.

D. Xin, L. Hu, and X. Kong. Alternative promoters influence alternative splicing at the genomic level. *PLoS ONE*, 3(6) :e2377, June 2008. doi : 10.1371/journal.pone.0002377. URL https://doi.org/10.1371/journal.pone.0002377.

H. Xing, Y. Mo, W. Liao, and M. Q. Zhang. Genome-wide localization of protein-DNA binding and histone modification by a bayesian change-point method with ChIP-seq data. *PLoS Computational Biology*, 8(7) :e1002613, July 2012. doi : 10.1371/journal.pcbi.1002613. URL https://doi.org/10.1371/journal.pcbi.1002613.

H. Xu, L. Handoko, X. Wei, C. Ye, J. Sheng, C.-L. Wei, F. Lin, and W.-K. Sung. A signal–noise model for significance analysis of ChIP-seq with negative control. *Bioinformatics*, 26(9) : 1199–1204, Apr. 2010. doi : 10.1093/bioinformatics/btq128. URL https://doi.org/10.1093/bioinformatics/btq128.

H. K. Yalamanchili, C. E. Alcott, P. Ji, E. J. Wagner, H. Y. Zoghbi, and Z. Liu. PolyA-miner : accurate assessment of differential alternative poly-adenylation from 3′seq data using vector projections and non-negative matrix factorization. *Nucleic Acids Research*, 48(12) :e69–e69, May 2020. doi : 10.1093/nar/gkaa398. URL https://doi.org/10.1093/nar/gkaa398.

J. Yang, D. Liu, X. Wang, C. Ji, F. Cheng, B. Liu, Z. Hu, S. Chen, D. Pental, Y. Ju, P. Yao, X. Li, K. Xie, J. Zhang, J. Wang, F. Liu, W. Ma, J. Shopan, H. Zheng, S. A. Mackenzie, and M. Zhang. The genome sequence of allopolyploid brassica juncea and analysis of differential homoeolog gene expression influencing selection. *Nature Genetics*, 48(10) :1225–1232, Sept. 2016. doi : 10.1038/ng.3657. URL https://doi.org/10.1038/ng.3657.

Y.-C. Yao and S.-T. Au. Least-squares estimation of a step function. *Sankhyā : The Indian Journal of Statistics, Series A*, pages 370–381, 1989.

A. Yap, P. Kindgren, C. C. des Francs-Small, T. Kazama, S. K. Tanz, K. Toriyama, and I. Small. AEF1/MPR25 is implicated in RNA editing of plastid atpf and mitochondrial nad5, and also promotes atpf splicing in arabidopsis and rice. *The Plant Journal*, 81(5) :661–669, Feb. 2015. doi : 10.1111/tpj.12756. URL https://doi.org/10.1111/tpj.12756.

K. Yap and E. V. Makeyev. Functional impact of splice isoform diversity in individual cells. *Biochemical Society Transactions*, 44(4) :1079–1085, Aug. 2016. doi : 10.1042/bst20160103. URL https://doi.org/10.1042/bst20160103.

B.-H. You, S.-H. Yoon, and J.-W. Nam. High-confidence coding and noncoding transcriptome maps. *Genome Research*, 27(6) :1050–1062, Apr. 2017. doi : 10.1101/gr.214288.116. URL https://doi.org/10.1101/gr.214288.116.

C. Zang, D. E. Schones, C. Zeng, K. Cui, K. Zhao, and W. Peng. A clustering approach for identification of enriched domains from histone modification ChIP-seq data. *Bioinformatics*, 25(15) :1952–1958, June 2009. doi : 10.1093/bioinformatics/btp340. URL https://doi.org/10.1093/bioinformatics/btp340.

C. Zhang, B. Zhang, L.-L. Lin, and S. Zhao. Evaluation and comparison of computational tools for RNA-seq isoform quantification. *BMC Genomics*, 18(1), Aug. 2017a. doi : 10.1186/s12864-017-4002-1. URL https://doi.org/10.1186/s12864-017-4002-1.

N. R. Zhang and D. O. Siegmund. A modified bayes information criterion with applications to the analysis of comparative genomic hybridization data. *Biometrics*, 63(1) :22–32,

Oct. 2006b. doi : 10.1111/j.1541-0420.2006.00662.x. URL https://doi.org/10.1111/j.1541-0420.2006.00662.x.

P. Zhang, J. Du, B. Sun, X. Dong, G. Xu, J. Zhou, Q. Huang, Q. Liu, Q. Hao, and J. Ding. Structure of human MRG15 chromo domain and its binding to lys36-methylated histone h3. *Nucleic Acids Research*, 34(22) :6621–6628, Nov. 2006. doi : 10.1093/nar/gkl989. URL https://doi.org/10.1093/nar/gkl989.

P. Zhang, , E. Dimont, T. Ha, D. J. Swanson, W. Hide, and D. Goldowitz. Relatively frequent switching of transcription start sites during cerebellar development. *BMC Genomics*, 18(1), June 2017b. doi : 10.1186/s12864-017-3834-z. URL https://doi.org/10.1186/s12864-017-3834-z.

Y. Zhang, T. Liu, C. A. Meyer, J. Eeckhoute, D. S. Johnson, B. E. Bernstein, C. Nusbaum, R. M. Myers, M. Brown, W. Li, and X. S. Liu. Model-based analysis of ChIP-seq (MACS). *Genome Biology*, 9(9), Sept. 2008. doi : 10.1186/gb-2008-9-9-r137. URL https://doi.org/10.1186/gb-2008-9-9-r137.

Y. Zhang, L. Tian, and C. Lu. Chloroplast gene expression : Recent advances and perspectives. *Plant Communications*, page 100611, May 2023. doi : 10.1016/j.xplc.2023.100611. URL https://doi.org/10.1016/j.xplc.2023.100611.

M. A. Zipeto, Q. Jiang, E. Melese, and C. H. Jamieson. RNA rewriting, recoding, and rewiring in human disease. *Trends in Molecular Medicine*, 21(9) :549–559, Sept. 2015. doi : 10.1016/j.molmed.2015.07.001. URL https://doi.org/10.1016/j.molmed.2015.07.001.

M. Zytnicki and I. González. Finding differentially expressed sRNA-seq regions with srnadiff. *PLOS ONE*, 16(8) :e0256196, Aug. 2021. doi : 10.1371/journal.pone.0256196. URL https://doi.org/10.1371/journal.pone.0256196.

Chapter A

Increased peak detection accuracy in over-dispersed ChIP-Seq data with supervised segmentation models

This article was published in the journal *BMC Bioinformatics* (doi.org/10.1186/s12859-021-04221-5).

Liehrmann *et al. BMC Bioinformatics* (2021) 22:323
https://doi.org/10.1186/s12859-021-04221-5

BMC Bioinformatics

RESEARCH Open Access

Increased peak detection accuracy in over-dispersed ChIP-seq data with supervised segmentation models

Arnaud Liehrmann[1,2*], Guillem Rigaill[1,2] and Toby Dylan Hocking[3]

*Correspondence: arnaud.liehrmann@universite-paris-saclay.fr
[2] Laboratoire de Mathématiques et Modélisation d'Evry (LAMME), Université Paris-Saclay, Université Evry, CNRS, 91037 Evry, France
Full list of author information is available at the end of the article

Abstract

Background: Histone modification constitutes a basic mechanism for the genetic regulation of gene expression. In early 2000s, a powerful technique has emerged that couples chromatin immunoprecipitation with high-throughput sequencing (ChIP-seq). This technique provides a direct survey of the DNA regions associated to these modifications. In order to realize the full potential of this technique, increasingly sophisticated statistical algorithms have been developed or adapted to analyze the massive amount of data it generates. Many of these algorithms were built around natural assumptions such as the Poisson distribution to model the noise in the count data. In this work we start from these natural assumptions and show that it is possible to improve upon them.

Results: Our comparisons on seven reference datasets of histone modifications (H3K36me3 & H3K4me3) suggest that natural assumptions are not always realistic under application conditions. We show that the unconstrained multiple changepoint detection model with alternative noise assumptions and supervised learning of the penalty parameter reduces the over-dispersion exhibited by count data. These models, implemented in the R package *CROCS* (https://github.com/aLiehrmann/CROCS), detect the peaks more accurately than algorithms which rely on natural assumptions.

Conclusion: The segmentation models we propose can benefit researchers in the field of epigenetics by providing new high-quality peak prediction tracks for H3K36me3 and H3K4me3 histone modifications.

Keywords: ChIP-seq, Histone modifications, Over-dispersion, Peak calling, Multiple changepoint detection, Likelihood inference, Supervised learning

Background

Chromatin immunoprecipitation followed by high-throughput sequencing (ChIP-seq) is amongst the most widely used methods in molecular biology [15]. This method aims to identify transcription factor binding sites [20, 22] or post-translational histone modifications [24, 25], referred to as histone marks, underlying regulatory elements. Consequently, this method is essential to deepen our understanding of transcriptional regulation. The ChIP-seq assay yields a set of DNA sequence reads which are aligned to

a reference genome and then counted at each genomic position. This results in a series $Y = (y_1, \ldots, y_n)$ of n non-negative integer count data ($y_i \in \mathbb{Z}_+$), hereafter called coverage profile, ordered along a chromosome. The binding sites or histone marks of interest appear as regions with high read density referred to as peaks in the coverage profile.

Since there is a biological interest in detecting these peaks, several methods, hereafter called peak callers (c), have been developed / adapted and used to filter out background noise and accurately identify the peak locations in the coverage profile. They take a coverage profile of length n and classify each base from it as a part of the background noise (0) or peak (1), i.e. $c : Y \to \{0, 1\}^n$. Among these peak callers we can mention MACS [26] and HMCan [2], two heuristics which are computationally fast but typically accurate only for a specific pattern, i.e. respectively sharp and broad peaks [7]. More recently, it has been proposed to solve the peak detection problem using either optimal constrained or unconstrained multiple changepoint detection methods [8, 12]. The constraints ensure that the segmentation model can be interpreted in terms of peaks and background noise which is a practitioner's request. The unconstrained one doesn't have an output segmentation with a straightforward interpretation in terms of peaks and needs to be followed by an ad-hoc post-processing rule to infer the start and end of peaks (see Fig. 2). For each of these methods, there are one or more tuning parameters that need to be set before solving the peak detection problem and that may affect the results accuracy.

In a supervised learning approach, Hocking et al. [7] introduced seven labeled histone mark datasets that are composed of samples from two different ChIP-seq experiments directed at histone modifications H3K36me3 and H3K36me3. In a recent study, after training different peak callers using these datasets, Hocking et al. [12] compared them and showed that the constrained segmentation model with count data following a Poisson distribution outperforms standard bioinformatics heuristics and the unconstrained segmentation model on these datasets.

Modeling question

From a modeling perspective the constrained segmentation model and the Poisson noise are certainly the most natural assumptions to detect peaks in coverage profiles. However, it is not clear that they are realistic:

- By looking at the shapes of the peaks in coverage profiles (see for instance in Fig. 1), we can see that the background noise and the top of the peaks are sometimes separated by one or more subtle changes. In contrast to the constrained segmentation model, the unconstrained one should be able to capture these subtle changes. One major issue is that the output segmentation of the unconstrained model does not have a straightforward interpretation in terms of peaks.
- Parametric models such as the negative binomial [14, 17] or the Gaussian, following a proper transformation of the count data for the latter [1, 13], are preferred over the Poisson one for the analysis of many high-througput sequencing datasets. Indeed, count data often exhibit more variability than the Poisson model expects which changes the interpretation of the model and makes it difficult to estimate its param-

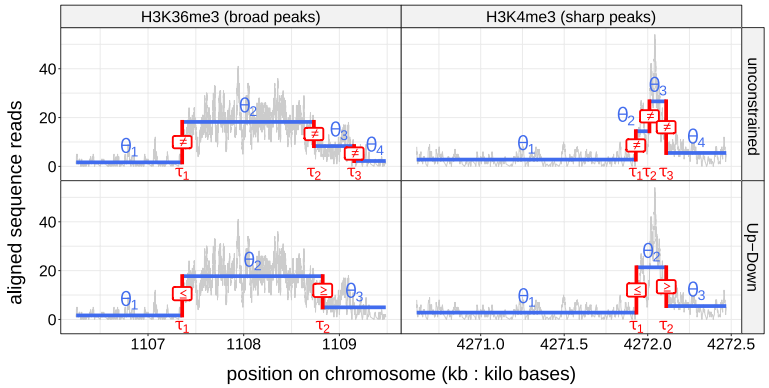


Fig. 1 Examples of ChIP-seq coverage profiles from the histone mark H3K36me3 and H3K4me3 datasets. (**Top**) In blue, a piecewise constant function affected by three unconstrained abrupt changes shown in red. (**Bottom**) In blue, a piecewise constant function affected by two constrained abrupt changes shown in red

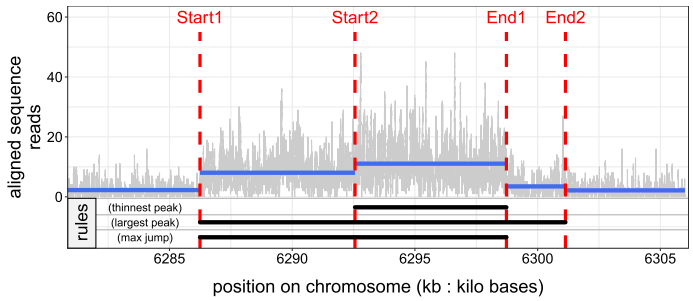


Fig. 2 (**Top**) Segmentation of a coverage profile containing one peak using the unconstrained model. The location of the changepoints on the chromosome are shown by red dotted lines. The mean of the segments are shown in blue. According to this segmentation there are two alternative starts and two alternative ends of the peak, i.e. four alternative variants of the same peak formed by the regions: [Start1:End1], [Start1:End2], [Start2:End1] and [Start2:End2]. (**Bottom**) Three different rules are proposed to interpret the segmentation as peaks. *Thinnest peak*: the resulting peak is defined by the region [Start2:End1]. *Largest peak*: the resulting peak is defined by the region [Start1:End2]. *Max jump*: the resulting peak is defined by the region [Start1:End1]

eters. These alternative parametric models are well known to reduce this phenomenon, also called over-dispersion.

In this work we try to start from these natural assumptions and show that it is possible to improve upon them.

Contribution

1. We show that the distribution of counts from H3K36me3 and H3K4me3 datasets exhibits over-dispersion which invalidates the Poisson assumption. The two alternative noise models we propose (negative binomial with constant dispersion parameter & Gaussian after Anscombe transformation) effectively reduce the over-dispersion on these datasets (see Fig. 4).

2. We propose a new and rather natural post-processing rule to predict the start and end of peaks in an estimated unconstrained segmentation (see Fig. 2). Indeed, in the unconstrained segmentation we can observe several up (respectively down) changes and it is not obvious which one should be considered as the start or end of the peak. We show that this new post-processing rule improves the accuracy of the unconstrained segmentation model in both H3K36me3 and H3K4me3 datasets compared to the same model with previous rules described by Hocking et al. [12] (see Fig. 5).
3. Hocking et al. [11] described a procedure to extract all optimal constrained segmentations for a range of peaks. It is an essential internal step in the supervised approach for learning the penalty parameter of segmentation models. In this work we generalize this procedure so that it works with the unconstrained segmentation model and the post-processing rule mentioned in the previous point (see Algorithm 1).
4. We describe a method to learn jointly both the penalty and dispersion parameters of segmentation models with a negative binomial noise. We then compare the accuracy of unconstrained and contrained segmention models with different noise distributions on the labeled H3K36me3 and H3K4me3 datasets (see Fig. 6).

Methods

Segmentation models for ChIP-seq data

Unconstrained segmentation model

The observed data $(y_1, \ldots, y_n)$ are supposed to be a realization of an independent random process $(Y_1, \ldots, Y_n)$. This process is drawn from a probability distribution $\mathcal{F}$ which depends on two parameters: θ is assumed to be affected by $K-1$ abrupt changes called changepoints and ϕ is constant. We denote τ_k the location of the kth changepoint with $k = \{1, \ldots, K-1\}$. By convention we introduce the fixed indices $\tau_0 = 0$ and $\tau_K = n$. The kth segment is formed by the observations $(y_{\tau_{k-1}+1}, \ldots, y_{\tau_k})$. θ_k stands for the parameter of the kth segment (see Fig. 1). Formally the unconstrained segmentation model [5], can be written as follows:

$$\forall i \mid \quad \tau_{k-1} + 1 \leq i \leq \tau_k, \quad Y_i \sim \mathcal{F}(\theta_k, \phi). \tag{1}$$

Constrained segmentation model

In order to have a segmentation model with a straightforward interpretation in terms of peaks, we add inequality constraints to the successive segment specific parameters $(\theta_1, \ldots, \theta_K)$ so that non-decreasing changes in these parameters are always followed by non-increasing changes. Therefore, we formally assume the following constrained segmentation model [8], hereafter called Up–Down:

$$\begin{aligned} &\forall i \mid \quad \tau_{k-1} + 1 \leq i \leq \tau_k, \quad Y_i \sim \mathcal{F}(\theta_k, \phi) \\ &\text{subject to} \begin{cases} \theta_{k-1} \leq \theta_k & \forall k \in \{2, 4, \ldots\} \\ \theta_{k-1} \geq \theta_k & \forall k \in \{3, 5, \ldots\} \end{cases}. \end{aligned} \tag{2}$$

Probability distributions

In the case of the Poisson distribution we have $\mathcal{F}(\theta_k, \phi) = \text{Pois}(\Lambda_k, \phi = \emptyset)$ where Λ_k stands for the mean and the variance of the kth segment. In the case of the Gaussian distribution we have $\mathcal{F}(\theta_k, \phi) = \mathcal{N}(\mu_k, \sigma^2)$ where μ_k is the mean of the kth segment and σ^2 is the variance assumed constant across the segments. Also in this case, the non-negative integer count data have been transformed in real values ($\mathbb{Z}_+ \to \mathbb{R}_+$) through an Anscombe transformation ($\sqrt{Y + \frac{3}{8}}$) which is a useful variance-stabilizing transformation for count data following a Poisson distribution [1]. In the case of the negative binomial distribution we have $\mathcal{F}(\theta_k, \phi) = \text{NB}(\mu_k, \phi)$ where μ_k is the the mean of the kth segment and ϕ is the dispersion parameter that needs to be learned on the data. In this parametrization σ_k^2, the variance of the kth segment, is $\mu_k + \phi^{-1}\mu_k^2$.

Optimization problems

In both unconstrained and constrained optimal multiple changepoint detection problems, the goal is to estimate the changepoint locations $(\tau_1, \ldots, \tau_{K-1})$ and the parameters $(\theta_1, \ldots, \theta_K)$ both resulting from the segmentation. Runge et al. [19] introduced *gfpop* (Graph-Constrained Functional Pruning Optimal Partitioning), an algorithm that solves both problems using penalized maximum likelihood inference. It implements several loss functions including the Gaussian, Poisson and negative binomial that allowed us to compare different noise models for the count data. The number of changepoints in a coverage profile being unknown, *gfpop* takes a non-negative penalty $\lambda \in \mathbb{R}_+$ parameter that controls the complexity of the output segmentation. Larger penalty λ values result in models with fewer changepoints. The extreme penalty values are $\lambda = 0$ which yields $n - 1$ changepoints, and $\lambda = \infty$ which yields 0 changepoint. The time complexity of *gfpop* is empirically $\mathcal{O}(Vn\log(n))$. Intuitively, V stands for the number states you will need to encode your priors about the form of the output segmentation, e.g. with the Up–Down model at each time the signal can be a part of the background noise (Down) or a peak (Up). Consequently, the empirical time complexity of *gfpop* with the Up–Down model is $\mathcal{O}(2n\log(n))$ while with the unconstrained model it is $\mathcal{O}(n\log(n))$.

Rules for inferring the start and end of peaks with the unconstrained segmentation model

As mentioned before, one of the main motivation of the Up–Down model is that it can be interpreted in terms of peaks which is a practitioner's request. In the case of the unconstrained model, the output segmentation may results in successive non-decreasing changes (Up*), e.g. in Fig. 2: $\text{Up}^* = \{\text{Start1}, \text{Start2}\}$, and successive non-increasing changes (Dw*), e.g. in Fig. 2: $\text{Dw}^* = \{\text{End1}, \text{End2}\}$, in the signal. Thus, it is necessary to specify a post-processing rule to select the start and end of peaks among the returned changepoints in respectively each Up* and Dw*. This results in $|\text{Up}^*| \times |\text{Dw}^*|$ alternatives of the same peak. *Rules.* We propose three different rules to select the start and end of peaks (see Fig. 2):

- *thinnest peak*: we select the last up change in Up* and the first down change in Dw* ;

- *largest peak rule*: we select the first up change in Up* and the last down change in Dw* ;
- *max jump*: we select the up and down change with the largest mean-difference in Up* and Dw*.

Hocking et al. [12] introduced similar rules to the *thinnest peak* and *largest peak*.

Labeled data for supervised learning peak detection

Tuning parameters

For each peak callers there are one or more tuning parameters that need to be set before solving the peak detection problem and that may greatly affect the result accuracy. For segmentation methods this parameter is the penalty λ which controls the number of peaks in the resulting segmentation, while for heuristics, such as MACS or HMCan, they use a threshold parameter whose value allows to only consider the top p peaks given their significance. Moreover, if we want to model the over-dipersion phenomenon observed in the count data using a negative binomial probability distribution, this is done at the cost of another parameter (ϕ) that we need to set as well. Its value may also affect the number of peaks in the resulting segmentation. In theory, if the correct noise model was known, it would be possible to use statistical arguments to choose the parameter to use. However, in practice the correct noise model is complex and unknown. There are many factors that influence the signal and noise patterns in real ChIP-seq data, e.g. experimental protocols, sequencing machines, alignment software. These factors results in poor accuracy for the detection of peaks [7]. Therefore, we will consider the supervised peak detection problem in which the value of tuning parameters can be learned using manually determined labels that indicate a presence or absence of peaks.

Benchmark datasets

Introduced by Hocking et al. [7], these seven labeled histone mark datasets are composed of samples from two different ChIP-seq experiments directed at modifications found on the histone 3 N-terminal tails. The first experiment is directed at histone H3 lysine 4 tri-methylation (H3K4me3), a modification localized in promoters. The second one is directed at histone H3 lysine 36 tri-methylation (H3K36me3), a modification localized in transcribed regions. Both these modifications are involved in the regulation of gene expression [21]. The histone modifications H3K4me3 and H3K36me3 are respectively characterized by sharp and broad peak patterns in coverage profiles. Expert biologists, with visual inspection, have annotated some regions by indicating the presence or absence of peaks. Then, they grouped the labels to form 2752 distinct labeled coverage profiles. Standard used for labeling by the expert biologists is described in Supplementary Text 1 of Hocking et al. [10].

Definition of labeled coverage profiles and errors

In the context of supervised peak detection each labeled coverage profile of size n, denoted $w \in \mathbb{Z}_+^n$, is a problem. Formally we have a set of M problems $(w_1, \ldots, w_M)$ where $M = 2752$. Each problem w_m is associated with a set of N labels $H_m = \{(s_1, e_1, h_1) \ \ldots, (s_N, e_N, h_N)\}$ where s is the start genomic location of the label, e is

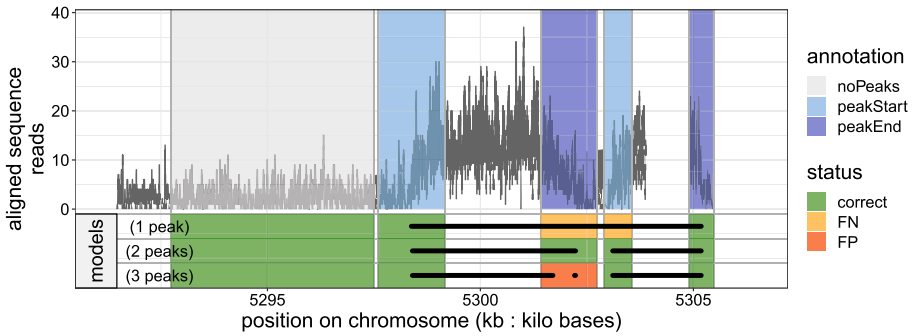


Fig. 3 (**Top**) Example of a ChIP-seq coverage profile annotated by an expert biologist. The labels represented by colored rectangles indicate the absence or presence of a peak, here characterized by its start and its end. (**Bottom**) The model with 1 peak in its output segmentation has an associated error of 2 (2 × False Negative). The model with 3 peaks has an associated error of 1 (1 × False Positive). The model with 2 peaks is a good model for which all the labels optimized on this coverage profile are correct

the end genomic location of the label and h is the type of the label. There are four types of labels that allow some flexibility in the annotation (see Fig. 3):

- *noPeaks* label stands for a region that contains only background noise with no peak. If any peak is predicted in this region, the label counts as a false positive ;
- *peaks* label means there is at least one overlapping peak in that region. Hence, one or more peaks in that region is acceptable. If there is not at least one overlapping peak predicted in this region, it counts as a false negative ;
- *peakStart* and *peakEnd* labels stand for regions which should contain exactly one peak start or end. If more than one peak start / end is predicted in this region, the label counts as a false positive. Conversely, if less than one peak start / end is predicted in this region, the label counts as a false negative.

The set of labels H_m is used to quantify the error E_m, i.e. the total number of incorrectly predicted labels (false positive + false negative) in the coverage profile w_m given the set of peaks returned by a peak caller.

Supervised algorithms for learning tuning parameters of negative binomial segmentation models

Objective function

The error function for a given problem w_m, denoted $E_m : \mathbb{R}^2_+ \to \mathbb{Z}_+$, is a mapping from the tuning parameters (ϕ, λ) of negative binomial segmentation models to the number of incorrectly predicted labels in the resulting optimal segmentation. With the supervised peak detection approach the goal is to provide predictions of ϕ and λ that minimize $E_m(\phi, \lambda)$. The exact computation of the 2-dimensional defined $E_m(\phi, \lambda)$ is intractable with respect to ϕ. Thus, we computed it over 16 ϕ values evenly placed on the log scale between 1 and 10,000, $\Phi = (\phi_1 = 1, \ldots, \phi_{16} = 10{,}000)$. Our results suggest that this grid of values is a good set of candidates to test in order to calibrate the dispersion parameter ϕ (see Additional file 1: Fig. 2). The exact computation of the error rate as a function of λ (ϕ remains constant), a piecewise constant function,

requires to retrieve all optimal segmentations up to 9 peaks. This way, on the advice of the biologists who annotated the benchmark datasets, we ensure that for each problem there is a segmentation with at least one false positive label and another with one false negative label. A procedure that retrieves one optimal segmentation for each targeted number of peaks P^* has already been described by Hocking et al. [11]. It can be used with the Up–Down model for which there is at most one optimal segmentation that results in P^* peaks but not with the unconstrained model for which there can be several ones. Indeed, the constraints in the Up–Down model require it to add, if the associated cost is optimal, 2 changepoints that lead to the formation of a new peak. With the unconstrained model adding a changepoint can either refine an already existing peak or, in combination with another changepoint, form a new peak. More generally there is a need of an algorithm that takes as input any penalized changepoint detection solver $\mathcal{S}$ with a penalty λ constant along the changepoints, optionally the dispersion parameter ϕ, and outputs all optimal segmentations between two peak bounds denoted $\underline{P}$ and $\overline{P}$. We present *CROCS* (Changepoints for a Range of ComplexitieS), an algorithm that meets this need.

Discussion of pseudocode

CROCS (*Algorithm* 1).**(i)** The algorithm begins by calling *SequentialSearch* [described underneath] to search two penalty bounds $\overline{\lambda}$ (line 6) and $\underline{\lambda}$ (line 5) that result in a segmentation with respectively $\underline{P}-1$ (line 3) and $\overline{P}+1$ (line 4) peaks. Indeed, using *gfpop* with the Up–Down model as solver $\mathcal{S}$, the number peaks in the resulting optimal segmentations is a non-increasing function of λ. This propriety guarantees that with the previous penalty bounds we can reach every optimal model from $\underline{P}$ to $\overline{P}$ peaks. For unconstrained segmentation models, we suspect it also should be true in the vast majority of cases. **(ii)** Then, the algorithm calls *CROPS* [described underneath] (line 7) to retrieve all the optimal segmentations between these two penalty bounds. **(iii)** Finally, a simple post-processing step (not shown in the algorithm) allows to remove segmentations with $P-1$ and $P+1$ peaks. The time complexity of the *CROCS* algorithm is bounded by the time complexity of the *CROPS* procedure, i.e. $\mathcal{O}(\mathcal{O}(\mathcal{S})(K_{\underline{\lambda}}-K_{\overline{\lambda}}))$, where $K_{\overline{\lambda}}$ and $K_{\underline{\lambda}}$ are the number of segments in optimal segmentations associated to respectively $\overline{\lambda}$ and $\underline{\lambda}$. $\mathcal{O}(\mathcal{S})$ is the time complexity of the solver $\mathcal{S}$, e.g. empirically $\mathcal{O}(2n\log(n))$ for *gfpop* with the Up–Down model.

- *SequentialSearch* is a procedure described by Hocking et al. [11] that takes as input a problem w_m, a target number of peaks P^* and outputs an optimal segmentation with P^* peaks in addition to the penalty λ for reaching it.
- *CROPS* is a procedure described by Haynes et al. [6] that takes as input a problem w_m, as well as two penalty bounds $\underline{\lambda}$ & $\overline{\lambda}$ and outputs all the optimal segmentations between these two bounds.

We slightly modified the original implementation of both *SequentialSearch* and *CROPS* in such way that they can work with any penalized changepoint detection solver $\mathcal{S}$ provided by the user.

Algorithm 1 *CROCS* (Changepoints for a Range of ComplexitieS): extract all optimal segmentations between $\underline{P}$ and $\overline{P}$ using a changepoint penalized solver $\mathcal{S}$

Input: Data w_m, lower bound $\underline{P}$, upper bound $\overline{P}$, solver $\mathcal{S}$, dispersion ϕ (optional)
Output: The details of optimal segmentations between $\underline{P}$ and $\overline{P}$ peaks
if $\underline{P} > 0$: $\underline{P} \leftarrow \underline{P} - 1$
$\overline{P} \leftarrow \overline{P} + 1$
$\underline{\lambda} \leftarrow$ *SequentialSearch*$(w_m, \overline{P}, \mathcal{S}, \phi)$ ▷ Hocking et al. [11]
$\overline{\lambda} \leftarrow$ *SequentialSearch*$(w_m, \underline{P}, \mathcal{S}, \phi)$
return *CROPS*$(w_m, \underline{\lambda}, \overline{\lambda}, \mathcal{S}, \phi)$ ▷ Haynes et al. [6]

Learning jointly ϕ and λ

Once the error function $E_m(\phi \in \Phi, \lambda)$ is computed for each problem of the training set, a natural way to learn the dispersion and penalty parameters is to select the pair of values $(\phi \in \Phi, \lambda)$ that achieves the global minimum error. We denote these values ϕ^* and λ^*. Recall that $E_m(\phi \in \Phi, \lambda)$ is piecewise constant on λ. The sum of $E_m(\phi \in \Phi, \lambda)$ over all problems is still piecewise constant on λ. Therefore, ϕ^* and λ^* can be easily retrieved using a sequential search. We refined the previous learning method, hereafter called *constant* λ, by taking advantage of the piecewise constant propriety of $E_m(\phi \in \Phi, \lambda)$. Indeed, the minimum error is not reached for a unique penalty value λ^* but an interval denoted $I_{\lambda,m}$. After fixing ϕ^*, we can use $I_{\lambda,m}$ computed for each problem of the training set in order to learn a function that predicts problem-specific λ values. This function is a solution of the interval regression problem described by Rigaill et al. [16]. We denote this learning method *linear* λ.

In the case of segmentation models with a Poisson or a Gaussian noise, the only tuning parameter that we need to learn is λ. Thus, the objective function becomes a 1-dimensional defined function denoted $E_m(\lambda)$. The methods we used to learn λ are similar than those presented above (see Hocking et al. [12] for more details).

Empirical results

Cross-validation setup and evaluation metric

In the following section, for each model compared, a 10-fold or 4-fold[1] cross-validation was performed on each of the seven datasets. Here, the results are shown by type of experiments (H3K36me3 & H3K4me3). The metric we used to evaluate the performance of our models is the test accuracy which can be formally written $1 - \left(\sum_{m \in \text{ test set}} E_m \,/\, \sum_{m \in \text{ test set}} |H_m|\right)$. One may be concerned about the size of the datasets used for supervised learning of the tuning parameters. We have shown in Additional file 1: Fig. 1 that only a dozens of labels are enough to learn tuning parameters and associated segmentations close to the model-specific maximum accuracy. By increasing the number of labels in the learning set, the accuracy also becomes more consistent between test folds.

[1] In order to satisfy the assumption about the independence between the training and test set in the cross-validation, we could not exceed 4-fold in two of the seven benchmark datasets (for more details see caption of Additional file 1: Table 1).

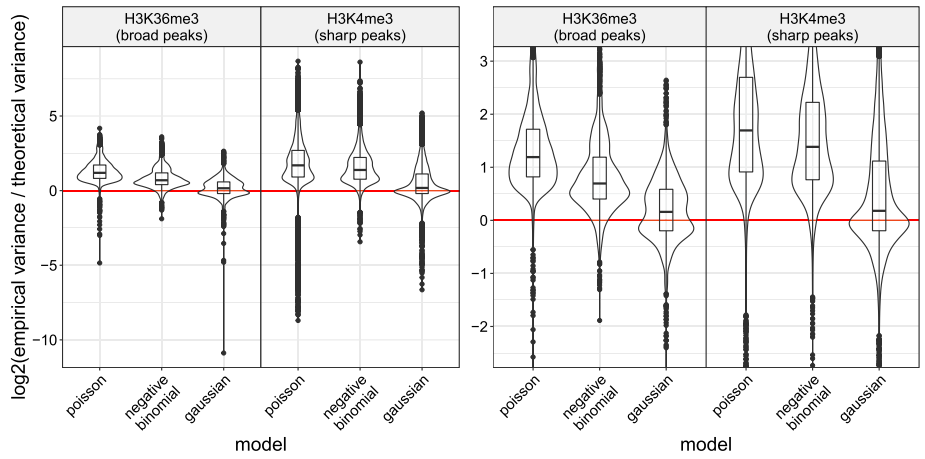


Fig. 4 The over-dispersion exhibited by count data under a Poisson noise model can be effectively reduced using a negative binomial or a Gaussian transformed noise model. The red indicator line stands for the equality of the theoretical and empirical variances. (**Left**) Observations above this line stand for over-dispersed count data. Observations under this line stand for under-dispersed count data. (**Right**) A zoom on the **Left** distributions

Learning of tuning parameters

In previous section we have described two methods for learning the tuning parameters of segmentation models. Based on results shown in Additional file 1: Fig. 3, for the rest of this section, the parameters of the models compared on H3K36me3 datasets are learned through the *constant* λ method. The parameters of the models compared on H3K4me3 datasets are them learned through the *linear* λ method.

The over-dispersion exhibited by count data under a Poisson noise model can be effectively reduced using a negative binomial or a Gaussian transformed noise model

Initially, we wanted to validate the presence of over-dispersion in count data following a Poisson distribution. In a second step, we wanted to confirm that alternative noise models such as the negative binomial or the Gaussian one, following an Anscombe transformation of the counts for the latter, could allow us to reduce this over-dispersion. A simple way to highlight the over-dispersion is to plot the $\log_2$ -ratio of the empirical and theoretical variances of count data. If the $\log_2$ -ratio is positive, the distribution of count data exhibits over-dispersion. If it is negative, the distribution of count data exhibits under-dispersion. If it is null, the dispersion of the count data does not show inconsistency with respect to the noise model. In Fig. 4, each observation corresponds to a segment from the segmentations selected during the cross-validation procedure for the 2752 coverage profiles. The segmentation were computed using *CROCS* with *gfpop* and the unconstrained model as solver. Then, We estimated the empirical and theoretical variances for each of the selected segments. In the case of the Poisson noise model, the estimated theoretical variance is formally written $\hat{\sigma_k}^2 = \hat{\mu}$, where $\hat{\mu}$ stands for the estimation of the mean of count data belonging to the same segment. For the negative binomial one it is formally written $\hat{\sigma}^2 = \hat{\mu} + \phi^{-1}\hat{\mu}^2$, where ϕ stands for the dispersion parameter learned during the cross-validation

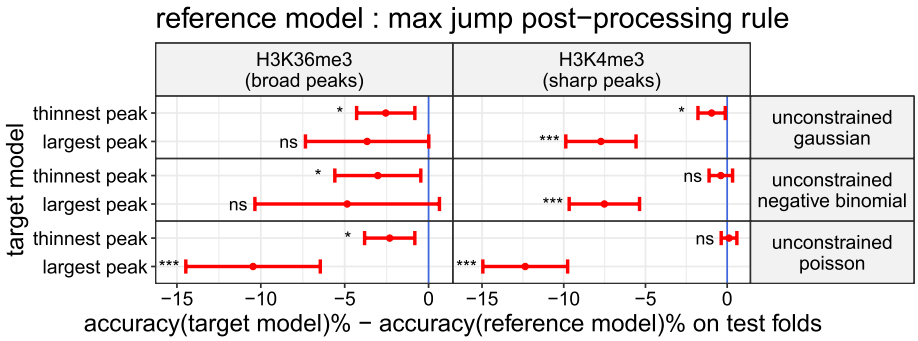


Fig. 5 Max jump is the most accurate rule for inferring the peaks in segmentations obtained through a unconstrained model. The mean of differences in accuracy and its 95% CI computed on the test folds pooled by types of experiment (H3K36me3 & H3K4me3) are shown in red. If the mean of differences in accuracy is negative (left side of the blue indicator line), the *max jump* rule is better in average than the target rule. The results of the paired t-test used to assess the difference of mean accuracy are summarized in the following way: non significant (ns) means adjusted p-value $>$ 0.05; * means adjusted p-value $\leq$ 0.05; *** means adjusted p-value $\leq$ 0.001

procedure. For the Gaussian one, the theoretical variance is assumed constant across the segments. We estimated it using the mean squared error computed over all segments. In Fig. 4 we can see that in both H3K36me3 and H3K4me3 datasets the median of the $\log_2$-ratio is above 1 with the Poisson noise model. Hence, For most observations the empirical variance is at least two times larger than the theoretical variance. Therefore, count data under the Poisson noise model shows a clear over-dispersion phenomenon. In both H3K36me3 and H3K4me3 datasets, the median of the $\log_2$-ratio is slightly closest to 0 with the negative noise model than with Poisson noise one (from 1.19 to 0.70 in H3K36me3 and 1.69 to 1.39 in H3K4me3). Therefore, the negative noise model helps partially correct this over-dispersion. The reduction is even greater with the Gaussian transformed noise model (from 1.19 to 0.16 in H3K36me3 and 1.69 to 0.18 in H3K4me3).

Max jump is the most accurate rule for inferring the peaks in segmentations obtained through the unconstrained model

Solving the peak detection problem with the unconstrained model requires to introduce a rule for selecting the changepoints corresponding to the start and end of the peaks in the output segmentation. We wanted to compare the peak detection accuracy of the new rule we propose (*max jump*) against the others (*largest peak* & *thinnest peak*) which have an equivalence in Hocking et al. [12]. In the user guide of how to create labels in ChIP-seq coverage profiles [7], the authors strongly advise to label peaks which are obviously up with respect to the background noise. Hence, we expected that the *max jump* rule, which sets the start and end of the peaks on the change with the largest mean-difference, performs at least as well as the other two rules. In Fig. 5, we look at the mean of differences in accuracy between each model with either the *largest peak* or *thinnest peak* rule, denoted target models, against the same model with the *max jump* rule, denoted reference model. In agreement with our expectation, we observe that for the different models in both H3K36me3 & H3K4me3 datasets, the mean accuracy of the *max jump*

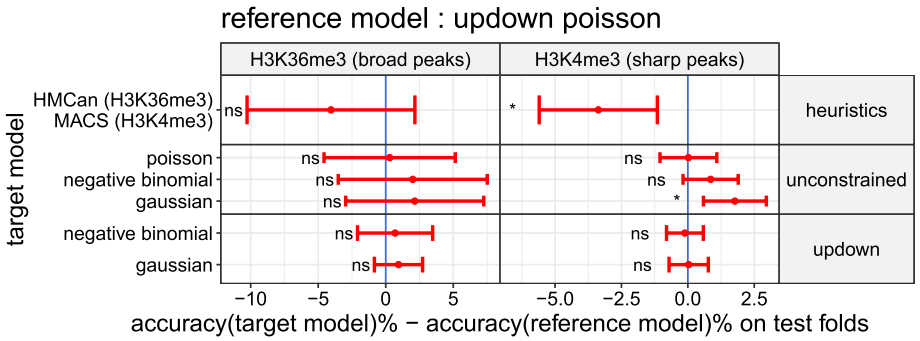


Fig. 6 The unconstrained model with a negative binomial or a Gaussian transformed noise is more accurate than previous state-of-the-art. The mean of differences in accuracy and its 95% CI computed on the test folds pooled by types of experiment (H3K36me3 & H3K4me3) are shown in red. If the mean of differences in accuracy is negative (left side of the blue indicator line), the Up–Down model with a Poisson noise is better in average than the target model. The results of the paired t-test used to assess the difference of mean accuracy are summarized in the following way: non significant (ns) means adjusted p-value $>$ 0.05; * means adjusted p-value $\leq$ 0.05

rule is greater than the mean accuracy of the *largest peak* rule (3.66–12.36% more accurate on average). Except for the unconstrained model with a Poisson noise in H3K4me3 (0.11% less accurate on average), the mean accuracy of the *max jump* rule is also greater than the mean accuracy of the *thinnest peak* (0.38–3.03% more accurate on average). In order to test if the mean accuracy of the target and the reference models are significantly different, we performed a paired t-test. The accuracy of each fold were previously pooled by type of experiments as it is suggested in Fig. 5. After correcting the p-values of the paired t-test with the Benjamini & Hochberg method, eight differences were still significant (adjusted p-value $<$ 0.05). As a result of these observations, for the next comparisons we will infer the peaks in the output segmentations obtained with the unconstrained model using the new *max jump* rule we propose.

The unconstrained model with a negative binomial or a Gaussian transformed noise is more accurate than previous state-of-the-art

We wanted to compare the peak detection accuracy of the Up–Down model with a Poisson noise[2] against other segmentation models such as the unconstrained or Up–Down model with either a negative binomial or a Gaussian transformed noise. HMCan, MACS and other heuristics have already been compared to the Up–Down model with a Poisson noise in Hocking et al. [12]. We included them again as a baseline from the bioinformatics literature. Both of them use a threshold that affects their peak detection accuracy and whose learning is also described in the previous cited study. Because we saw in previous results that a negative binomial or Gaussian transformed noise effectively reduces the over-dispersion exhibited by count data under a Poisson noise, we expected that the unconstrained or Up–Down model with these alternative noises will improve the peak detection accuracy on the test set. In Fig. 6 we look at the mean of differences in

[2] Model built on natural assumptions to detect peaks in coverage profiles and actual state-of-the-art on H3K36me3 and H3K4me3 datasets.

accuracy between the Up–Down model with a Poisson noise, denoted reference model, against other segmentation models and heuristics, denoted target models. In agreement with our expectation, we can see that the unconstrained model with a negative binomial noise has a mean accuracy greater than the reference model in both H3K36me3 and H3K4me3 datasets (respectively 2.0% and 0.86% more accurate on average). It has also a greater mean accuracy with a Gaussian transformed noise (respectively 2.15% and 1.77% more accurate on average). As described previously, in order to test if the mean accuracy of the target and the reference models are significantly different, we performed a paired t-test. After correcting the p-values, the unconstrained model with a Gaussian transformed noise was still significant (adjusted p-value < 0.05). Note that the unconstrained model with a Poisson noise has a mean accuracy similar to reference model (the mean of differences in accuracy $< 0.5\%$ in both datasets). Thus, the improvement in accuracy cannot be attributed solely to the unconstrained model with the *max jump* rule but also to the distribution chosen for the noise. In disagreement with our expectation, with the Up–Down model the use of alternative noise distributions does not improve significantly the accuracy compared to the Poisson one (mean of differences in accuracy $< 1\%$ in H3K36me3 and $< 0.1\%$ in H3K4me3).

The Up–down segmentation models are more robust than the heuristics from the bioinformatics literature HMCan and MACS

In addition to comparing the peak detection accuracy, we wanted to assess the robustness of segmentation models against the heuristics HMCan and MACS. To assess the robustness of the segmentation models and heuristics we used the coverage profiles from biological replicates available in each of the seven labeled histone mark datasets. The value of tuning parameters for the segmentation models and heuristics are the same as those learned during the cross-validation procedure. As explained in the introduction, the peak calling problem can be seen as a binary classification problem. In this framework each base from the coverage profiles are classified as a part of the background noise (0) or peak (1). Hence, the robustness can be assessed by computing the distance between partitions of the coverage profiles from the biological replicates. The more the distance between these partitions is close to zero the more the segmentation model or the heuristic is robust. As a metric we used the normalized information distance, or NID, which has a range between 0 and 1 [3, 23]. For each genomic chunk we computed the NID between all pairs of biological replicates. In Fig. 7 we look a the mean of differences of NID between segmentation models and the heuristics HMCan or MACS. We can see that the mean of the NID of Up–Down models, independently of the noise model, is lower than with the heuristics HMcan and MACS in both H3K36me3 and H3K4me3 datasets (respectively from 0.09 to 0.12 and 0.02 to 0.03 less distant on average). After correcting the p-values of the paired t-test with the Benjamini & Hochberg method, five differences were still significant (adjusted p-value < 0.05). Regarding the unconstrained models, except for the negative binomial noise model in the H3K24me3 datasets (NID is lower by 0.09 in average & paired t-test with adjusted p-value < 0.01), there is no clear improvement in robustness compared to the heuristics HMCan or MACS. With the Poisson model, which do no reduce the over-dispersion, we conclude

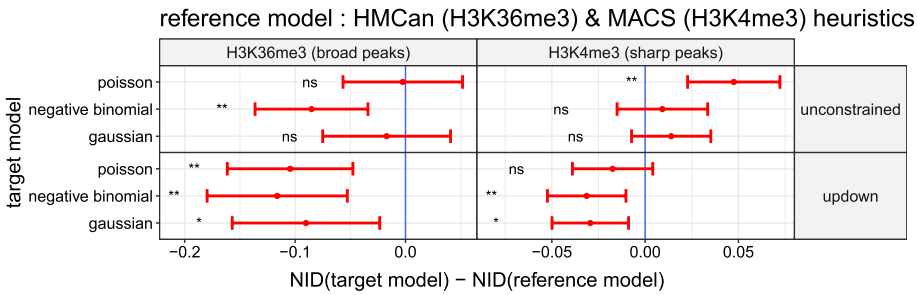


Fig. 7 The Up–down segmentation models are more robust than the heuristics from the bioinformatics literature HMCan and MACS. The mean of differences of the normalized information distance (NID) and its 95% CI are shown in red. If the mean of differences in NID is negative (left side of the blue indicator line), the target segmentation model is more robust in average than HMCan (H3K36me3) or MACS (H3K4me3). The results of the paired t-test used to assess the difference of mean of NID are summarized in the following way: non significant (ns) means adjusted p-value $>$ 0.05; * means adjusted p-value $\leq$ 0.05; ** means adjusted p-value $\leq$ 0.01

even the opposite in the H3K4me3 datasets (NID is longer by 0.05 in average, paired t-test with adjusted p-value $<$ 0.01).

Discussion

Modeling of over-dispersed ChIP-seq count data

We have seen in Fig. 4 that count data under a Poisson noise model exhibit over-dispersion in H3K36me3 and H3K4me3 datasets. We have shown that this over-dispersion can be effectively reduced in these datasets using either a negative binomial or a Gaussian transformed noise model.

The use of a negative binomial noise model implies that we must be able to estimate a suitable value for the ϕ dispersion parameter. We have proposed to learn it jointly with the penalty of the segmentation model directly on the labeled coverage profiles. More precisely, a constant ϕ is selected because it minimizes the label errors of the training set. The negative binomial combined with the constant dispersion parameter allows the phenomenon of over-dispersion to be slightly reduced.

With the Gaussian noise model there are no additional parameters than the penalty of the segmentation model to set. This is an advantage compared to the negative binomial one. In this study, in order to satisfy the Gaussian proprieties, we transformed the count data with an Anscombe transformation which is highly appreciated for its variance stabilization properties. Gaussian transformed noise model allowed to reduce the over-dispersion even more efficiently than the negative binomial noise model on the H3K4me3 and H3K36me3 datasets, while being simpler to implement.

Segmentation models for peak detection in ChIP-seq count data

The unconstrained model seems to capture more subtle changes in count data than the Up–Down one which have sometimes a poor fit to the signal (see Fig. 1). One major issue of the unconstrained model is its output segmentation which doesn't have a straightforward interpretation in terms of peaks compared to the Up–Down one. The introduction of the *max jump* rule (see Fig. 2), which have shown to perform at least as

well as rules proposed in Hocking et al. [12] (*thinnest peak* & *largest peak*), helps to correct this weakness (see Fig. 5).

In Fig. 6 we have seen that when combining the unconstrained model with a negative binomial or a Gaussian transformed noise it is possible to improve upon the natural and current state-of-the-art on the peak detection accuracy, the Up–Down model with a Poisson noise, in both H3K36me and H3K4me3 datasets. We argue that this improvement is likely explained by the ability of the negative binomial and the Gaussian transformation to reduce the over-dispersion as illustrated in Fig. 4. In summary, we believe that the better we model dispersion the better we improve the accuracy of the segmentation model. Figure 7 have shown that the unconstrained segmentation model with noise models reducing over-dispersion are also at least as robust as MACS or HMCan heuristics. It is an important criterion showing the applicability of our proposed models.

Still in Fig. 6, we have seen that the Up–Down model with a negative binomial or a Gaussian transformed noise, which reduce the over-dispersion phenomenon, doesn't improve the accuracy upon the Up–Down model with a Poisson noise. One hypothesis to explain these results is that the constraints, which lead to the reduction of the space of optimal reachable segmentations with the Up–Down model, also reduce the probability of adding biologically uninformative changepoints induced by the over-dispersion. Consequently, the Up–Down model has the advantage to be a model with good internal over-dipsersion resistance properties but is bounded by its poor adaptability to the signal. We argue the constraints also explain that the Up–Down model is more robust than the unconstrained model and the MACS and HMcan heuristics (see Fig. 7).

We have added several supplementary figures (see Additional file 1: Figs. 4–10) which illustrate typical results from the test folds for the MACS and HMCan heuristics as well as our proposed segmention models.

Segmentation models applied to other types of ChIP-seq experiments

In this paper, the broad (H3K36me3) and sharp (H3K4me3) histone signals have been discussed. Previous studies already demonstrated the applicability of optimal changepoint algorithms to other types of experiment. For example, Fig. 7 in Hocking and Bourque [9] showed that optimal changepoint algorithms on H3K9me3 and H3K27me3 data typically result in peaks with intermediate sizes (3.5–3.9 kb on average) compared with the relatively small H3K4me3 (1.0–1.7 kb) and relatively large H3K36me3 (35.8–48.0 kb). The peak calling of transcription factor binding sites such as MAX, SRF and NRSF was also previously tested (see Supplementary Fig. 3 in Hocking et al. [7]). By reducing the over-dispersion in count data with the Gaussian transformed or the negative binomial noise models, we would expect similar improvements in accuracy for these other experiment types. Furthermore, we did not test our proposed models on mixed signal like Pol II. We leave the two last points for future research.

Conclusion

We developed the *CROCS* algorithm that computes all optimal models between two peak bounds, given any segmentation algorithm with constant penalty λ for each changepoint. This set of optimal segmentations is essential to compute the error rate function, which is in turn used in the supervised approach for learning the tuning parameters of

the segmentation models. We proposed to solve the peak detection problem by using the unconstrained segmentation model that takes advantage of the *max jump* rule we introduced as well as the negative binomial or Gaussian transformed noise model. We have shown that this model improves upon the accuracy of the model built on natural assumptions (constrained segmentation (Up–Down) with Poisson noise model) in both H3K36me3 and H3K4me3 datasets. The unconstrained model with the negative binomial or Gaussian transformed noise model can be used to provide new high-quality peak prediction tracks for H3K36me3 and H3K4me3 histone modifications. These peak prediction tracks will be a more accurate reference for researchers in the field of epigenetics who want to analyze these data.

Future work

Our results suggest that with both negative binomial and Gaussian transformed noise models the over-dispersion could be further reduced. Regarding the negative binomial noise model, one could think about predicting a local dispersion parameter for each coverage profile. Furthermore, the literature about Gaussian transformations is wide and a comparative study integrating segmentation models with different transformations for count data, e.g. the Box–Cox transformation, arcsin square root transformation or log-transformation, would also be an interesting avenue for future work. As described in Anscombe [1] some of these well-known transformations have, in theory, better variance-stabilizing proprieties for over-dispersed count data than the one we used in this study ($\sqrt{Y + \frac{3}{8}}$). Still, they are highly dependent on the estimation of the dispersion parameter ϕ which in our case can be directly taken into account in the statistical model, i.e by using the negative binomial noise model implemented in *gfpop*.

In this paper we explored two different segmentation models, the unconstrained segmentation model and a constrained segmentation model where each non-decreasing change is followed by an non-increasing change in the mean (Up–Down). The *gfpop* method makes it possible to model changepoints even more precisely by constraining for example the minimum size of jumps or the minimum size of segments. It would be interesting in future work to test other constrained models or to model the auto-correlation [4, 18] in the context of the peak detection problem in ChIP-seq data.

Abbreviations

ChIP-seq: Chromatin immunoprecipitation followed by high-throughput sequencing; Up–Down: Constrained segmentation model where each non-decreasing change is followed by an non-increasing change in the mean; gfpop: Graph-constrained functional pruning optimal partitioning; NID: Normalized information distance.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-021-04221-5.

Additional file 1. Supplementary materials for "Increased peak detection accuracy in over-dispersed ChIP-seq data with supervised segmentation models".

Acknowledgements

Not applicable.

Authors contributions

AL designed the study, wrote the code, performed the analyses, interpreted the results and wrote the manuscript. GR and TH supervised and designed the study, helped with the code, interpreted the results and provided manuscript

feedback. The authors wish it to be known that, in their opinion, GR and TH should be regarded as joint last authors. All authors read and approved the final manuscript.

Funding
AL and TH were funded by a Northern Arizona University startup grant. GR was supported by an ATIGE grant from Genopole. The IPS2 benefits from the support of the LabExSaclay Plant Sciences-SPS.

Availability of data and materials
The labeled histone mark data are available here: https://rcdata.nau.edu/genomic-ml/chip-seq-chunk-db/. The scripts used to compute the models and analyse the results are available in the "aLiehrmann/chip_seq_segmentation_paper" *GitHub* repository: https://github.com/aLiehrmann/chip_seq_segmentation_paper. A reference implementation of the *CROCS* algorithm is available in the R package of the same name: https://github.com/aLiehrmann/CROCS. The package's vignette describes the supervised learning procedure and the user can easily adapt the code to his own data.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Institut des Sciences des Plantes de Paris-Saclay (IPS2), Université Paris-Saclay, Université Evry, CNRS, INRAE, 91405 Orsay, France. [2]Laboratoire de Mathématiques et Modélisation d'Evry (LAMME), Université Paris-Saclay, Université Evry, CNRS, 91037 Evry, France. [3]School of Informatics, Computing, and Cyber Systems (SICCS), Northern Arizona University, 86011 Flagstaff, AZ, USA.

Received: 21 January 2021 Accepted: 19 May 2021
Published online: 14 June 2021

References

1. Anscombe FJ. The transformation of poisson, binomial and negative-binomial data. Biometrika. 1948;35:246–54.
2. Ashoor H, Herault A, Kamoun A, Radvanyi F, Bajic VB, Barillot E, Boeva V. Hmcan: a method for detecting chromatin modifications in cancer samples using chip-seq data. Bioinformatics. 2013;29:2979–86.
3. Chiquet J, Rigaill G, Sundqvist M. Aricode: efficient computations of standard clustering comparison measures (2020). https://CRAN.R-project.org/package=aricode
4. Cho H, Fryzlewicz P. Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. J R Stat Soc Ser B (Statistical Methodology). 2015;77:475–507.
5. Cleynen A, Lebarbier E. Segmentation of the poisson and negative binomial rate models: a penalized estimator. ESAIM Prob Stat. 2014;18:750–69.
6. Haynes K, Eckley IA, Fearnhead P. Computationally efficient changepoint detection for a range of penalties (2017)
7. Hocking TD, Goerner-Potvin P, Morin A, Shao X, Pastinen T, Bourque G. Optimizing chip-seq peak detectors using visual labels and supervised machine learning. Bioinformatics. 2017;33:491–9.
8. Hocking T, Rigaill G, Bourque G. Peakseg: constrained optimal segmentation and supervised penalty learning for peak detection in count data. Proc Mach Learn Res. 2015;37:324–32.
9. Hocking TD, Bourque G. Machine learning algorithms for simultaneous supervised detection of peaks inmultiple samples and cell types. Pac Symp Biocomput. 2020;25:367–78.
10. Hocking TD, Rigaill G, Fearnhead P, Bourque G. A log-linear time algorithm for constrained changepoint detection. arXiv:1703.03352 (2017)
11. Hocking TD, Rigaill G, Fearnhead P, Bourque G. Generalized functional pruning optimal partitioning (GFPOP) for constrained changepoint detection in genomic data. arXiv:1810.00117 (2018)
12. Hocking TD, Rigaill G, Fearnhead P, Bourque G. Constrained dynamic programming and supervised penalty learning algorithms for peak detection in genomic data. J Mach Learn Res. 2020;21:1–40.
13. Law CW, Chen Y, Shi W, Smyth GK. voom: precision weights unlock linear model analysis tools for rna-seq read counts. Genome Biol. 2014;15.
14. Love M, Huber W, Anders S. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. Genome Biol. 2014;15.
15. Marinov GK. A decade of chip-seq. Brief Funct Genom. 2018;17:77–9.
16. Rigaill G, Hocking T, Vert J-P, Bach F. Learning sparse penalties for change-point detection using max margin interval regression. Proc Mach Learn Res. 2013;28:172–80.
17. Robinson MD, McCarthy DJ, Smyth GK. edger: a bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics. 2010;26:139–40.
18. Romano G, Rigaill G, Runge V, Fearnhead P. Detecting abrupt changes in the presence of local fluctuations and autocorrelated noise. arXiv:2005.01379 (2020)

19. Runge V, Hocking TD, Romano G, Afghah F, Fearnhead P, Rigaill G. gfpop: an R package for univariate graph-constrained change-point detection. arXiv:2002.03646 (2020)
20. Schmidt D, Wilson MD, Ballester B, Schwalie PC, Brown GD, Marshall A, Kutter C, Watt S, Martinez-Jimenez CP, Mackay S, Talianidis I, Flicek P, Odom DT. Five-vertebrate chip-seq reveals the evolutionary dynamics of transcription factor binding. Science. 2010;1036–1040.
21. Sims RJ, Nishioka K, Reinberg D. Histone lysine methylation: a signature for chromatin function. Trends Genet. 2003;19:629–39.
22. Valouev A, Johnson DS, Sundquist A, Medina C, Anton E, Batzoglou S, Myers RM, Sidow A. Genome-wide analysis of transcription factor binding sites based on chip-seq data. Nat Methods. 2008;5:829–34.
23. Vinh NX, Epps J, Bailey J. Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J Mach Learn Res. 2010;11:2837–54.
24. Young MD, Willson TA, Wakefield MJ, Trounson E, Hilton DJ, Blewitt ME, Oshlack A, Majewski IJ. Chip-seq analysis reveals distinct h3k27me3 profiles that correlate with transcriptional activity. Nucl Acids Res. 2011;7415–7427.
25. Zhang B, Zheng H, Huang B, Li W, Xiang Y, Peng X, Ming J, Wu X, Zhang Y, Xu Q, Liu W, Kou X, Zhao Y. Allelic reprogramming of the histone modification h3k4me3 in early mammalian development. Nature. 2016;537:553–7.
26. Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, Nusbaum C, Myers RM, Brown M, Li W, Liu XS. Model-based analysis of chip-seq (macs). Genome Biol. 2008;9.

Publisher's Note

Supplementary materials for "Increased peak detection accuracy in over-dispersed ChIP-seq data with supervised segmentation models"

datasets	type of ChIP-Seq experiment	number of folds
H3K36me3_AM_immune	H3K36me3 (broad peaks)	10
H3K36me3_TDH_immune	H3K36me3 (broad peaks)	4
H3K36me3_TDH_other	H3K36me3 (broad peaks)	4
H3K4me3_PGP_immune	H3K4me3 (sharp peaks)	10
H3K4me3_TDH_immune	H3K4me3 (sharp peaks)	10
H3K4me3_TDH_other	H3K4me3 (sharp peaks)	10
H3K4me3_XJ_immune	H3K4me3 (sharp peaks)	10

Table 1: Summary of the number of folds in the cross-validation procedure by dataset. Two of the seven labeled histone mark datasets, i.e. H3K36me3_TDH_immune & H3K36me3_TDH_other, can be considered small datasets as they include biological replicates from four independent genomic chunks. In order to satisfy the assumption of independence between the training and test set in the cross-validation, we could not exceed 4-fold for both of them.

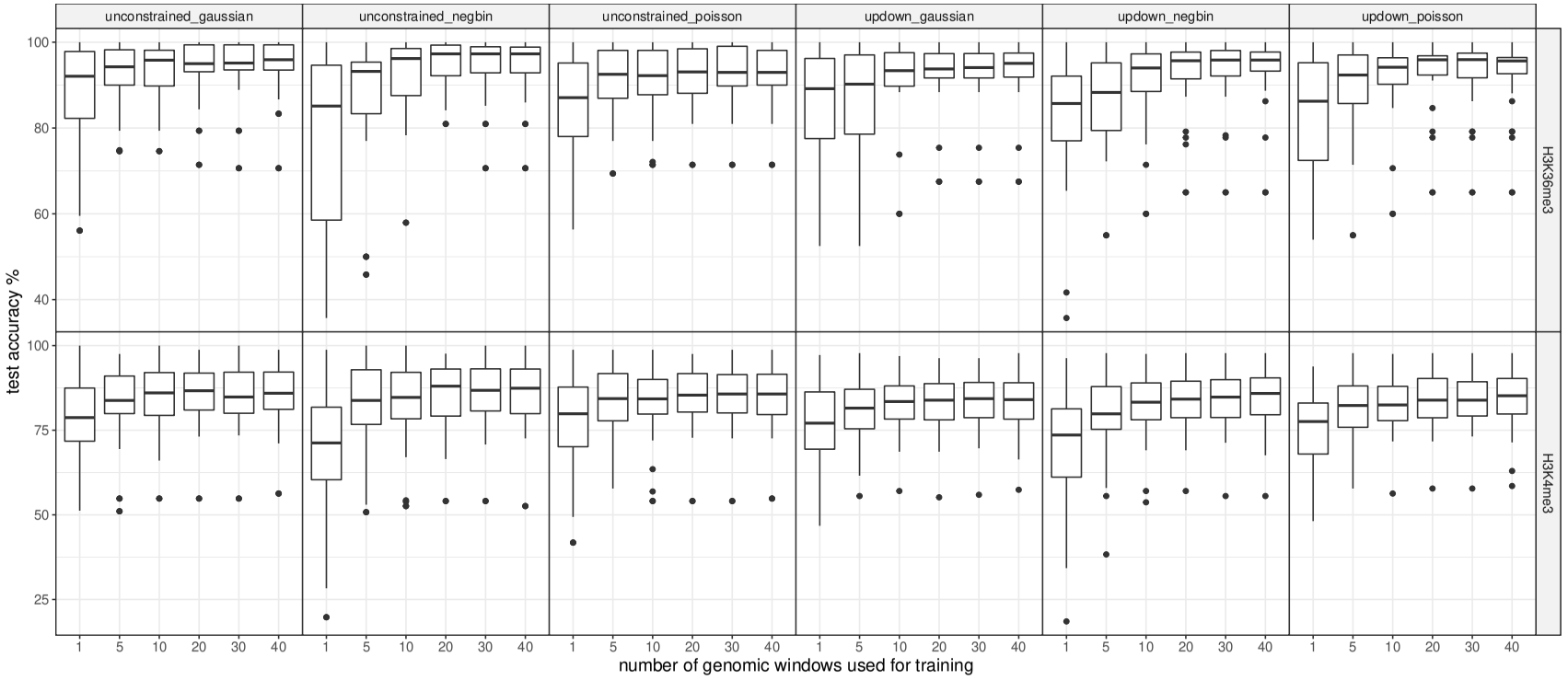


Figure 1: To determine how many labels are necessary to learn a model close to the maximum accuracy we used the cross-validation procedure described in the manuscript. Each test fold is assessed for a variable number of labeled genomic windows from the training folds (from 1 to 40). In all datasets, 10 genomic windows, with on average 5 labels per window, are enough to learn penalties and associated segmentation close to the model-specific maximum accuracy. Note that in addition to improving the average accuracy on test folds, a few genomic windows are enough to effectively reduce the variance of the results.

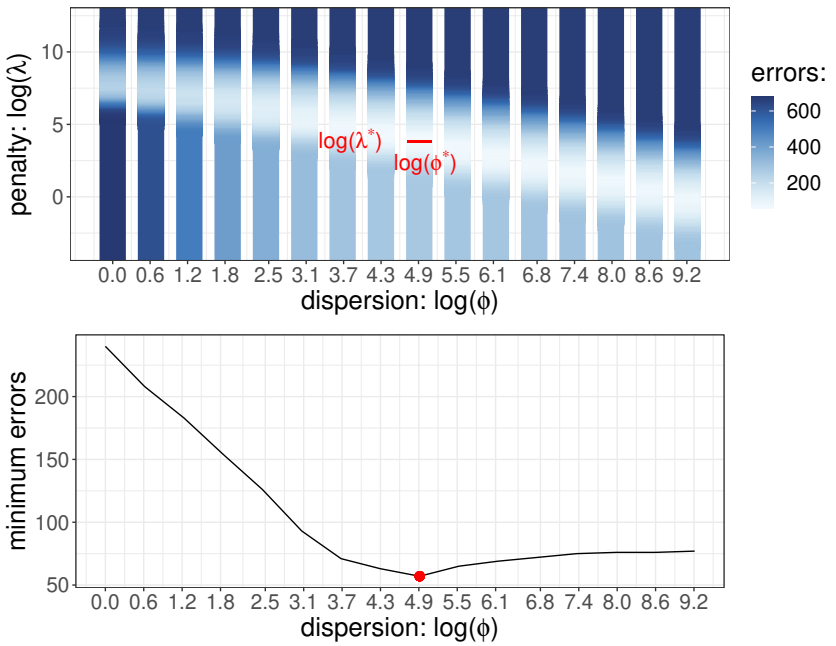


Figure 2: (**Top**) Visualization of $\sum_{m\in \text{ training set}} E_m(\phi \in \Phi, \lambda)$. The global minimum error (57), shown in red ■, is reached for $\lambda^* = 46.86$ and $\phi^* = 135.94$. (**Bottom**) For each ϕ_i, i.e 16 values evenly placed on the log scale between 1 and 10000, the minimum error of $E_m(\phi_i, \lambda)$ has been plotted. We can see the errors growing constantly at the left en right side of ϕ^* which suggests that this range of ϕ is appropriate for learning a suitable dispersion parameter value.

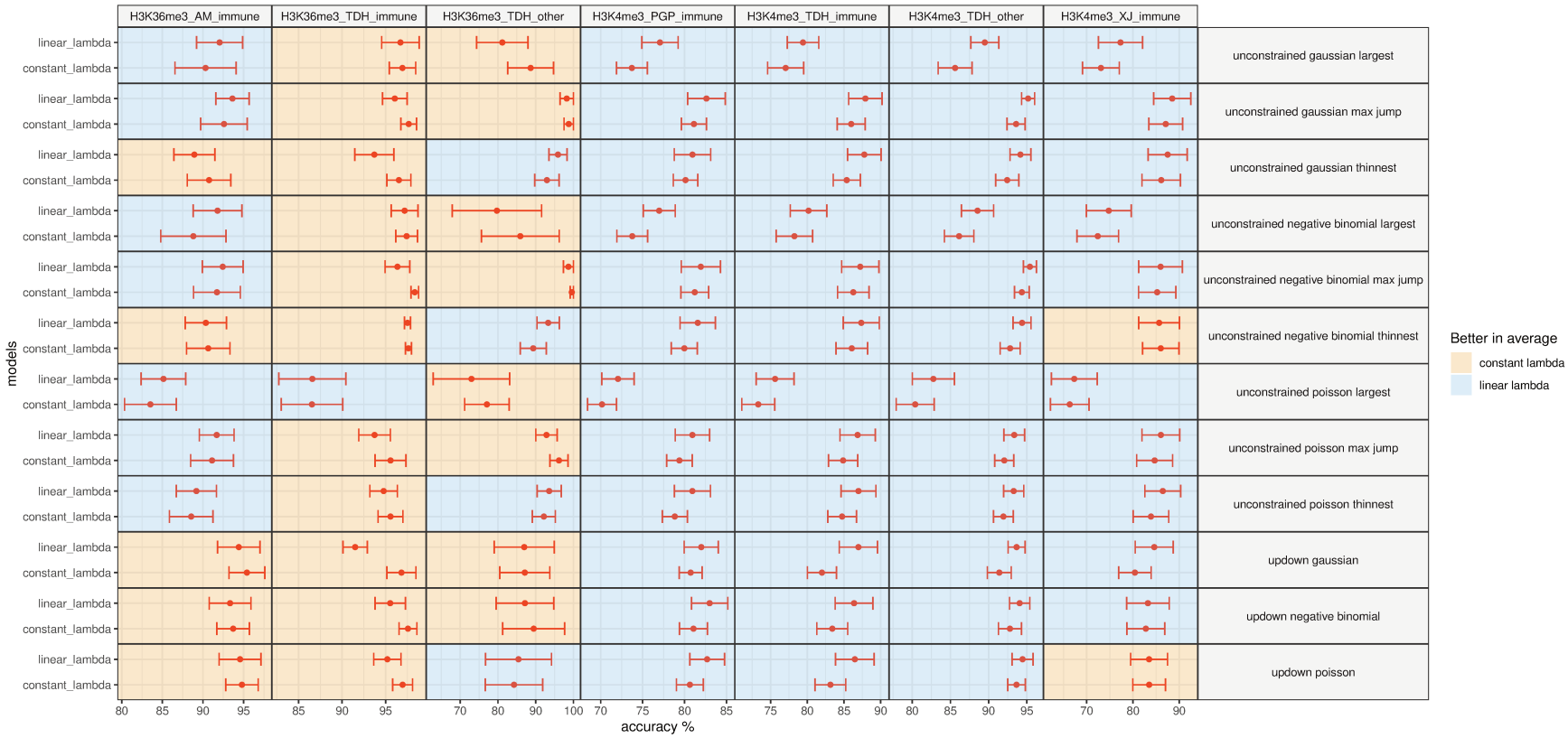


Figure 3: The *linear* λ learning method improves the accuracy of the segmentation models upon the *constant* λ learning method on H3K4me3 datasets. The mean accuracy and its 95% CI computed on the test folds is shown in red ■. In 46 of the 48 comparisons on the H3K4me3 datasets, the *linear* λ learning method was better in average than the *constant* λ learning method. After pooling the folds by type of experiment, we performed a paired t-test on each comparison. After correcting the p-values with the Benjamini & Hochberg method, 6 (/12) differences in mean accuracy were still significant (adjusted p-value < 0.05). The concerned models are: unconstrained gaussian max jump; unconstrained gaussian largest; unconstrained negative binomial largest; unconstrained poisson thinnest; updown gaussian; updown negative binomial. In H3K36me3 datasets, there is no clear trend on which learning method is the best. 24 of the 36 comparisons are in favour of the *constant* λ learning method. After pooling the folds by type of experiment and performed a paired t-test following by a correction of the p-values, none of the 12 differences in mean were significant.

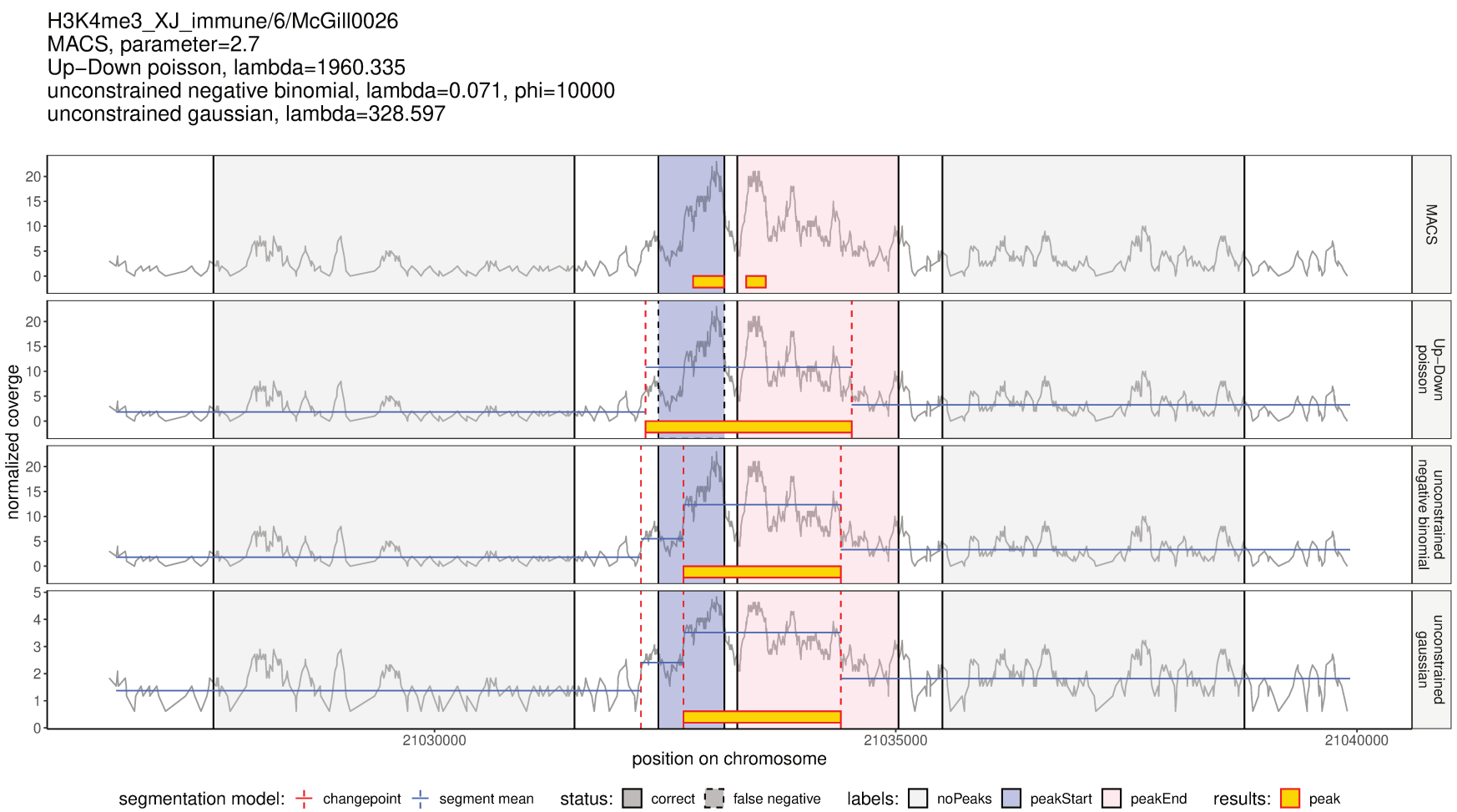


Figure 4: Visualization of the trained MACS, trained Up-Down Poisson, trained unconstrained negative binomial and trained unconstrained Gaussian models, on the normalized coverage profile (in the Gaussian case the data are additionally transformed as explained in the manuscript) from one of the biological samples (McGill0026) of the ChIP-Seq experiment directed against H3K4me3 histone marks. (**Top**) Summary of the tuning parameter values learned on the training set for each model. The parameter learned for MACS is the q-value threshold parameter.

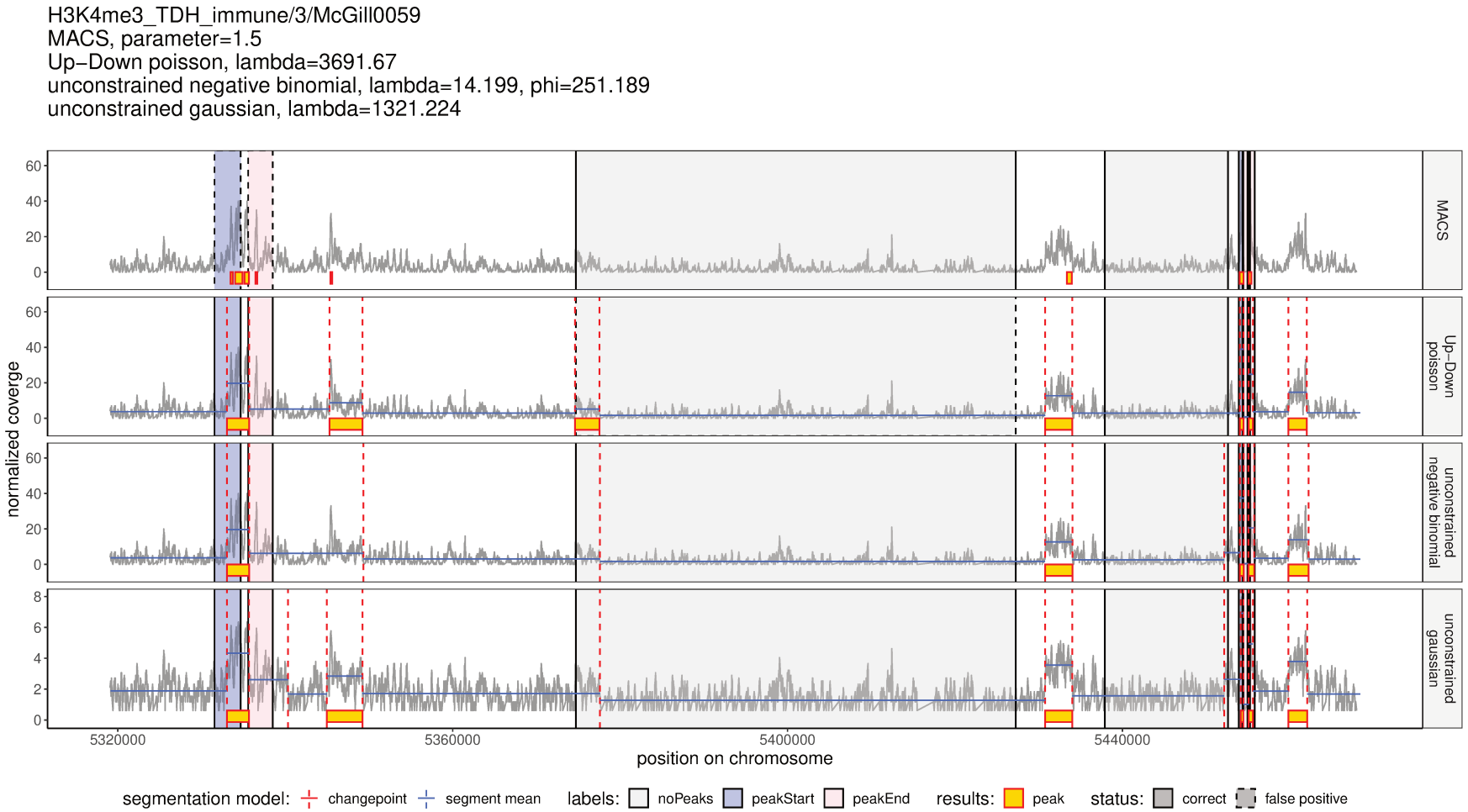


Figure 5: Visualization of the trained MACS, trained Up-Down Poisson, trained unconstrained negative binomial and trained unconstrained Gaussian models, on the normalized coverage profile (in the Gaussian case the data are additionally transformed as explained in the manuscript) from one of the biological samples (McGill0059) of the ChIP-Seq experiment directed against H3K4me3 histone marks. (**Top**) Summary of the tuning parameter values learned on the training set for each model. The parameter learned for MACS is the q-value threshold parameter.

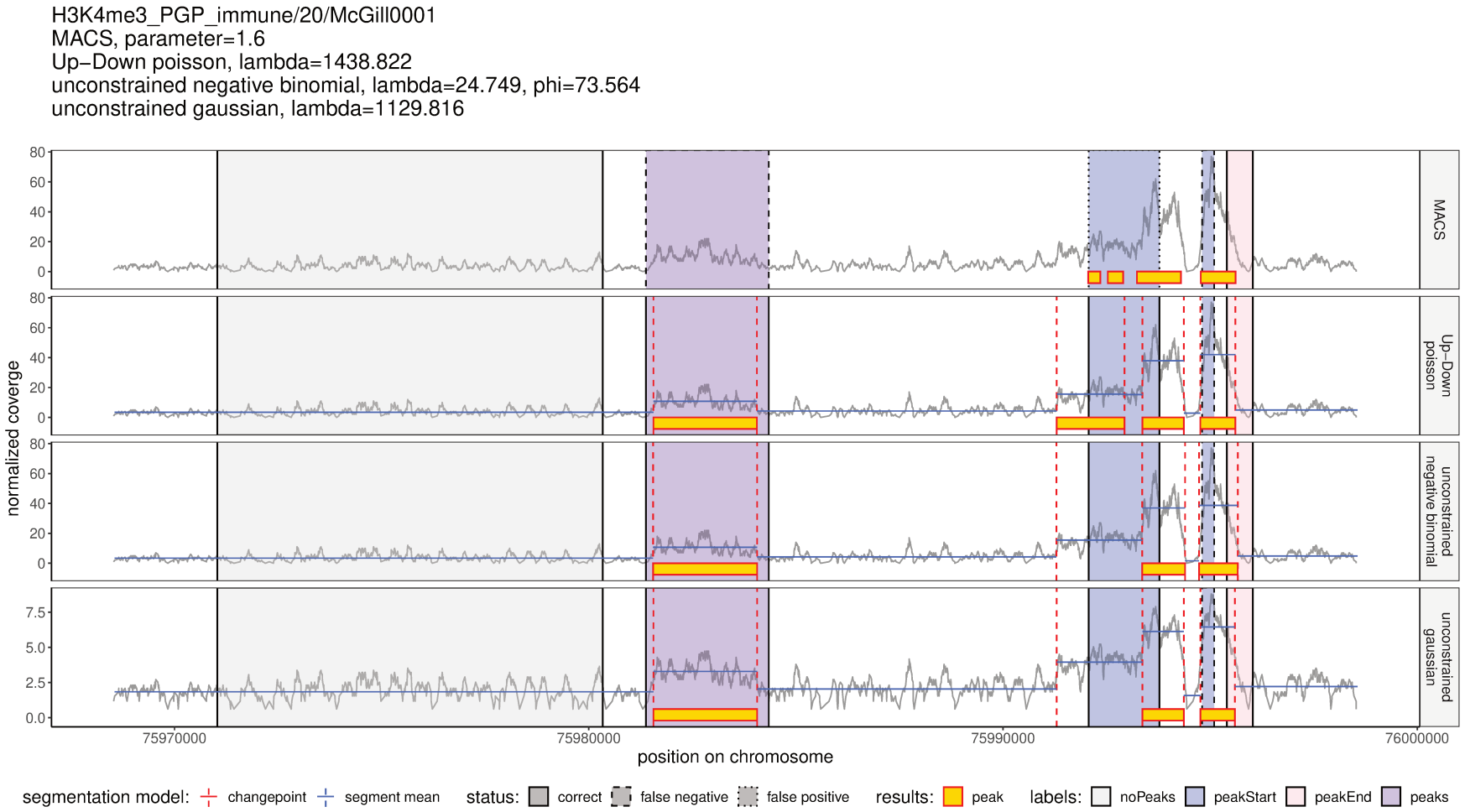


Figure 6: Visualization of the trained MACS, trained Up-Down Poisson, trained unconstrained negative binomial and trained unconstrained Gaussian models, on the normalized coverage profile (in the Gaussian case the data are additionally transformed as explained in the manuscript) from one of the biological samples (McGill0001) of the ChIP-Seq experiment directed against H3K4me3 histone marks. (**Top**) Summary of the tuning parameter values learned on the training set for each model. The parameter learned for MACS is the q-value threshold parameter.

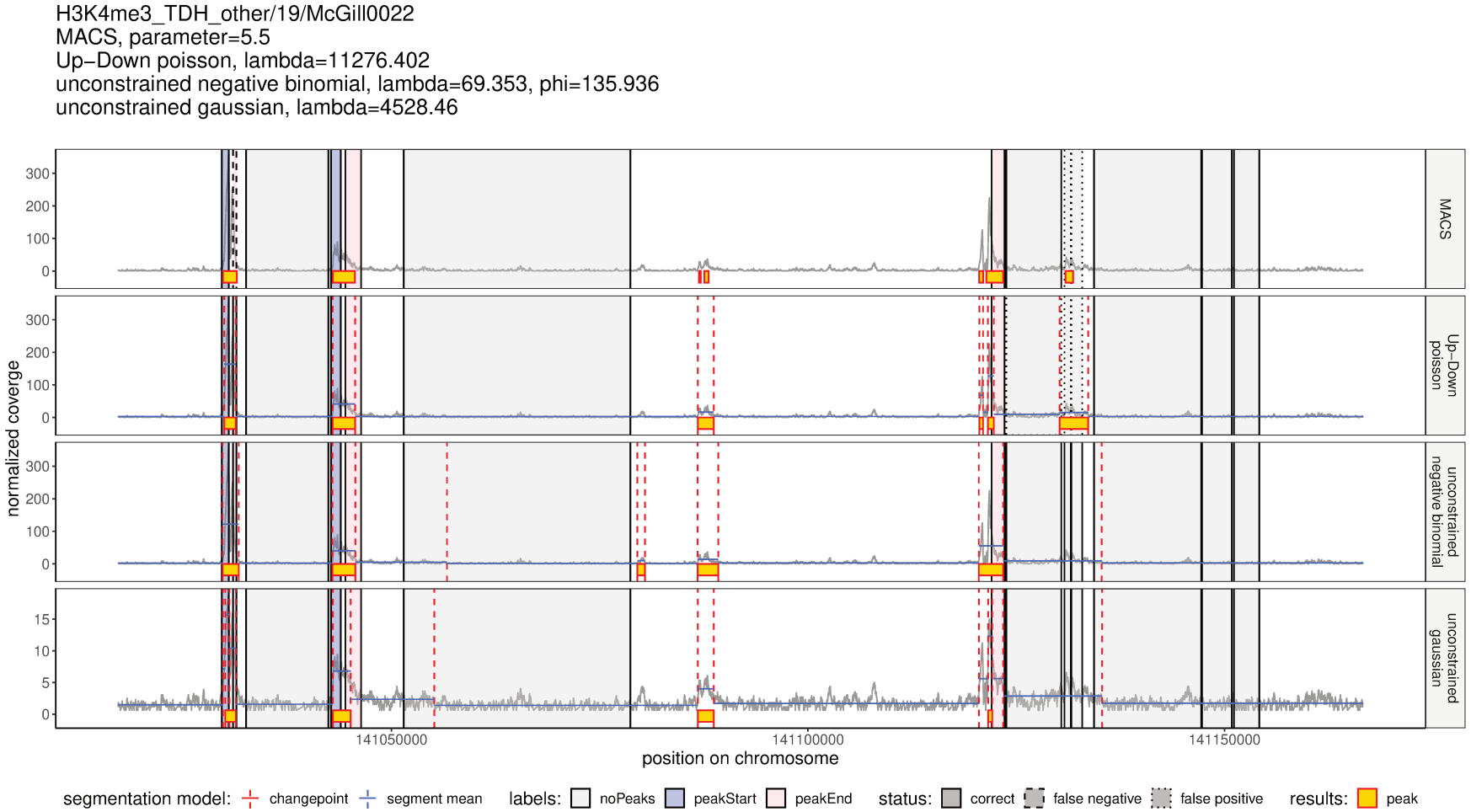


Figure 7: Visualization of the trained MACS, trained Up-Down Poisson, trained unconstrained negative binomial and trained unconstrained Gaussian models, on the normalized coverage profile (in the Gaussian case the data are additionally transformed as explained in the manuscript) from one of the biological samples (McGill0022) of the ChIP-Seq experiment directed against H3K4me3 histone marks. (**Top**) Summary of the tuning parameter values learned on the training set for each model. The parameter learned for MACS is the q-value threshold parameter.

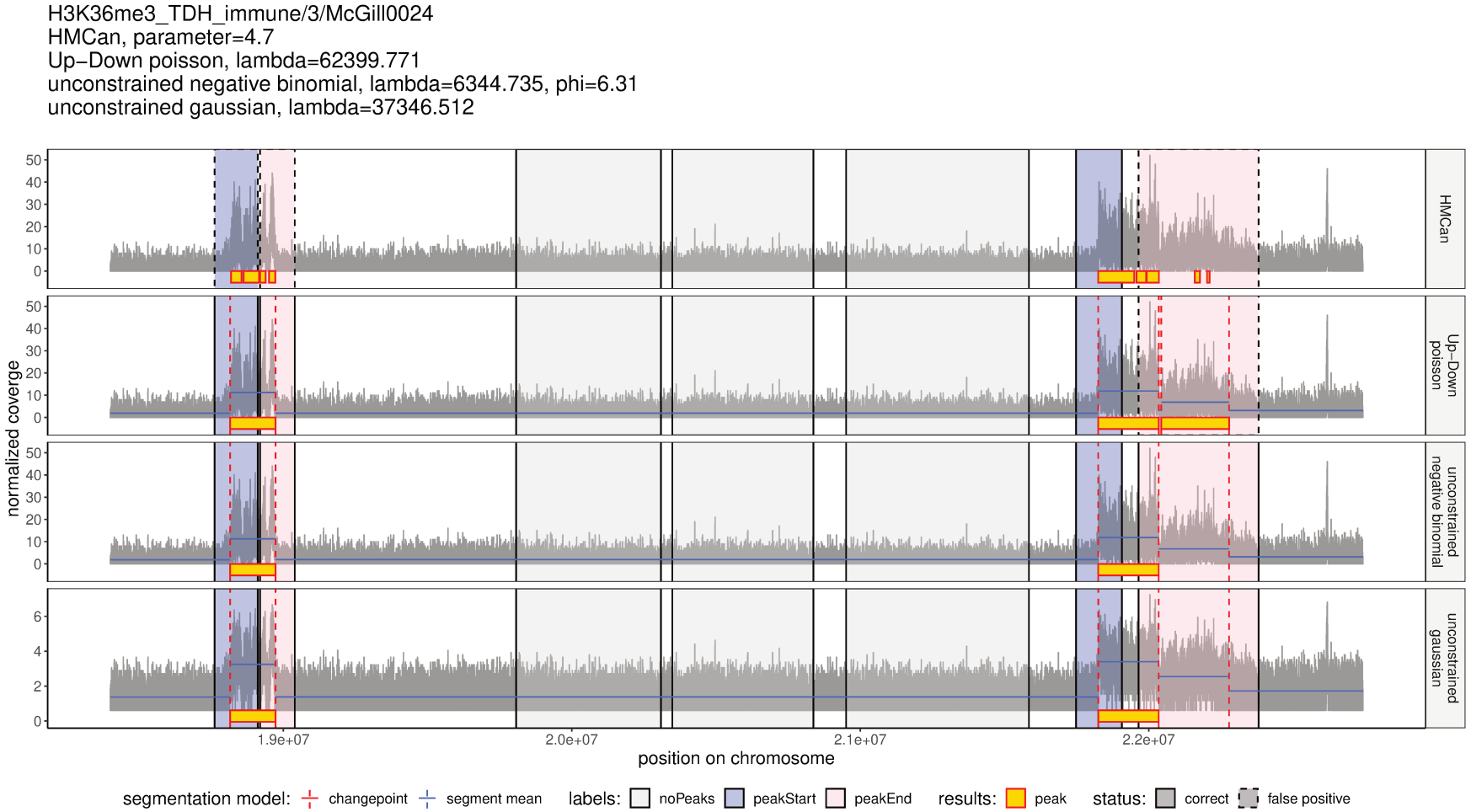


Figure 8: Visualization of the trained HMCan, trained Up-Down Poisson, trained unconstrained negative binomial and trained unconstrained Gaussian models, on the normalized coverage profile (in the Gaussian case the data are additionally transformed as explained in the manuscript) from one of the biological samples (McGill0024) of the ChIP-Seq experiment directed against H3K36me3 histone marks. (**Top**) Summary of the tuning parameter values learned on the training set for each model. The parameter learned for HMCan is the finalThreshold parameter.

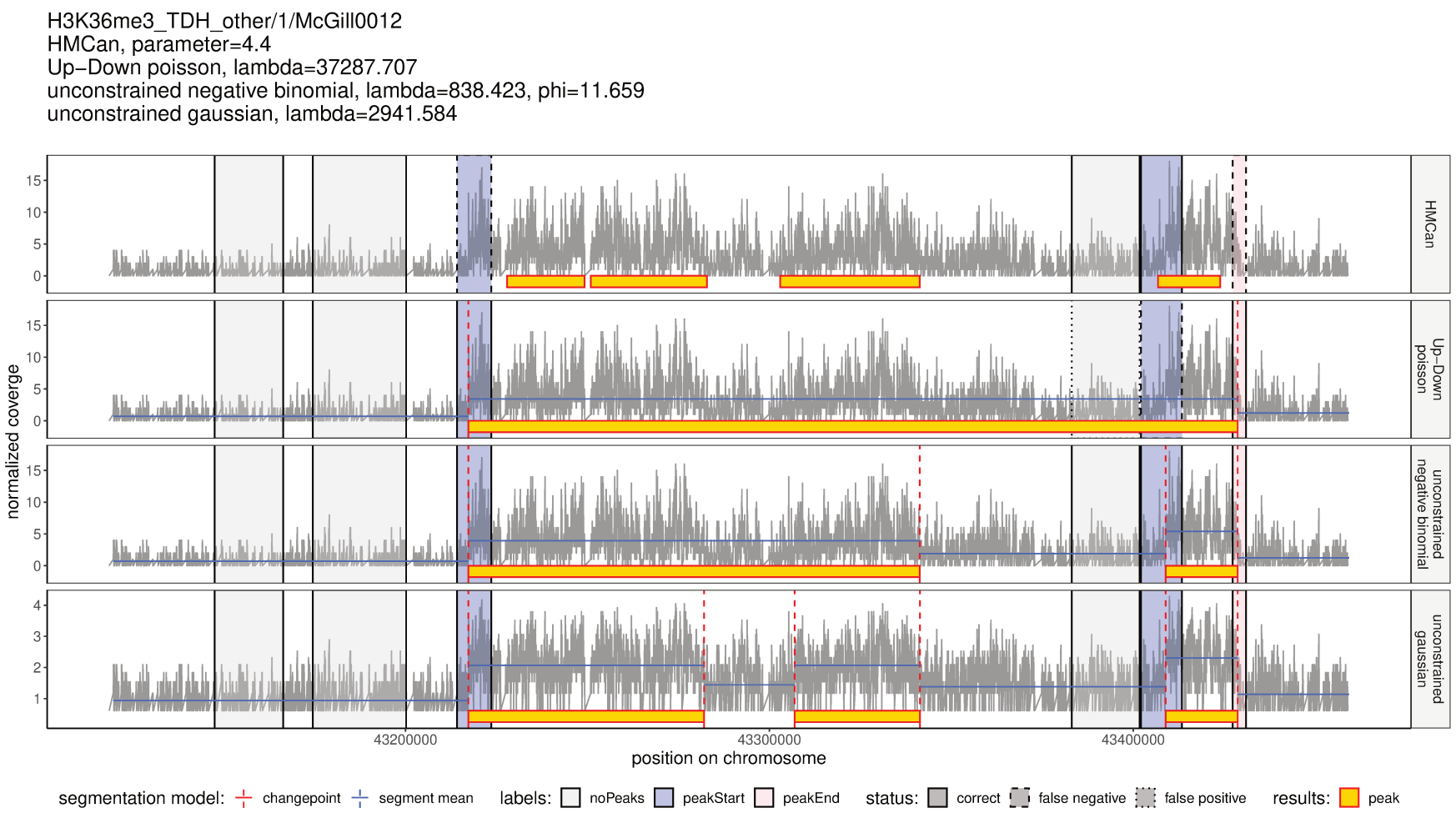


Figure 9: Visualization of the trained HMCan, trained Up-Down Poisson, trained unconstrained negative binomial and trained unconstrained Gaussian models, on the normalized coverage profile (in the Gaussian case the data are additionally transformed as explained in the manuscript) from one of the biological samples (McGill0012) of the ChIP-Seq experiment directed against H3K36me3 histone marks. (**Top**) Summary of the tuning parameter values learned on the training set for each model. The parameter learned for HMCan is the finalThreshold parameter.

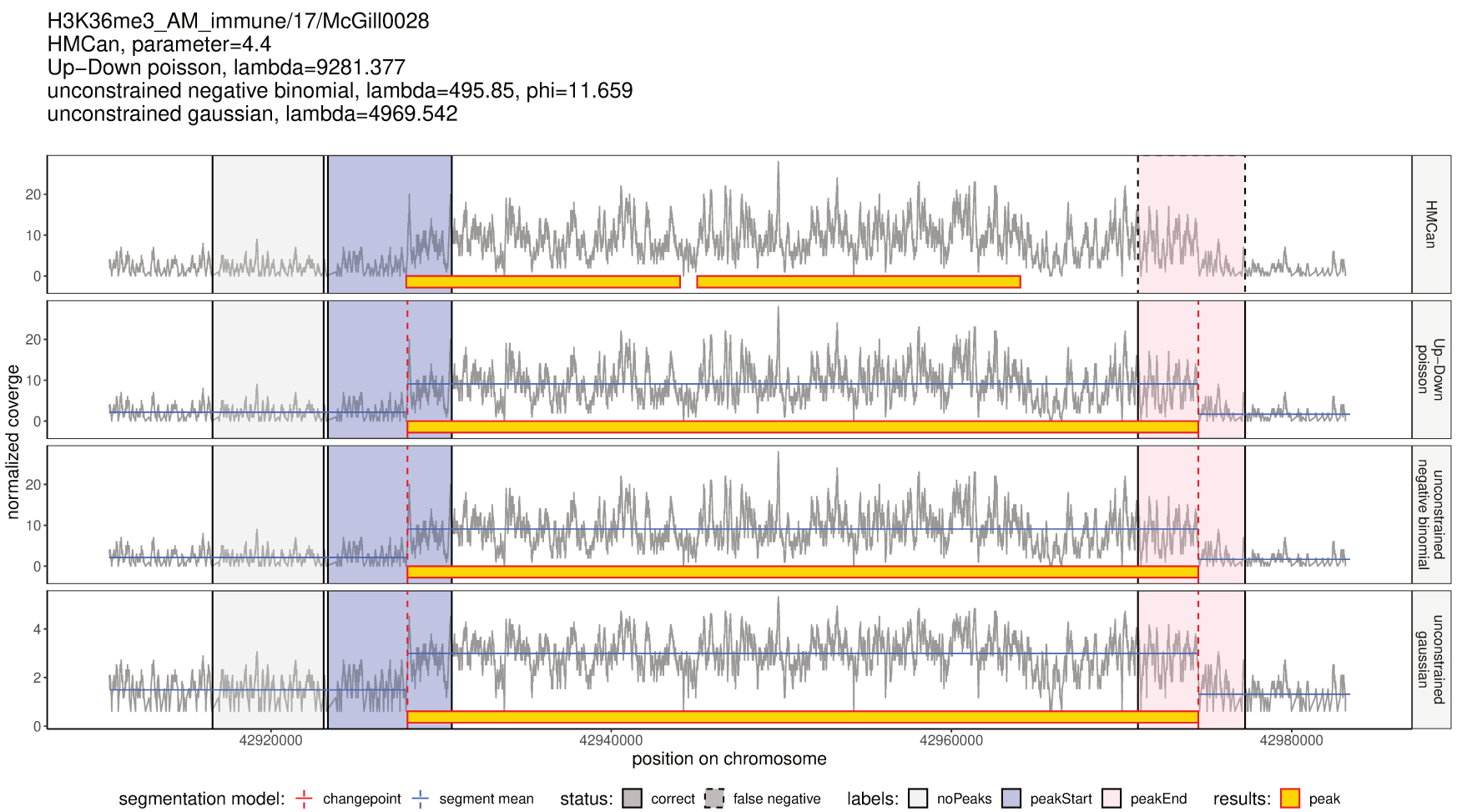


Figure 10: Visualization of the trained HMCan, trained Up-Down Poisson, trained unconstrained negative binomial and trained unconstrained Gaussian models, on the normalized coverage profile (in the Gaussian case the data are additionally transformed as explained in the manuscript) from one of the biological samples (McGill0028) of the ChIP-Seq experiment directed against H3K36me3 histone marks. (**Top**) Summary of the tuning parameter values learned on the training set for each model. The parameter learned for HMCan is the finalThreshold parameter.

Chapter B

Ms.FPOP : an exact and fast segmentation algorithm with a multiscale penalty

B.1 Ms.FPOP

This article has been submitted to the journal *Journal of Computational and Graphical Statistics* and is already available on *arXiv* (doi.org/10.48550/arXiv.2303.08723).

Ms.FPOP: an exact and fast segmentation algorithm with a multiscale penalty

Arnaud Liehrmann [*1,2]
and
Guillem Rigaill[1,2]
[1] Institute of Plant Sciences Paris-Saclay (IPS2)
[2] Laboratoire de Mathématiques et de Modélisation d'Evry (LaMME)

May 22, 2023

Abstract

Given a time series in $\mathbb{R}^n$ with a piecewise constant mean and independent noises, we propose an exact dynamic programming algorithm to minimize a least square criterion with a multiscale penalty promoting well-spread changepoints. Such a penalty has been proposed in Verzelen *et al.* (2020), and it achieves optimal rates for changepoint detection and changepoint localization.

Our proposed algorithm, named Ms.FPOP, extends functional pruning ideas of Rigaill (2015) and Maidstone *et al.* (2017) to multiscale penalties. For large signals, $n \geq 10^5$, with relatively few real changepoints, Ms.FPOP is typically quasi-linear and an order of magnitude faster than PELT. We propose an efficient C++ implementation interfaced with R of Ms.FPOP allowing to segment a profile of up to $n = 10^6$ in a matter of seconds.

Finally, we illustrate on simple simulations that for large enough profiles ($n \geq 10^4$) Ms.FPOP using the multiscale penalty of Verzelen *et al.* (2020) is typically more powerfull than FPOP using the classical BIC penalty of Yao (1989).

Keywords: changepoint detection, multiscale penalty, maximum likelihood inference, discrete optimization, dynamic programming, functional pruning

*Corresponding author : arnaud.liehrmann@universite-paris-saclay.fr

1 Introduction

A National Research Council report [Council et al., 2013] identifies changepoint detection as one of the "inferential giants" in massive data analysis. Detecting changepoints, whether a posteriori or online, is important in areas as diverse as bioinformatics [Olshen et al., 2004, Picard et al., 2005], econometrics and finance [Bai and Perron, 2003, Thies and Molnár, 2018], climate [Reeves et al., 2007], autonomous driving [Galceran et al., 2017], computer vision [Ranganathan, 2012] and neuroscience [Jewell et al., 2020]. The most common and prototypical changepoint detection problem is that of detecting changes in mean of a univariate gaussian signal :

$$y_t = f_t + \varepsilon_t, \quad \text{for } t = 1, \dots, n, \tag{1}$$

where f_t is a deterministic piecewise constant with changepoints whose number D and locations, $0 < \tau_1 < \ldots < \tau_D < n$, are unknown, and ε_t are independant and follow a Gaussian distribution of mean 0 and variance 1. A large number of approaches have been proposed to solve this problem (amongst many others [Yao, 1989, Lebarbier, 2005, Harchaoui and Lévy-Leduc, 2010, Frick et al., 2014, Fryzlewicz, 2020], see [Aminikhanghahi and Cook, 2017, Truong et al., 2020] for a review).

Recently, [Verzelen et al., 2020] characterize optimal rates for changepoint detection and changepoint localization and proposed a least-squares estimator with a multiscale penalty achieving these optimal rates. This multiscale penalty depends on minus the log-length of the segments which promotes well spread changepoints. It can be written as :

$$\sum_{d=1}^{D+1} \gamma + \beta \log(n) - \beta \log(\tau_d - \tau_{d-1}), \tag{2}$$

where $\gamma = qL$ and $\beta = 2L$ with q positive and $L > 1$, and with the convention that $\tau_0 = 0$ and $\tau_{D+1} = n$.

Up to a multiplicative constant this penalty is always smaller than the BIC penalty ($2\log(n)$) [Yao, 1989]. Intuitively, it favors balanced segmentation as:

- the penalty of a fixed sized segment (r) increases with n : $\beta \log(n/r)$.

- while the penalty for a segment whose size is proportional to n ($\alpha.n$) is constant of n : $\beta \log(1/\alpha)$.

Contribution In this paper, we propose a dynamic programming algorithm, named Ms.FPOP optimizing a slightly more general penalty. where the $\log(\tau_d - \tau_{d-1})$ is replaced by $g(\tau_d - \tau_{d-1})$ for an arbitrary function g satistying assumption A1.

Existing works Ms.FPOP extends functional pruning techniques as in PDPA or FPOP [Rigaill, 2015, Maidstone et al., 2017] to the case of multiscale penalties. A key condition for FPOP and PDPA is that the cost function is point additive (condition C1 in [Maidstone et al., 2017]). As we will explain in more details later, this condition is not verified for the multiscale penalty (2), making the extension not trivial. The key idea behind functionnal pruning is to store the set of parameter values for which a particular change is optimal. For a classical penalty (i.e. with a point additive cost function) this set gets smaller with every new datapoint. This is not the case with the multiscale penalty making the update more complex. A key insight of Ms.FPOP is to store a slightly larger set that is easy to update.

Importantly, it is possible to optimize the multiscale criteria of [Verzelen et al., 2020] using inequality based pruning as in PELT. We will call Ms.PELT this strategy. However for large signals with relatively few true changepoints it is our experience that Ms.PELT is quadratic while Ms.FPOP is quasi-linear. For example it can be seen on Figure 1.A that it takes about 193 seconds for Ms.PELT to process a signal of size $n = 128000$ without any changepoint. In the same amount of time Ms.FPOP can process signals of size larger than $n = 4 \times 10^6$. In the presence of true changepoints, (one every thousand datapoints) Ms.PELT as expected is much faster but still slower than Ms.FPOP (see Figure 1.B).

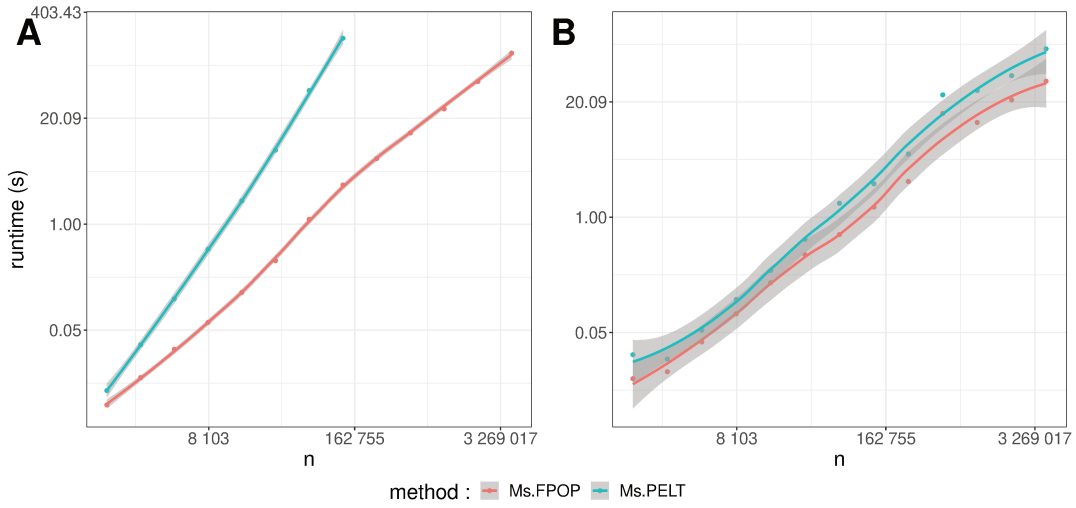


Figure 1: Runtimes of PELT and Ms.FPOP as a function of n to optimize the multiscale penalty of [Verzelen et al., 2020] with $\beta = 2.25$ and $q = 9$ on an Intel Core i7-10810U CPU @ 1.10GHzx12 computer for signals without changes (A) or signal with a change of size 1 every thousand datapoints (B).

Outline In the rest of the paper we will (1) introduce our notations, (2) review the key idea behind FPOP, (3) explain how and under which conditions we extend FPOP to multiscale penalty, (4) study the performance of Ms.FPOP relative to FPOP for various signals and (5) conclude with a discussion.

1.1 Multiple changepoint model

In this section we describe our changepoint notations and the multiscale criteria we want to optimize.

Segmentations and set of segmentations For any n in $\mathbb{N}$ we write $1:n = \{1, \cdots, n\}$. For any integer $D \geq 0$ we define a segmentation with D changes of $1:n$ as an ordered subset of $1:(n-1)$ of size D, with τ_j the location of the j^{th} change for j in $1, \ldots, D$. It will be usefull to also consider the dummy indices $\tau_0 = 0$ and $\tau_{D+1} = n$. We call $\mathcal{M}^D_{1:n}$ the set of all such segmentations in D changes and $\mathcal{M}_{1:n}$ the union of all these sets : $\bigcup_{0 \leq D \leq n-1} \mathcal{M}^D_{1:n}$. For any segmentation τ in $\mathcal{M}_{1:n}$ we note $|\tau|$ the number of segments of τ. In other words,

if τ is in $\mathcal{M}_{1:n}^D$ then $|\tau| = D + 1$. We can enumerate the elements of $\mathcal{M}_{1:n}$ and we get :

$$|\mathcal{M}_{1:n}| = \sum_{D=0}^{n-1} |\mathcal{M}_{1:n}^D| = \sum_{D=0}^{n-1} \binom{n-1}{D} = 2^{n-1}$$

Multiscale penalized likelihood Under the piecewise constant model (1) a classical method to estimate the position and the number of changes is to optimize a penalized likelihood criterion. It is common to use a penalty that is linear in the number of changepoints [Yao, 1989, Killick et al., 2012, Maidstone et al., 2017] and optimization wise the goal is to compute:

$$\begin{aligned} \boldsymbol{\tau}_n^* &= \underset{\tau \in \mathcal{M}_{1:n}}{\arg\min} \left\{ \sum_{j=1}^{|\tau|} \min_{\mu} \left(\sum_{i=\tau_{j-1}+1}^{\tau j} (y_i - \mu)^2 \right) + \alpha |\tau| \right\}, \\ F_n &= \min_{\tau \in \mathcal{M}_{1:n}} \left\{ \sum_{j=1}^{|\tau|} \min_{\mu} \left(\sum_{i=\tau_{j-1}+1}^{\tau j} (y_i - \mu)^2 \right) + \alpha |\tau| \right\}, \end{aligned} \tag{3}$$

where α is a constant to be calibrated (*e.g.* $\alpha = 2\log(n)$).

Here we consider a more general penalty that depends on the length of the segments:

$$\begin{aligned} \boldsymbol{\tau}_n^* &= \underset{\tau \in \mathcal{M}_{1:n}}{\arg\min} \left\{ \sum_{j=1}^{|\tau|} \min_{\mu} \left(\sum_{i=\tau_{j-1}+1}^{\tau j} (y_i - \mu)^2 - \beta g(\tau_j - \tau_{j-1}) \right) + \alpha |\tau| \right\}, \\ F_n &= \min_{\tau \in \mathcal{M}_{1:n}} \left\{ \sum_{j=1}^{|\tau|} \min_{\mu} \left(\sum_{i=\tau_{j-1}+1}^{\tau j} (y_i - \mu)^2 - \beta g(\tau_j - \tau_{j-1}) \right) + \alpha |\tau| \right\}, \end{aligned} \tag{4}$$

where g is a function satistfying assumption A1 described in the next paragraph, and α and β are constants to be calibrated. We recover the multiscale criteria proposed in [Verzelen et al., 2020] taking $g = \log$, $\alpha = \gamma + \beta g(n)$, and γ a constant that remains to be chosen. We recover the classical penalty of [Yao, 1989] taking $g = 0$, $\alpha = 2\log(n)$.

Assumption 1. *$h(t, s, s') = g(t - s') - g(t - s)$ is a non-decreasing function in t and $\lim_{t\to\infty} h(t, s, s') = 0$, therefore $h(t, s, s') \leq 0$.*

This assumption will be useful later to bound the difference between the cost of two changes s and s'. Intuitively, assumption A1 states that g favors older changes but that asymptotically (large enough t relative to s and s') this advantage for older changes vanishes. Importantly, this assumption is true for the multiscale penalty proposed in [Verzelen et al., 2020] as $\beta > 0$ and $g(t-s') - g(t-s) = \log(1-(s'-s)/(t-s))$ is increasing with t.

1.2 Optimization with dynamic programming

In this section we explain how one can optimize equation (4) using dynamic programming ideas with (i) inequality based pruning and (ii) functional pruning.

Dynamic programming with inequality based pruning The penalised cost of a segmentation τ inside the arg min of equation (4) can be written as a sum over all segments of τ :

$$\sum_{j=1}^{|\tau|} \min_{\mu} \left(\sum_{i=\tau_{j-1}+1}^{\tau j} (y_i - \mu)^2 - \beta g(\tau_j - \tau_{j-1}) + \alpha \right),$$

therefore the optimisation can be done iteratively using the Optimal Partionning (OP) algorithm proposed in [Jackson et al., 2005] using dynamical programming ideas developped in [Auger and Lawrence, 1989] and [Bellman, 1961]. It is possible to speed up calculations using the PELT algorithm [Killick et al., 2012] because equation (4) of [Killick et al., 2012] is true at least for constant $K = -\beta(\max_{1\leq\ell\leq n}\{g(\ell)\} - 2\min_{1\leq\ell\leq n}\{g(\ell)\})$ (see Appendix A). If g is concave (such as in the penalty (2) proposed in [Verzelen et al., 2020]), K can be chosen much closer to zero : $K = -\beta(g(2) - 2g(1))$ (see Appendix A), or adaptively to the last segment length ℓ : $K_\ell = -\beta(g(\ell) + g(1) - g(\ell+1))$ (see Appendix B). Our implementation of PELT optimizing (4) with $g = \log$ and $K_\ell = -\beta\log(\frac{1}{\ell}+1)$ is called Ms.PELT. Note that $K_\ell \leq -\beta\log(2)$.

As shown in the Figure 1, if the number of real changepoints is not linear in n, for $g = \log$, and a positive β, Ms.PELT is quadratic. This makes the analysis of large profiles with 10^5 or 10^6 datapoints long and unpractical (*e.g.* more than 100 seconds for a profile with 10^5 datapoints and one changepoint, more than 1 hour for a profile with 10^6 datapoints and one changepoint).

Dynamic programming with functional pruning In the rest of the paper, we present a functional pruning algorithm (called Ms.FPOP), in the sense of the PDPA [Rigaill, 2015] or FPOP [Maidstone et al., 2017], to solve (4). Ms.FPOP optimizes (4) in a matter of seconds even for $n = 10^6$. As the cost of equation (4) is not point-additive, condition C1 of [Maidstone et al., 2017] is not true, and maintaining the set of means for which a change is optimal is more complex. Our key idea is to maintain a slightly larger set that is easier to update.

2 Functional pruning

2.1 Functional pruning optimal portioning (FPOP)

To better explain Ms.FPOP, we first review some of the key elements of FPOP to optimize equation (3). FPOP introduces for every change s its best cost as function of the last parameter μ at time t, $\widetilde{f}_{t,s}(\mu)$. Formally this is:

$$\widetilde{f}_{t,s}(\mu) = F_s + \sum_{i=s+1}^{t} (y_i - \mu)^2 + \alpha, \quad \text{with} \quad \widetilde{f}_{t,t}(\mu) = F_t + \alpha \quad \text{and} \quad F_0 = -\alpha. \tag{5}$$

$\widetilde{f}_{t,s}(\mu)$ is a second degree polynomial in μ defined by three coefficients : $a_2\mu^2 + a_1\mu + a_0$ with $a_2 = t - s$, $a_1 = -2\sum_{i=s+1}^{t} y_i$ and $a_0 = F_s + \alpha + \sum_{i=s+1}^{t} y_i^2$. The update of these coefficients is straightforward using the following formula:

$$\widetilde{f}_{t,s}(\mu) = \widetilde{f}_{t-1,s}(\mu) + (y_t - \mu)^2. \tag{6}$$

At each time step t, FPOP updates the minimum of all $\widetilde{f}_{t,s}(\mu)$, denoted $\widetilde{F}_t(\mu) = \min_{s \leq t} \left\{ \widetilde{f}_{t,s}(\mu) \right\}$. The key idea behind FPOP is that to compute and update $\widetilde{F}_t(\mu)$ one only need to consider changes s with a none empty "living-set" : $\mathcal{F}_t = \{s \leq t | Z^*_{t,s} \neq \emptyset\}$ where the "living-set" of change s is $Z^*_{t,s} = \{\mu | \widetilde{f}_{t,s}(\mu) = \widetilde{F}_t(\mu)\}$. Given those definitions we have $\widetilde{F}_t(\mu) = \min_{s \in \mathcal{F}_t} \left\{ \widetilde{f}_{t,s}(\mu) \right\}$. In other words, s is pruned as soon as its "living-set" is empty, which is justified because

$$Z^*_{t,s} \supset Z^*_{t+1,s} \quad \text{and} \quad Z^*_{t,s} = \emptyset \implies Z^*_{t+1,s} = \emptyset \ . \tag{7}$$

Note that we can then retrieve F_t by minimizing $\widetilde{F}_t(\mu)$ on μ.

2.2 Ms.FPOP: functional pruning for a multiscale penalty

Ms.FPOP optimizes equation (4). As for FPOP we introduce for every change s its best cost as a function of the last parameter μ at time t, $\widetilde{f}_{t,s}(\mu)$. Formally this is :

$$\widetilde{f}_{t,s}(\mu) = F_s + \sum_{i=s+1}^{t} (y_i - \mu)^2 + \alpha - \beta g\left(t - s\right), \tag{8}$$

with $\widetilde{f}_{t,t}(\mu) = F_t + \alpha$ and $F_0 = -\alpha$. As in FPOP, $\widetilde{f}_{t,s}(\mu)$ can be stored as a second degree polynomial in μ. The update is also straightforward using the following formula:

$$\widetilde{f}_{t,s}(\mu) = \widetilde{f}_{t-1,s}(\mu) + (y_t - \mu)^2 + \beta g\left(t - 1 - s\right) - \beta g\left(t - s\right) \tag{9}$$

Analogously to FPOP we can calculate F_t by minimizing $\widetilde{f}_{t,s}$ both on μ and s. The main difference with FPOP is that the rule (7) is no longer true for Ms.FPOP because $\widetilde{f}_{t,s}(\mu) - \widetilde{f}_{t,s'}(\mu)$ depends on t:

$$\widetilde{f}_{t,s}(\mu) - \widetilde{f}_{t,s'}(\mu) = F_s - F_{s'} + \sum_{i=s+1}^{s'} (y_i - \mu)^2 + \beta(\underbrace{g(t - s^{'}) - g(t - s))}_{\substack{\textit{a function varying} \\ \textit{with t, s et s'}}}). \tag{10}$$

Because of that, in the course of the algorithm we need to re-evaluate the set on which the candidate change s is better than s' at various t, $I_{t,s,s'}$ with $s < s^{'}$:

$$I_{t,s,s'} = \{\mu \mid \widetilde{f}_{t,s}(\mu) \ \leq \ \widetilde{f}_{t,s'}(\mu)\}. \tag{11}$$

For arbitrary functions g the set $I_{t,s,s'}$ may vary drastically from one t to the next. Using assumption A1 we can control those variations.

2.2.1 Update of the candidate changes living set ($Z_{t,s}$)

Rather than evaluating the exact living set $Z^*_{t,s}$ of all changes, we are seeking to update a slightly larger set, $Z_{t,s}$, including $Z^*_{t,s}$ and such that if $Z_{t,s}$ is empty we can guarantee that $Z^*_{t+h,s}$ is also empty for all $h > 0$. The possibility of defining such a $Z_{t,s}$ depends on the property of the function g.

Assume A1 we propose to update $Z_{t+1,s}$ as follow:

$$Z_{t+1,s} = Z_{t,s} \cap \overbrace{\Big(\bigcap_{s' \in \mathcal{A}_{t,s}} I_{t+1,s,s'}\Big)}^{comparison\ with\ future\ changes} \setminus \overbrace{\Big(\bigcup_{s'' \in \mathcal{B}_s} I_{\infty,s'',s}\Big)}^{comparison\ with\ past\ changes} , \tag{12}$$

where $\mathcal{A}_{t,s}$ is any subset of $\{s+1, ..., t\}$, $\mathcal{B}_s$ is any subset of $\{1, ..., s-1\}$, and $I_{\infty,s,s'}$ correspond to $I_{t,s,s'}$ when $t \to \infty$ (which is properly define under assumption A1).

Pruning Based on update (12) it should be clear that if $Z_{t,s}$ is empty so are all $Z_{t+h,s}$, for $h > 0$. In the next lemma we show that $Z_{t,s}$ includes $Z^*_{t,s}$. Therefore we further have that if $Z_{t,s}$ is empty so are all $Z^*_{t+h,s}$, and change s can be pruned.

Lemma 1. *Taking* $Z_{s,s} =]min_i y_i, \max_i y_i[$, *updating* $Z_{t+1,s}$ *using equation* (12) *and assuming A1 we have*

$$Z^*_{t,s} \subset Z_{t,s} , \tag{13}$$

and for an integer $h > 0$

$$Z^*_{t+h,s} \subset Z_{t+1,s} . \tag{14}$$

Proof. For any t, we will prove by induction that for any t' in $\{s, \cdots, t\}$ we have $Z^*_{t,s} \subset Z_{t',s}$.

For $t' = s$ and for any t larger or equal to s we have (by definition of $Z_{s,s}$) that $Z^*_{t,s} \subset Z_{t',s} = Z_{s,s}$.

Now assume that for $t' < t$ we have $Z^*_{t,s} \subset Z_{t',s}$. As h is non-decreasing for any $t'+1 \leq t$ we have the following two inclusions :

$$I_{t,s,s'} \subset I_{t'+1,s,s'}. \tag{15}$$

$$I_{\infty,s,s'} \subset I_{t'+1,s,s'} \tag{16}$$

Therefore for $t' < t$ we have

$$
\begin{aligned}
Z^*_{t,s} &= \Big(\bigcap_{s<s'\leq t} I_{t,s,s'}\Big)\backslash\Big(\bigcup_{s''<s} I_{t,s'',s}\Big) && \text{by definition of } Z^*_{t,s}\\
Z^*_{t,s} &\subset Z_{t',s} \cap \Big(\bigcap_{s<s'\leq t} I_{t,s,s'}\Big)\backslash\Big(\bigcup_{s''<s} I_{t,s'',s}\Big) && \text{by induction}\\
&\subset Z_{t',s} \cap \Big(\bigcap_{s<s'\leq t} I_{t'+1,s,s'}\Big)\backslash\Big(\bigcup_{s''<s} I_{\infty,s'',s}\Big) && \text{using equation (15) and (16)}\\
&\subset Z_{t',s} \cap \Big(\bigcap_{s'\in \mathcal{A}_{t',s}} I_{t'+1,s,s'}\Big)\backslash\Big(\bigcup_{s''\in \mathcal{B}_s} I_{\infty,s'',s}\Big) && \text{by definition of } \mathcal{A}_{t',s} \text{ and } \mathcal{B}_s.
\end{aligned}
$$

Using equation (12) we thus get that $Z^*_{t,s} \subset Z_{t'+1,s}$, proving the induction.

To recover equation (14) we notice from update (12) that $Z_{t+1,s} \subset Z_{t,s}$ and apply equation (13). □

2.2.2 Ms.FPOP algorithm, choice of $\mathcal{A}_{t,s}$ and $\mathcal{B}_s$

The update rule (12) suggest that for each candidate change s we should compare it future change s' in $\mathcal{A}_{t,s}$, and past change s'' in $\mathcal{B}_s$. For past candidate changes $s^{''}$ this comparison can be done once and for all considering that t goes to infinity ($I_{\infty,s'',s}$). For future candidate changes $s^{'}$, on the contrary, it might be usefull to update the interval $I_{t,s,s'}$. Performing at each time step, for each s, a comparison with all s' is time consuming. Intuitively, the complexity of each time step is in $\mathcal{O}(\text{number of candidate changes}^2)$. Ideally, for each s, one would like to make the minimum number of comparisons that would result in its pruning. In the Algorithm 1 we consider a generic *sampling* function of s' that returns $\mathcal{A}_{t,s}$ (see the Sampling Strategies paragraph in section 3).

Algorithm 1: Ms.FPOP

Input: $Y = (y_1, ..., y_n)$, α, β, $g = log(.)$

Output: set of last best changes cp_n

$n \leftarrow |Y|$;

$F_0 \leftarrow -\alpha$;

$cp_0 \leftarrow \emptyset$;

$R_1 \leftarrow \{0\}$;

$D \leftarrow [min(Y),\ max(Y)]$;

$Z_{0,0} \leftarrow D$;

$\widetilde{f}_{0,0} \leftarrow F_0 + \alpha\ (= 0)$;

for $t \leftarrow 1, ..., n$ **do**

  for $s \in R_t$ **do**

    $\widetilde{f}_{t,s}(\mu) \leftarrow \widetilde{f}_{t,s}(\mu) + (y_t - \mu)^2 + \beta \times g(t-1-s) - \beta \times g(t-s)$;

  end

  $F_t \leftarrow \min_{s \in R_t}(\min_{\mu \in Z_{t,s}}(\widetilde{f}_{t,s}(\mu)))$;

  $s_t \leftarrow \arg\min_{s \in R_t}(\min_{\mu \in Z_{t,s}}(\widetilde{f}_{t,s}(\mu)))$;

  $cp_t \leftarrow (cp_{s_t}, s_t)$;

  $\widetilde{f}_{t,t} \leftarrow F_t + \alpha$;

  $Z_{t,t} \leftarrow D$;

  for $s \in R_t$ **do**

    $Z_{t,t} \leftarrow Z_{t,t} \setminus I_{\infty,s,t}$;

    $\mathcal{A}_{t,s} \leftarrow sample(\{s^{'} \in \{R_t \cup \{t\}\} : s^{'} > s\})$;

    $Z_{t,s} \leftarrow Z_{t,s} \cap (\bigcap_{s' \in\ \mathcal{A}_{t,s}} I_{t,s,s'})$;

  end

  $R_{t+1} \leftarrow \{s \in \{R_t \cup \{t\}\} : Z_{t,s} \neq \emptyset\}$;

end

3 Rcpp implementation of Ms.FPOP algorithm

Ms.FPOP R package The dynamic programming and functional pruning procedures describe in the algorithm 1 are implemented in C++. The input and output operations are interfaced with the R programming language thanks to the Rcpp R package. The main function `MsFPOP()` takes as input the sequence of obervations, a vector of weights for these obervations, the parameters β and α of the multiscale penalty. The function returns the

set of optimal changepoints in the sense of (4). Analogously, we implemented a version of the PELT algorithm, `MsPELT()`, that optimizes (4).

Sampling strategies To recover $\mathcal{A}_{t,s}$ we consider either an exhaustive sampling of all future changes $s' > s$ in R_t or a uniform random subsampling of them without replacement. The main function parameter `size` can be set by the user to specify for each s the number of sampled s'. In the appendix we compare the runtime of different sampling strategies (see Appendix D).

4 Simultation study

4.1 Calibration of constants γ and β from the multiscale penalty

Paper [Verzelen et al., 2020] does not recommend values for γ and β in their penalty (2). As explained in detail below, we calibrated those values to control the percentage of falsely detecting at least one change in profiles simulated without any actual change.

No change simulation We repeatedly simulate *iid* Gaussian signals of mean 0, variance 1 and varying lengths n ($n \in \{10^2, 10^3, 10^4, 10^5, 2.5 \times 10^5\}$). On these profiles we run Ms.FPOP for different γ values (ten γ values evenly spaced on the interval $[1, 20]$) and different β values ($\beta \in \{2, 2.25, 2.5, 2.75, 3\}$).

Percentage of false detection We denote $R_{>0}$ as the proportion of replicates for which Ms.FPOP returns at least one changepoint. These changepoints are false positives. Our goal is to find a combination of β and γ such that

$$R_{>0} < 0.05 \text{ (significance level)} \quad . \tag{17}$$

Empirical results In Figure 2 we observe that, by setting $\beta = 2.25$, a conservative range of γ satisfying inequality (17) can be reached for $\gamma \in [7.5, 10]$. Note that this interval satisfy inequality (17) for all tested n and β (see Appendix C).

Based on these results, in the following simulations we set $\gamma = 9$ and $\beta = 2.25$[1] for all methods optimizing (4) (Ms.FPOP, Ms.PELT). We set $\alpha = 2\sigma^2 \log(n)$ for all methods optimizing (3) (FPOP, PELT).

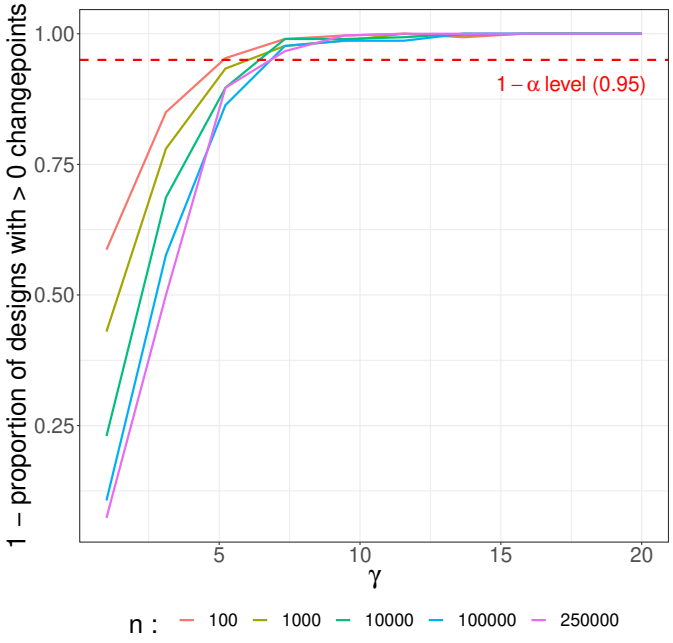


Figure 2: **Proportion of stationary Gaussian signal replicates on which Ms.FPOP returns at least one changepoint ($R_{>0}$).** $R_{>0}$ is computed for a series of γ and profile lengths (see *Design of Simulations)*. In these simulations we set $\beta = 2.25$. Results for other β values are availables in Appendix C.

4.2 Evalutation of Ms.FPOP: speed benchmark

Design of simulations We repeatedly simulate *iid* Gaussian signals with 10^5 datapoints. The profiles are affected by one or more changepoints in their mean ($D \in \{1, 5, 10, 15, 20, 25, 30, 45, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850, 900, 950, 1000\}$). The mean of segments alternates between 0 and 1, starting with 0. The variance of each segment is fixed at 1. On these profiles we run two methods optimizing the penalized likelihood defines in (3): PELT [Killick et al., 2012] and FPOP

[1]This is equivalent to setting $L = 1.125$ and $q = 8$ in equations (31) and (32) of [Verzelen et al., 2020]

[Maidstone et al., 2017], as well as methods optimizing the multiscale penalized likelihood defines in (4): Ms.PELT and Ms.FPOP. For Ms.FPOP, after comparisons with other sampling strategies (see Appendix D), we choose to randomly sample one future candidate change.

Metric For each replicate we time in seconds the compared methods.

Empirical results In Figure 3 we firstly observe that for both criteria (multiscale penalized likelihood and penalized likelihood), functional pruning methods are always faster than inequality based pruning ones. Indeed, Ms.FPOP and FPOP are always faster than Ms.PELT and PELT, respectively. The smaller D, the larger the time difference between functional pruning methods and inequality based pruning ones. For $D = 1$, Ms.FPOP runs in 2.4 seconds in average and is about 50 times faster than Ms.PELT (121.3 seconds in average). For $D = 1000$, Ms.FPOP runs in 0.7 second in average and is about 1.3 times faster than Ms.PELT (0.9 second in average). Marginally to D, FPOP runs always under 0.05 seconds. Similar trends can be observed on *iid* Gaussian signals with 10^6 datapoints (see Appendix D).

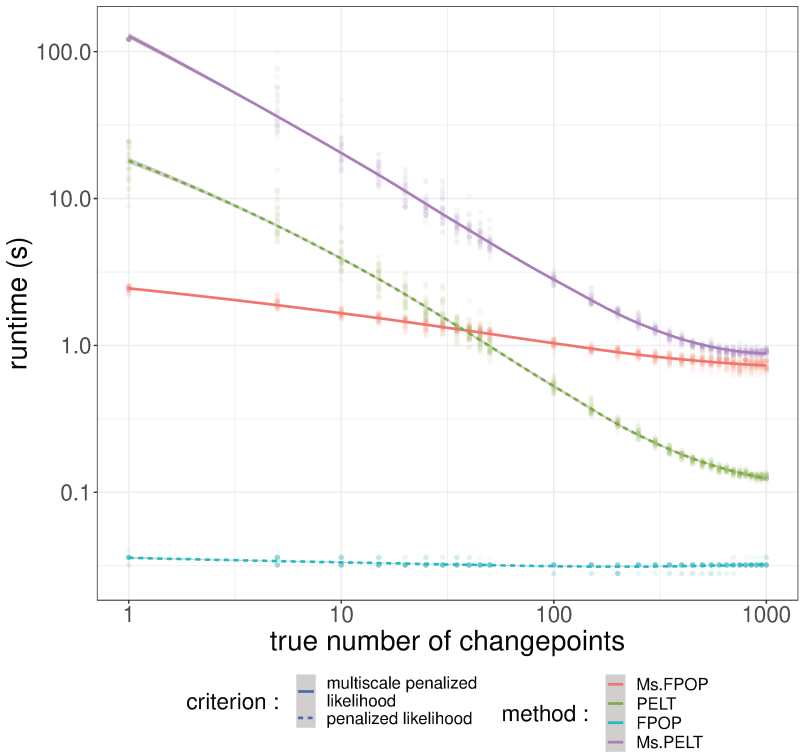


Figure 3: **Runtimes as a function of the true number of changepoints.** We timed PELT, Ms.PELT, FPOP, Ms.FPOP on profiles of length $n = 10^5$ with varying number of true changepoints D (see *Design of Simulations*) on an Intel Core i7-10810U CPU @ 1.10GHzx12 computer. The comparison between sampling strategies of future candidate changes implemented in Ms.FPOP and the comparison of PELT, FPOP, Ms.FPOP on profiles of length $n = 10^6$ are availables in Appendix D.

4.3 Evalutation of Ms.FPOP relative to FPOP: accuracy benchmarks

In this section we seek to illustrate using minimalist simulations the performances of the multiscale criteria proposed in [Verzelen et al., 2020] and implemented in Ms.FPOP relative to the BIC criteria proposed in [Yao, 1989] and implemented in FPOP.

4.3.1 Hat simulations

Design of simulations We repeatedly simulate *iid* Gaussian signals of varying size $n \in \{10^3, 10^4, 10^5\}$. Each signal is affected by two changepoints. The second changepoint (τ_2) is fixed at position $\lfloor \frac{2n}{3} \rfloor$ while we vary the position of the first changepoint (τ_1) (see Figure 4.A). τ_1 takes a series of 30 positive integers evenly spaced on the log scale on the interval $[1, \lfloor \frac{n}{3} \rfloor]$. We also look at the symmetry of this series builds around $\lfloor \frac{n}{3} \rfloor$ (i.e. $\lfloor \frac{2n}{3} \rfloor - \tau_1$, see dotted lines in Appendix E). Note that for $\tau_1 = \lfloor \frac{n}{3} \rfloor$ the segmentation is balanced. The means of the three resulting segments are set to $\mu_1 = 0$, $\mu_2 = \sqrt{\frac{100}{n}}$ and $\mu_3 = 0$. We run both Ms.FPOP and FPOP on these profiles. Ms.FPOP incorporates a multiscale penalty, while FPOP assigns equal weight to all segment sizes and serves as a reference point for comparison with Ms.FPOP. We anticipate that the multiscale penalty in Ms.FPOP will lead to more accurate segmentations of profiles with well-spread changepoints compared to FPOP. Additionally, as the size of the data (n) increases, we expect Ms.FPOP to get similar performance or outperform FPOP in terms of accuracy for all segment sizes.

Metric We denote R_2 the proportion of replicates for which a method returns exactly two changepoints. We also denote Δ_{R_2}, the $\log_2$-ratio between R_2 of Ms.FPOP and FPOP.

Empirical results In Figure 4.B and Appendix E we observe that with both Ms.FPOP and FPOP, R_2 increases when τ_1 tends towards $\lfloor \frac{n}{3} \rfloor$ (balanced segmentation). Note that the maximum is reached before $\tau_1 = \lfloor \frac{n}{3} \rfloor$.

Furthermore, in agreement with our expectations, in Figure 4.B we observe that Δ_{R_2} increases when τ_1 tends towards $\lfloor \frac{n}{3} \rfloor$. When n increases, the differences observed on small segments in favor of FPOP ($\Delta_{R_2} < 0$) disappear ($\Delta_{R_2} \to 0$) and the differences on other segments in favor of Ms.FPOP ($\Delta_{R_2} > 0$) are accentuated.

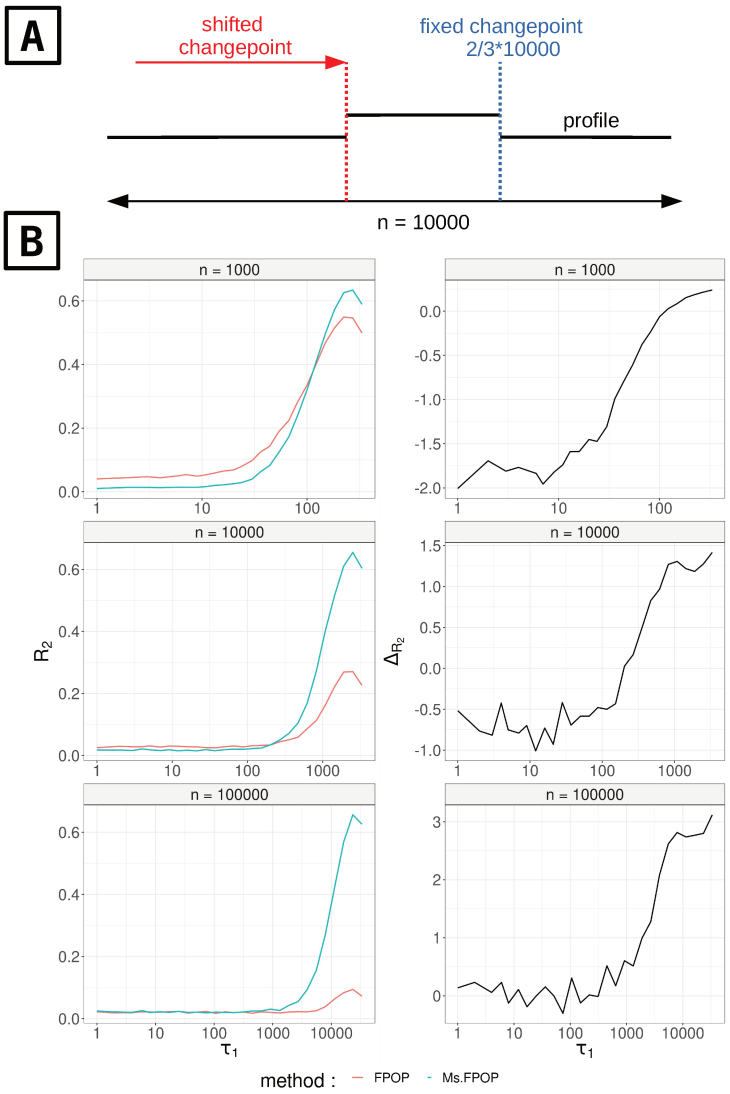


Figure 4: **Ms.FPOP increases the probability of finding well spread changepoints on *hat simulations*.** - (A) denoised profile with two changepoints. The second changepoint is fixed at $\lfloor \frac{2n}{3} \rfloor$ while the first one (τ_1) varies on the interval $[1, \lfloor \frac{n}{3} \rfloor]$. The means of the three resulting segments are set to $\mu_1 = 0$, $\mu_2 = \sqrt{\frac{100}{n}}$ and $\mu_3 = 0$ which gives the profile a hat-like appearance. An iid Gaussian noise of mean 0 and variance one is then added (see *Design of simulations*). - (B) The proportion of replicates for which Ms.FPOP and FPOP return two changepoints (R_2) as well as the $\log_2$-ratio of the two estimations (Δ_{R_2}) are computed for varying τ_1 and n.

4.3.2 Extended range of simulation scenarios

Design of simulations Following a protocol written by Fearnhead *et al.* 2020, we simulate different scenarios of *iid* Gaussian signals. Each scenario is defined by a combination of D, n, τ, μ. For each scenario we vary the variance σ^2 (see Supplementary Data of [Fearnhead and Rigaill, 2020]). All the simulated profiles, with a variance one, can be seen in see Appendix H. Based on these initial scenarios we simulate another set of profiles in which profile lengths are multiplied so that each segments contain at least 300 datapoints. These new set of simulated profiles can be seen in Appendix G. For each scenario and tested σ^2 we simulate 300 replicates.

Metric We denote $AE\%$, the average number of times a method is at least as good as other methods in terms of absolute difference between the true number of changes and the estimated number of changes (Δ_D), mean squared error (MSE) or adjusted rand index (ARI). The closer to 100 (AE%), the better the method. See Supplementary Data of [Fearnhead and Rigaill, 2020] for a formal definition of this criterion.

Empirical results On the simulation of [Fearnhead and Rigaill, 2020] in which a large portion of the segments have a length under 100 the performance of Ms.FPOP are worse than FPOP and MOSUM [Meier et al., 2021] on almost all scenarios except $Dt7$ that do not contain any changepoint (see Appendix H).

On the second set of profiles, using Δ_D as comparison criterion, we observe on Figure 5 that Ms.FPOP get similar performance or is better than FPOP and MOSUM in all scenarios marginaly to σ^2. The results are similar when we use MSE or ARI as a criterion of comparison (see Appendix G).

5 Discussion

Extending functional pruning techniques to the multiscale penalty In section 2.2 we have explained how to extend functional pruning techniques to the case of multiscale penalty. In Figures 1 and 3 we have seen that for large signals ($n \geq 10^5$) with few

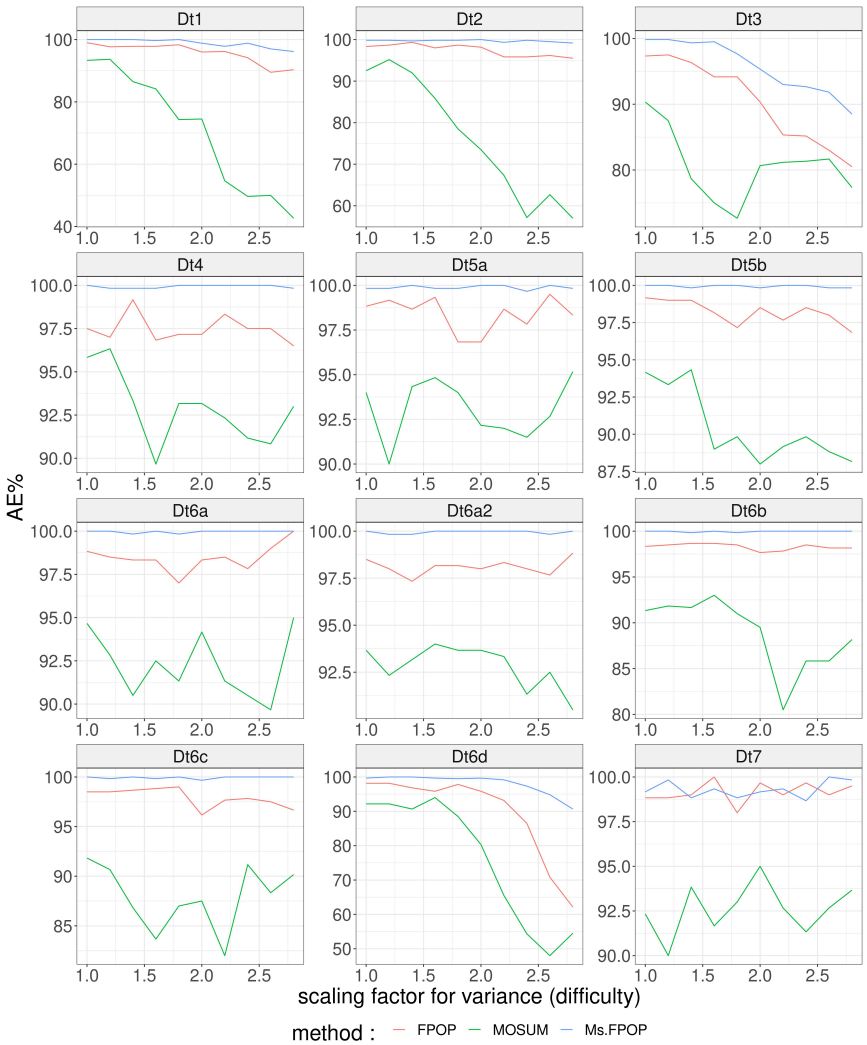


Figure 5: **AE% as a function of the scaling factor for the variance (comparison criterion : Δ_D).** The average number of times a method is at least as good as other methods in terms of Δ_D is computed for FPOP, Ms.FPOP, and MOSUM on different scenarios of *iid* Gaussian signals and varying σ^2. The smallest segment length is greater or equal to 300 (see *Design of Simulations*). Each panel stands for the results on one scenario. Corresponding profiles can be viewed in Appendix G.

changepoints, Ms.FPOP is an order of magnitude faster than Ms.PELT (which relies on inequality based pruning, see Appendix A and B). Even when the number of changepoints increased linearly with the size of the data, Ms.FPOP was still faster than Ms.PELT.

The main update rule (12) of our dynamic programming algorithm suggests to compare each candidate change s with a set of future candidate changes s'. As we have seen in Appendix D, the strategy of randomly drawing one s' according to a uniform distribution is the best strategy and allows us to tackle large signals. It is likely that uniform sampling is not optimal. The algorithm alternates between good draws (leading to a strong reduction of $Z_{t,s}$ or even the pruning of s) and bad draws (leading to a weak reduction $Z_{t,s}$). On average this is sufficient but improvements are possible. In particular the study of $h(t, s, s') = \log(\frac{t-s'}{t-s})$ (see Assumption A1), suggests disfavoring s' that are too recent or that have been compared recently.

Calibration of γ and β from the multiscale penalty The least-squares estimator with multiscale penalty proposed by [Verzelen et al., 2020] involves two constants γ and β that still need to be investigated. Using signals simulated under the null hypothesis (no changepoint) we have seen that it is possible to find a pair of constants $\gamma = 9$ and $\beta = 2.25$ for which Ms.FPOP controls $R_{>0}$. Under this setting we have shown on *hat* (see section 4.3.1) and *step* (see Appendix F) simulations that Ms.FPOP is more powerful than FPOP on segmentations with well-spread changepoints. This difference of power grows with n. For segmentation with small segments FPOP is more powerful Ms.FPOP when n is small ($\approx 10^3$), but for larger n ($\geq 10^4$) this difference disappears.

We also tested Ms.FPOP on the benchmark proposed in [Fearnhead and Rigaill, 2020]. The performances of Ms.FPOP are not so good on the original benchmark containing mostly small profiles with small segments but much better for an extended benchmark with larger profiles (see section 4.3.2).

Without additional work on the calibration of the constants, we would thus recommend using Ms.FPOP for large profiles ($\geq 10^4$).

Unknown variance All our simulations have been done on signals with known variance, σ^2. However, in real-world situations, this may not always be the case. One approach is to estimate σ^2 and then plugging-in it in the problem, *i.e* scaling the signal or the penalty by $\frac{1}{\sigma^2}$ or σ^2, respectively. A robust estimate of σ^2 can be obtained by calculating the variance

of $\Delta_Y = Y_{i+1} - Y_i$ using either the median absolute deviation or the estimator suggested in [Hall et al., 1990]. As an alternative, [Verzelen et al., 2020] pointed out that one could calibrate the multiplicative constant L of the penalized least-squares estimator using the slope heuristic [Arlot, 2019]. Investigating the performances of these various approaches is outside the scope of this paper.

Declarations

Funding This work was supported by an ATIGE grant from Genopole. The IPS2 benefits from the support of the LabEx Saclay Plant Sciences-SPS.

Conflict of interest The authors declare that they have no conflict of interest.

Code availability The scripts used to generate the figures are available in the following GitHub repository: `https://github.com/aLiehrmann/MsFPOP_paper`. A reference C++ implementation of the Ms.FPOP (and Ms.PELT) algorithm is available in the R package of the same name: `https://github.com/aLiehrmann/MsFPOP`.

Acknowledgements We thank Nicolas Verzelen from MISTEA Laboratory, INRAE (Montpellier, France) for helpful discussions regarding the multiscale penalty.

Supplemental material

Appendix A PELT for multiscale penalized likelihood

Appendix B Adaptative PELT for concave multiscale penalty

Appendix C Ms.FPOP: calibration of constants γ and β from the multiscale penalty

Appendix D Ms.FPOP speed benchmark

Appendix E FPOP vs Ms.FPOP: simulations on hat profiles

Appendix F FPOP vs Ms.FPOP: simulations on step profiles

Appendix G FPOP vs Ms.FPOP vs MOSUM: simulations on several scenarios of Gaussian signals (segments length > 300)

Appendix H FPOP vs Ms.FPOP vs MOSUM: simulations on several scenarios of Gaussian signals

References

S. Aminikhanghahi and D. J. Cook. A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367, 2017.

S. Arlot. Minimal penalties and the slope heuristics: a survey. 2019.

I. E. Auger and C. E. Lawrence. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology*, 51:39–54, 1989.

J. Bai and P. Perron. Computation and analysis of multiple structural change models. *Journal of applied econometrics*, 18(1):1–22, 2003.

R. Bellman. On the approximation of curves by line segments using dynamic programming. *Communications of the ACM*, 4:284, 1961.

N. R. Council et al. Frontiers in massive data analysis. 2013.

P. Fearnhead and G. Rigaill. Relating and comparing methods for detecting changes in mean. *Stat*, 9(1), Jan. 2020.

K. Frick, A. Munk, and H. Sieling. Multiscale change point inference. *Royal Statistical Society*, 76:495–580, 2014.

P. Fryzlewicz. Detecting possibly frequent change-points: Wild binary segmentation 2 and steepest-drop model selection. *Journal of the Korean Statistical Society*, 49(4):1027–1070, 2020.

E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment. *Autonomous Robots*, 41(6):1367–1382, 2017.

P. Hall, J. Kay, and D. Titterington. Asymptotically optimal difference-based estimation of variance in nonparametric regression. *Biometrika*, 77(3):521–528, 1990.

Z. Harchaoui and C. Lévy-Leduc. Multiple change-point estimation with a total variation penalty. *Journal of the American Statistical Association*, 105(492):1480–1493, 2010.

B. Jackson, J. Scargle, D. Barnes, S. Arabhi, A. Alt, P. Gioumousis, E. Gwin, P. San, L. Tan, and T. T. Tsai. An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters*, 12:105–8, 2005.

S. W. Jewell, T. D. Hocking, P. Fearnhead, and D. M. Witten. Fast nonconvex deconvolution of calcium imaging data. *Biostatistics*, 21(4):709–726, 2020.

R. Killick, P. Fearnhead, and I. A. Eckley. Optimal detection of changepoints with a linear computational cost. *Statistics and Computing*, 107:1590–98, 2012.

E. Lebarbier. Detecting multiple change-points in the mean of a gaussian process by model selection. *Signal Processing*, 87:717–36, avril 2005.

R. Maidstone, T. Hocking, G. Rigaill, and P. Fearnhead. On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, 27:519–33, 2017.

A. Meier, C. Kirch, and H. Cho. mosum: A package for moving sums in change-point analysis. *Journal of Statistical Software*, 97:1–42, 2021.

A. B. Olshen, E. S. Venkatraman, R. Lucito, and M. Wigler. Circular binary segmentation for the analysis of array-based dna copy number data. *Biostatistics*, 5(4):557–572, 2004.

F. Picard, S. Robin, M. Lavielle, C. Vaisse, and J.-J. Daudin. A statistical approach for array cgh data analysis. *BMC Bioinformatics*, 6(1):27, 2005.

A. Ranganathan. Pliss: labeling places using online changepoint detection. *Autonomous Robots*, 32(4):351–368, 2012.

J. Reeves, J. Chen, X. L. Wang, R. Lund, and Q. Q. Lu. A review and comparison of changepoint detection techniques for climate data. *Journal of Applied Meteorology and Climatology*, 46(6):900–915, June 2007.

G. Rigaill. A pruned dynamic programming algorithm to recover the best segmentations with 1 to kmax change-points. *Journal de la Société Française de Statistique*, 156:180–205, 2015.

S. Thies and P. Molnár. Bayesian change point analysis of bitcoin returns. *Finance Research Letters*, 27:223–227, Dec. 2018.

C. Truong, L. Oudre, and N. Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, Feb. 2020.

N. Verzelen, M. Fromont, M. Lerasle, and P. Reynaud-Bouret. Optimal change-point detection and localization. *arXiv preprint arXiv:2010.11470*, 2020.

Y.-C. Yao. Least-squares estimation of a step function. *Indian Journal of Statistics*, 51: 370–81, 1989.

Supplemental material for "Ms.FPOP: an exact and fast segmentation algorithm with a multiscale penalty"

May 22, 2023

Appendix A PELT for multiscale penalized likelihood

Following the notation of the PELT paper [Killick et al., 2012] the cost of a segment from $s+1$ to s', $s+1:s'$ is defined as $\mathcal{C}_{s+1:s'} = \sum_{i=s+1}^{s'}(y_i - \bar{y}_{s+1:s'})^2 - \beta g(s'-s)$. In what follow we consider three time points $s < s' < t$. Let $\ell = s' - s$ denote the length of the sequence of observations between time s and s' and $\ell' = t - s'$ denote the length of the sequence of observations between time s' and t.

The key condition to apply the PELT algorithm [Killick et al., 2012] is that up to a constant K adding a changepoints always reduce the cost, that is :

Assumption 1.

$$\mathcal{C}_{s+1:s'} + \mathcal{C}_{s'+1:t} + K \leq \mathcal{C}_{s+1:t} \tag{1}$$

The following lemma ensure that such K exists for any n and provide explicit values for K in general and if g is concave.

Lemma 1. *(a) For any function g from $\mathbb{R}$ to $\mathbb{R}$, $\beta \geq 0$, and any n, Assumption 1 is true at least for $K = 2\beta \min_{1\leq\ell\leq n}\{g(\ell)\} - \beta \max_{1\leq\ell\leq n}\{g(\ell)\}$. (b) If g is concave the condition is true for $K = -\beta g(2) + 2\beta g(1)$.*

Proof. We first note that

$$m_n = \min_{1\leq \ell < n} \left\{ \min_{\substack{1\leq \ell' < n \\ \ell+\ell' \leq n}} \{g(\ell) + g(\ell') - g(\ell+\ell')\} \right\}$$

is well defined as the minimum of a finite set. By definition of m_n we thus have, for any $1 \leq s < s' < t \leq n$ and for any $K < \beta m_n$, that

$$-\beta g(s'-s) - \beta g(t-s') + K \quad \leq \quad -\beta g(t-s)$$

Combining this with

$$\sum_{i=s+1}^{s'} (y_i - \bar{y}_{s+1:s'})^2 + \sum_{i=s'+1}^{t} (y_i - \bar{y}_{s'+1:t})^2 \quad \leq \quad \sum_{i=s+1}^{t} (y_i - \bar{y}_{s+1:t})^2.$$

we recover that equation (1) is true for any $K < \beta m_n$.

Now for any ℓ, ℓ' in $\{1, \ldots, n\}^2$ such that $\ell + \ell' \leq n$ we have

$$2 \min_{1\leq \ell \leq n} \{g(\ell)\} - \max_{1\leq \ell \leq n} \{g(\ell)\} \leq g(\ell) + g(\ell') - g(\ell+\ell').$$

Hence we get

$$2 \min_{1\leq \ell \leq n} \{g(\ell)\} - \max_{1\leq \ell \leq n} \{g(\ell)\} \leq m_n,$$

and we recover (a).

In case g is concave using the technical lemma 2 two times we get :

$$\min_{\substack{1\leq \ell' < n \\ \ell+\ell' \leq n}} \{g(\ell) + g(\ell') - g(\ell+\ell')\} = g(\ell) + g(1) - g(\ell+1) \tag{2}$$

and

$$\min_{1\leq \ell < n} \left\{ \min_{\substack{1\leq \ell' < n \\ \ell+\ell' \leq n}} \{g(\ell) + g(\ell') - g(\ell+\ell')\} \right\} = 2g(1) - g(2)$$

For example, if $g = \log$ we get $K = -\beta \log(2)$ □

Lemma 2. *If g is concave then for any $\delta > 0$, the function $h : x \to g(x+\delta) - g(x)$ is non increasing.*

Proof. Consider any $\delta' > 0$. We have $x + \delta = (1 - \alpha)x + \alpha(x + \delta + \delta')$ for $\alpha = \delta/(\delta + \delta')$ and similarly $x + \delta' = (1 - \alpha')x + \alpha'(x + \delta + \delta')$ with $\alpha' = \delta'/(\delta + \delta')$. Using concavity we have

$$\begin{aligned} g(x + \delta) &\geq (1 - \alpha)g(x) + \alpha g(x + \delta + \delta') \\ g(x + \delta') &\geq (1 - \alpha')g(x) + \alpha' g(x + \delta + \delta'). \end{aligned}$$

Suming these two lines and noting that $\alpha + \alpha' = 1$ we get $g(x+\delta) - g(x) \geq g(x+\delta'+\delta) - g(x+\delta')$

□

Appendix B Adaptative PELT for concave multiscale penalty

In the following lemma we show that for our multiscale penalty assuming the function g is concave the constant K in theorme 3.1 of [Killick et al., 2012] can be chosen adaptively to the length of the last segment.

Lemma 3. *If g is concave and $\beta \geq 0$. then if at time s' we have,*

$$F_s + \sum_{i=s+1}^{s'} (y_i - \bar{y}_{s+1:s'})^2 - \beta g(\ell) + K_{s'-s=\ell} \geq F_{s'},$$

with $K_\ell = \beta(g(\ell) + g(1) - g(\ell + 1))$ then for any time t larger than s' we have :

$$F_s + \sum_{i=s+1}^{t} (y_i - \bar{y}_{s+1:t})^2 - \beta g(\ell + \ell') \geq F_{s'} + \sum_{i=s'+1}^{t} (y_i - \bar{y}_{s'+1:t})^2 - \beta g(\ell'),$$

and thus for any time $t \geq s'$, a change at s can never be optimal. Taking $g = \log$ we get $K_\ell = -\beta \log(\frac{1}{\ell} + 1) \leq -\beta \log(2)$.

Proof. We follows the proof of Theorem 3.1 of [Killick et al., 2012] using the fact that if g is concave then equation 2 is true. □

Appendix C Ms.FPOP: calibration of constants γ and β from the multiscale penalty

The following plots were generated to calibrate the constants in the multiscale penalty of [Verzelen et al., 2020]. They are generated as explained in section 4.1.

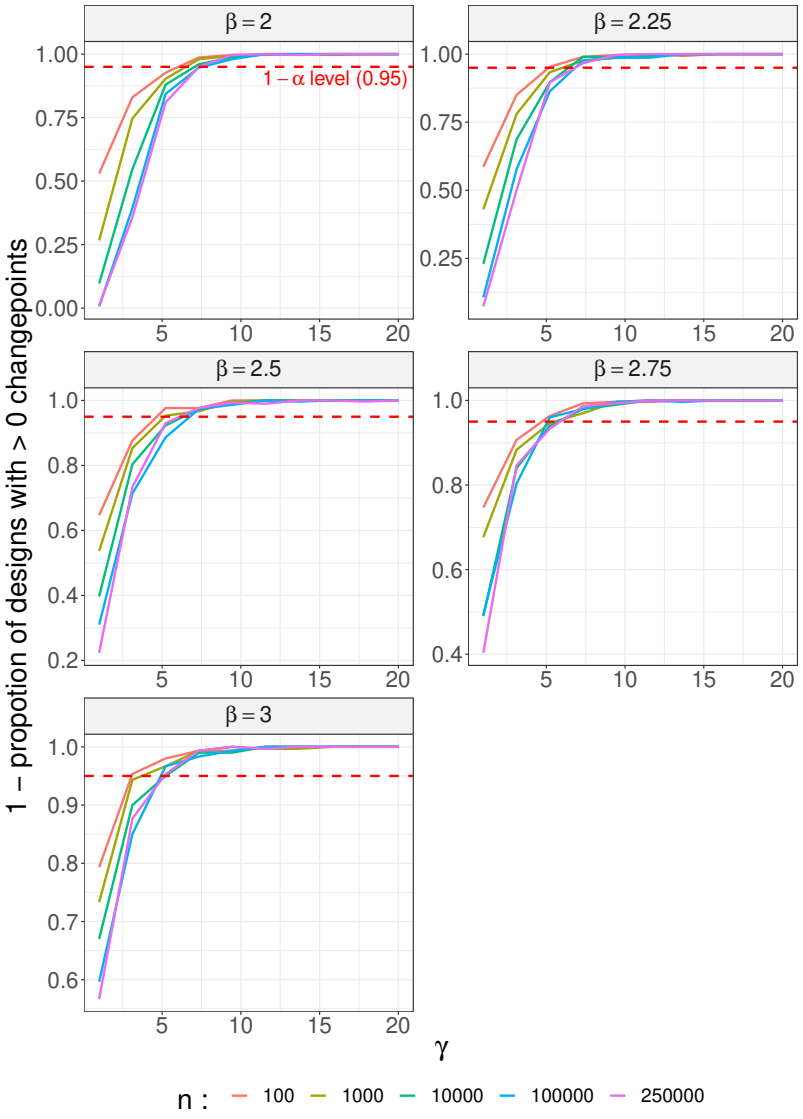


Figure 1: **Proportion of stationary Gaussian process replicates on which Ms.FPOP returns at least one changepoint ($R_{>0}$).** $R_{>0}$ is computed for a series of γ, β, and profile lengths (see *Design of Simulations* in section 4.1.

Appendix D Ms.FPOP speed benchmark

Sampling strategies We compared the runtime of Ms.FPOP for various sampling strategies (see section 2.2.2). We tested sampling 1, 2, 3 and all future changes. We call these strategies respectively rand 1, rand 2, rand 3 and all. We tested them on the simulation described in section 4.2.

It can be seen on the Figure D2 that sampling 1 future change uniformaly at random is the fastest for all true number of changes and $n = 10^5$.

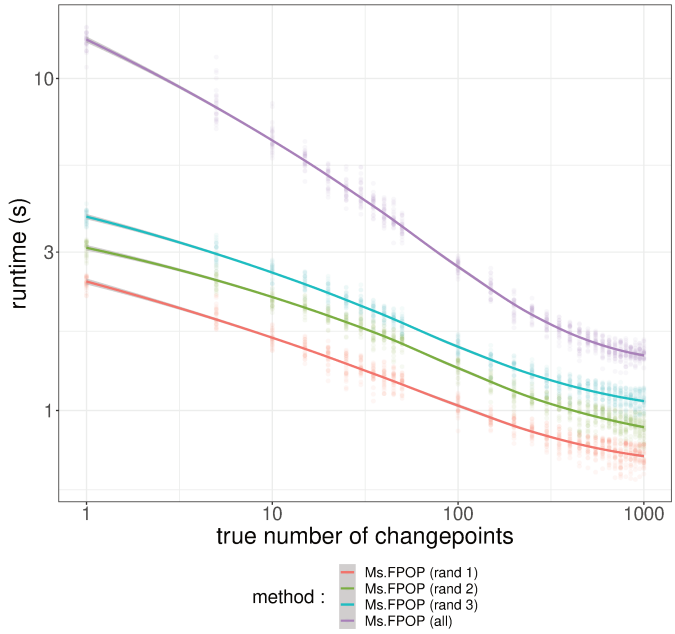


Figure 2: **Runtimes as a function of the true number of changepoints.** We timed strategies consisting in randomly sampling one, two, three, four or all future candidates on profiles of length $n = 10^5$ with varying number of true changepoints D (see *Design of Simulations* in section 4.2) on an Intel Core i7-10810U CPU @ 1.10GHzx12 computer. We observe that, marginaly to D, randomly sampling 1 future candidate (Ms.FPOP rand 1) is faster than randomly sampling more than one future candidates, in particular all future candidates.

Larger profile lengths ($n = 10^6$) Figure D3 is obtained as explained in section 4.2, with $n = 10^6$ and $D \in \{1, 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000, 5500, 6000, 6500,$

$7000, 7500, 8000, 8500, 9000, 9500, 10000\}$.

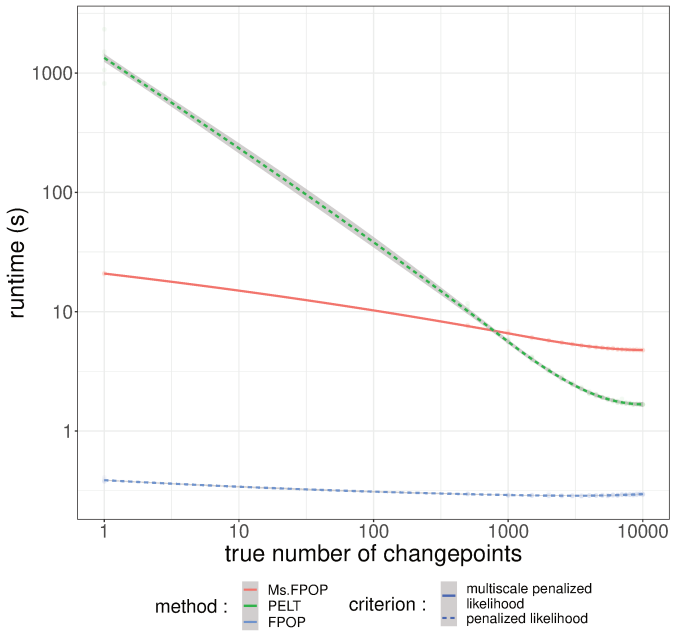


Figure 3: **Runtimes as a function of the true number of changepoints.** We timed PELT, FPOP, Ms.FPOP on profiles of length $n = 10^6$ with varying number of true changepoints D (as described above) on an Intel Core i7-10810U CPU @ 1.10GHzx12 computer.

Appendix E FPOP vs Ms.FPOP: simulations on hat profiles

Figure E4 is obtained as explained in section 4.3.1.

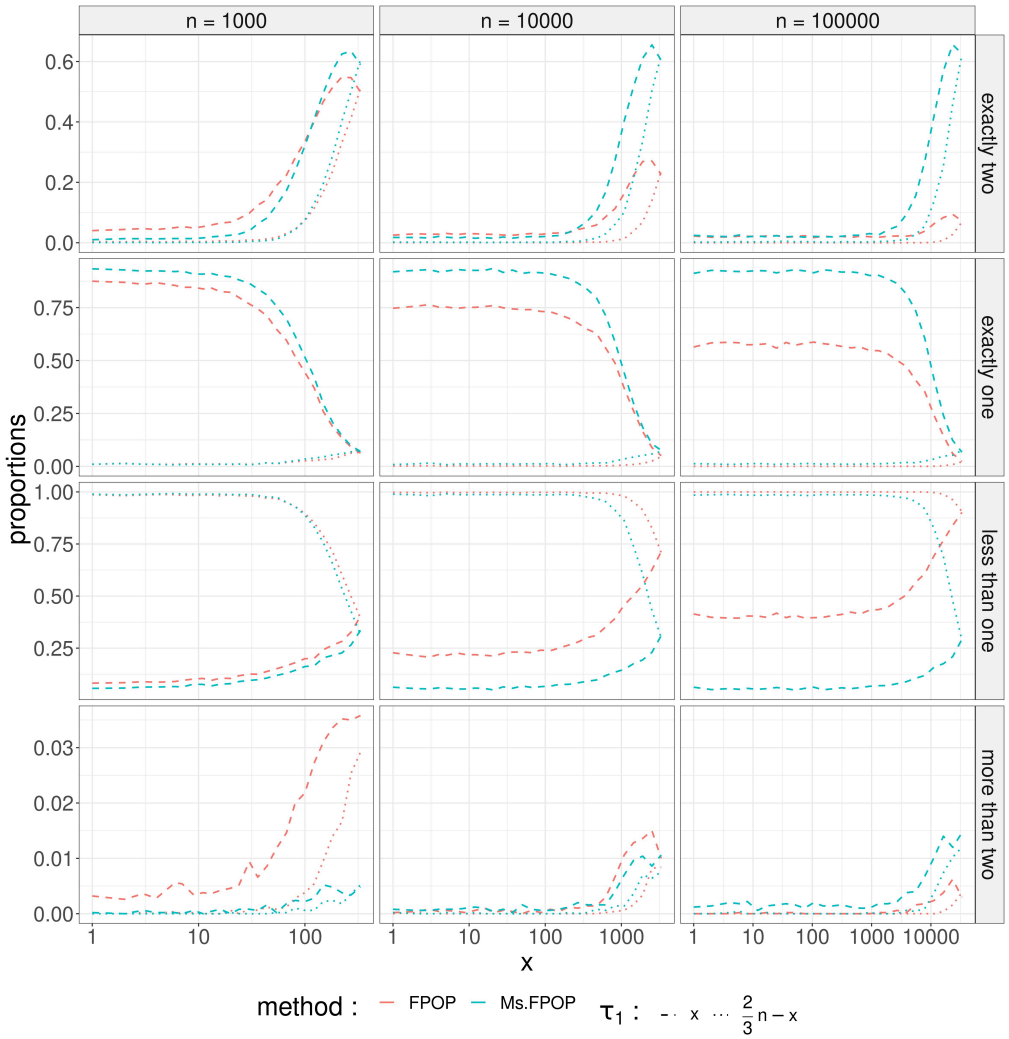


Figure 4: **Ms.FPOP increases the probability of finding well spread changepoints on *hat profiles*.** The proportion of replicates for which Ms.FPOP and FPOP return two changepoints, one changepoint, zero changepoint, and more than two changepoints are computed for varying $\tau_1 \in [1, \lfloor \frac{2}{3}n \rfloor - 1]$ and $n \in [10^3, 10^4, 10^5]$ on the hat-like profiles (see *Design of Simulations* in section 4.3.1).

Appendix F FPOP vs Ms.FPOP: simulations on step profiles

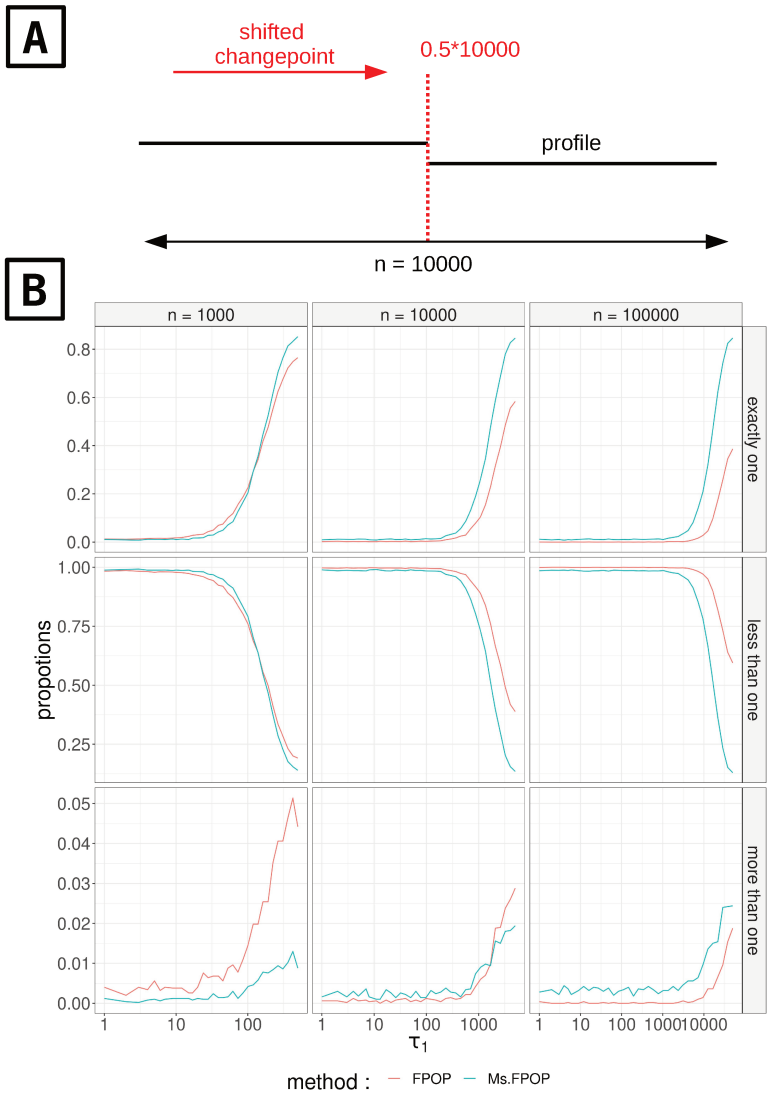


Figure 5: **Ms.FPOP increases the probability of finding well spread changepoint on *step profiles*.** - (A) denoised profile with one changepoint. The first and unique changepoint (τ_1) varies on the interval $[1, \lfloor \frac{n}{2} \rfloor]$ (40 positive integers evenly spaced on the log scale). The difference in mean between the second and the first segment is set to $\sqrt{\frac{70}{n}}$. An *iid* Gaussian noise of variance one is then added. - (B) The proportion of replicates on which Ms.FPOP and FPOP return one changepoint, less than one changepoint and more than one changepoints are computed for varying profile lengths ($n \in 10^3, 10^4, 10^5$) and τ_1.

Appendix G FPOP vs Ms.FPOP vs MOSUM: simulations on several scenarios of Gaussian signals (segments length > 300)

Figures G6, 5, G7, G8 were obtained as explained in section 4.3.2 when considering the benchmark in [Fearnhead and Rigaill, 2020]. On these simulations a large portion of the segments have a length under 100.

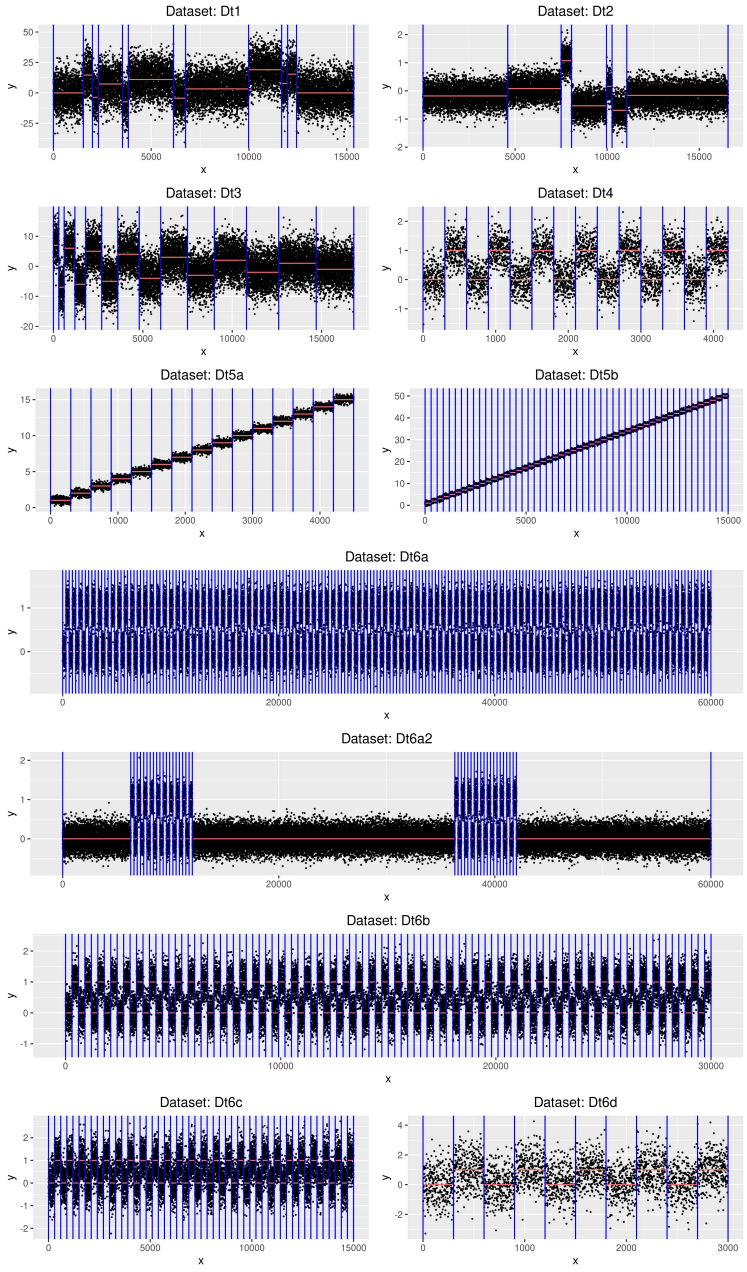


Figure 6: **Simulated scenarios of Gaussian signals with minimum segments length equal to 300.** All scenarios have been simulated following the protocol written by Fearnhead *et al.* (2020). The length of each segment are scaled so that, in each profile, all segments contain at least 300 datapoints.

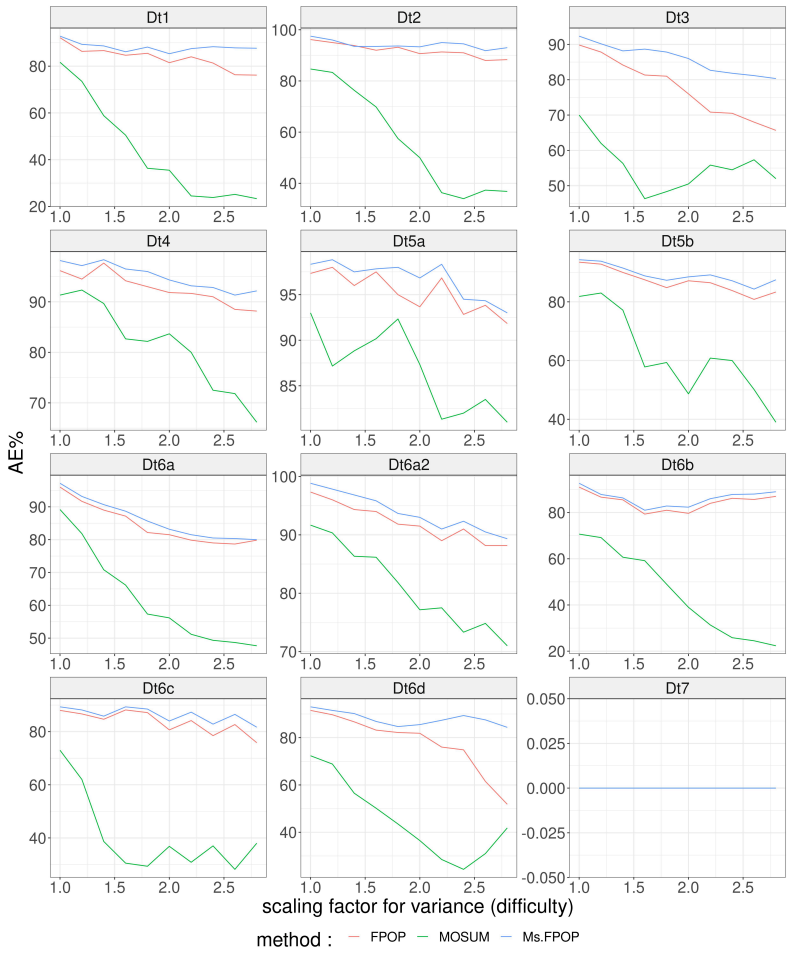


Figure 7: **AE% as a function of the scaling factor for the variance (comparison criterion : ARI).** The average number of times a method is at least as good as other methods in terms of ARI is computed for FPOP, Ms.FPOP, and MOSUM on different scenarios of *iid* Gaussian signals and varying σ^2. The smallest segment length is greater or equal to 300. Each panel stands for the results on one scenario. Corresponding profiles can be viewed in Figure G6.

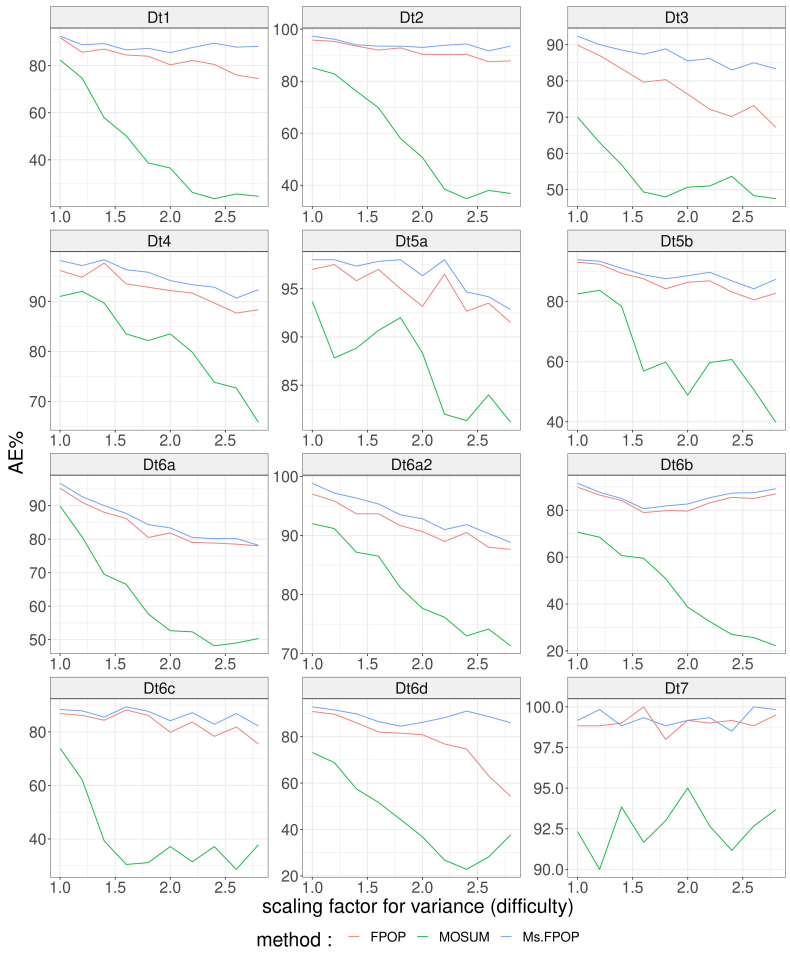


Figure 8: **AE% as a function of the scaling factor for the variance (comparison criterion : MSE).** The average number of times a method is at least as good as other methods in terms of MSE is computed for FPOP, Ms.FPOP, and MOSUM on different scenarios of *iid* Gaussian signals and varying σ^2. The smallest segment length is greater or equal to 300. Each panel stands for the results on one scenario. Corresponding profiles can be viewed in Figure G6.

Appendix H FPOP vs Ms.FPOP vs MOSUM: simulations on several scenarios of Gaussian signals

Figures H9, H10, H11, H12 were obtained as explained in section 4.3.2 when considering an extension of the benchmark in [Fearnhead and Rigaill, 2020]. Based on the initial scenarios we simulated another set of profiles in which segments length are multiplied so that each of segments contain at least 300 datapoints.

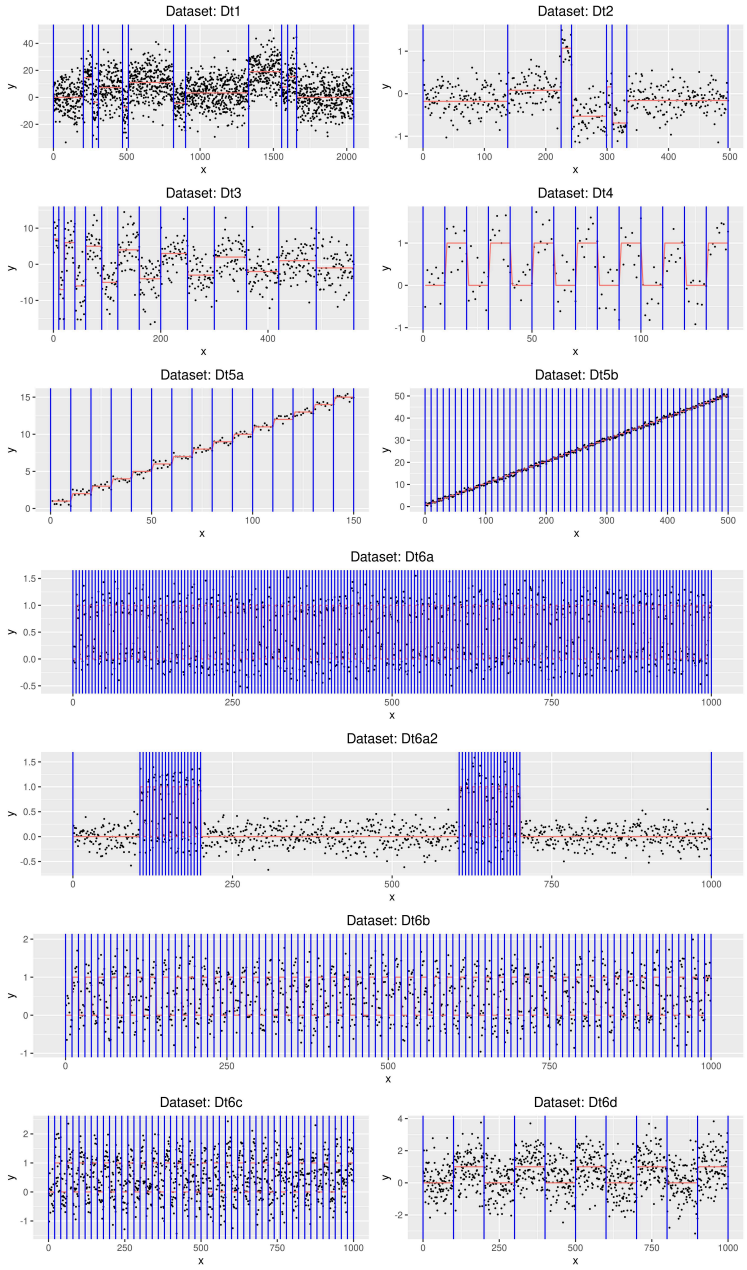


Figure 9: **Simulated scenarios of Gaussian signals.** All scenarios have been simulated following the protocol written by Fearnhead *et al.* (2020).

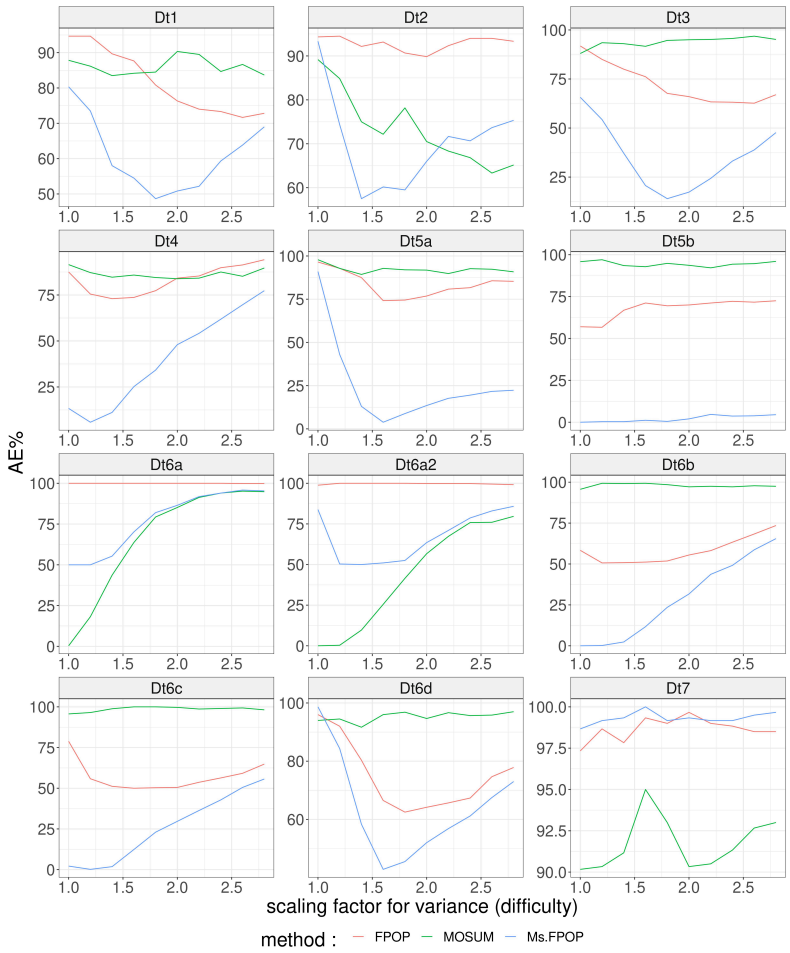


Figure 10: **AE% as a function of the scaling factor for the variance (comparison criterion : Δ_D).** The average number of times a method is at least as good as other methods in terms of Δ_D is computed for FPOP, Ms.FPOP, and MOSUM on different scenarios of *iid* Gaussian signals and varying σ^2. Each panel stands for the results on one scenario. Corresponding profiles can be viewed in Figure H9.

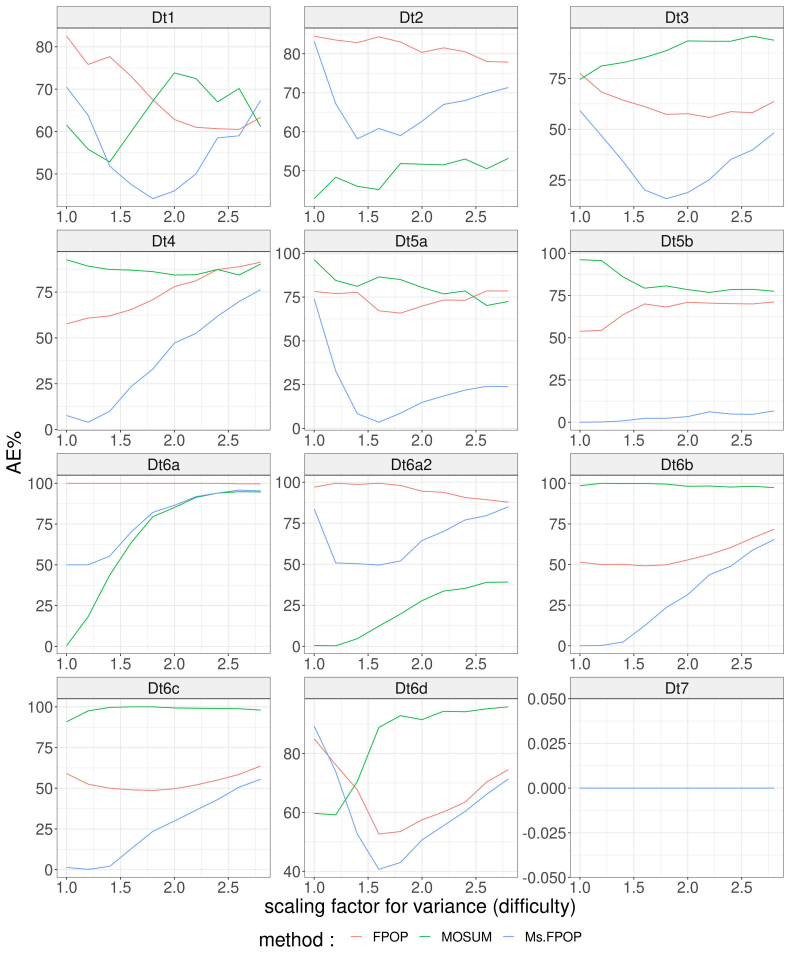


Figure 11: **AE% as a function of the scaling factor for the variance (comparison criterion : ARI).** The average number of times a method is at least as good as other methods in terms of ARI is computed for FPOP, Ms.FPOP, and MOSUM on different scenarios of *iid* Gaussian signals and varying σ^2. Each panel stands for the results on one scenario. Corresponding profiles can be viewed in Figure H9.

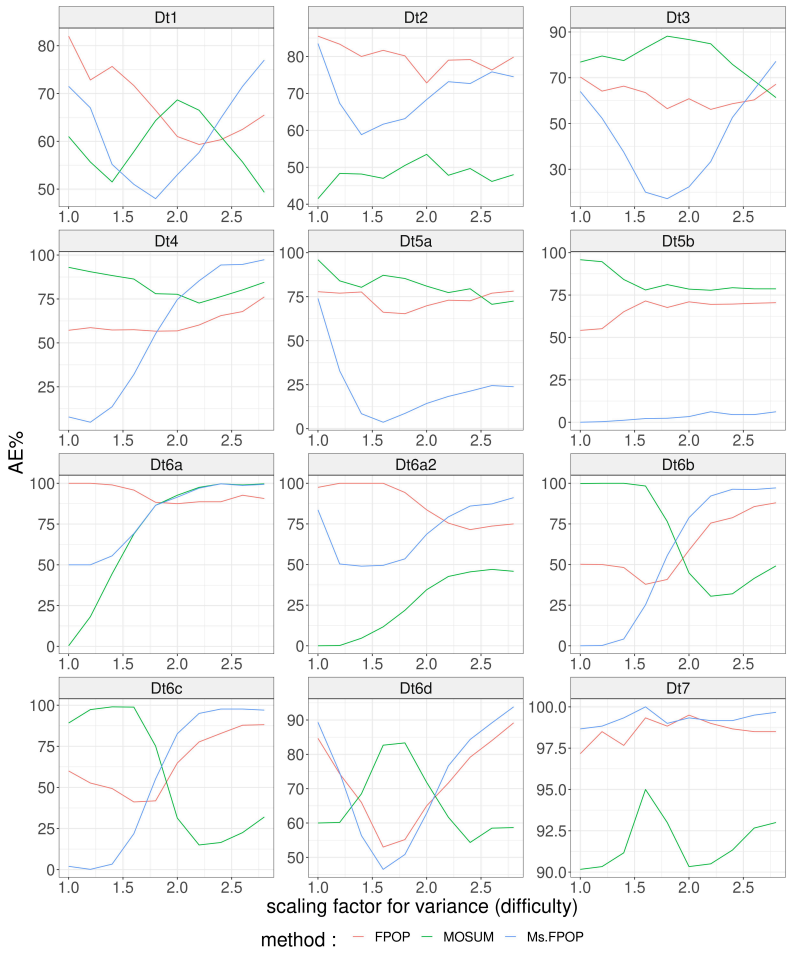


Figure 12: **AE% as a function of the scaling factor for the variance (comparison criterion : MSE).** The average number of times a method is at least as good as other methods in terms of MSE is computed for FPOP, Ms.FPOP, and MOSUM on different scenarios of *iid* Gaussian signals and varying σ^2. Each panel stands for the results on one scenario. Corresponding profiles can be viewed in Figure H9.

References

P. Fearnhead and G. Rigaill. Relating and comparing methods for detecting changes in mean. *Stat*, 9(1), Jan. 2020.

R. Killick, P. Fearnhead, and I. A. Eckley. Optimal detection of changepoints with a linear

computational cost. *Statistics and Computing*, 107:1590–98, 2012.

N. Verzelen, M. Fromont, M. Lerasle, and P. Reynaud-Bouret. Optimal change-point detection and localization. *arXiv preprint arXiv:2010.11470*, 2020.

B.2 Implementation of Ms.FPOP

To improve the readability of this section, I have defined a syntax for each programming concept I will refer to, namely : ***A_class***, *an_attribute / a_variable*, *An_object_method()*, *A_class_method()*. When needed, the reader is also encouraged to refer to the corresponding class diagram B.1.

Candidate. The ***Candidate*** class defines a last changepoint candidate. This changepoint is characterized by its position s. Its cost function $\widetilde{f}_{t,s}(\mu)$ can be broken down and stored in three separate attributes. The constant part of this function $(F_s + \alpha)$ is stored in the attribute named *cost_up_to_s*. The quadratic form $(\sum_{i=s+1}^{t}(y_i - \mu)^2 + \alpha)$, an object of type ***Quadratic***, is stored in the *quad* attribute. Lastly, the penalty component, which is dependent on the length of the last segment is stored in the attribute *pen*. The living set of the candidate, an object of type ***Ordered_list_of_intervals***, is saved in the z attribute.

This class has a constructor that allows a changepoint candidate to be instantiated by specifying parameters that correspond to the various attributes mentioned above. There are several methods to interact with a changepoint candidate :

- *Minimum_of_cost_function()* returns the minimum of the cost function by first calling *Minimum()* on the quadratic form, which returns the optimal cost of the last segment, then adding *cost_up_to_s* and *pen* ;
- *Set_penalty()* updates *pen* with the specified value ;
- *Add_quadratic()* adds the current point y_i to the quadratic form by adding appropriate coefficients $(a_0 = a_0 + y_i^2; \quad a_1 = a_1 + 2y_i; \quad a_2 = a_2 + 1)$;
- *GetZ(), Get_s()* are the accessors for the attributes z and s, respectively ;
- *Compare_to_past_candidate()* updates a changepoint candidate's z attribute by comparing it with changepoint candidates introduced before it. This procedure corresponds to step 18 in Algorithm 1 of Liehrmann and Rigaill [2023]. The naive approach of subtracting $I_{\infty,s'',s}$ from z for each past candidate s'' can become complex if z contains more than one interval. I suggest an alternative approach. We instantiate the sorted union of $I_{\infty,s'',s}$ using the appropriate constructor from the ***Ordered_list_of_intervals*** class. Then we seek the complement of the previously formed sorted union of intervals in $z = [\min(Y), \max(Y)]$. This step is performed by calling the *Complementary_in()* method. The returned object of type ***Ordered_list_of_intervals*** is used to update z ;
- *Compare_to_future_candidate()* updates the current changepoint candididates's z attribute by comparing it with a sample of the changepoint candidates introduced after it, denoted as s'. This procedure corresponds to step 20 in Algorithm 1 of Liehrmann and Rigaill [2023]. We start by instantiating the intersection of $I_{t,s,s'}$ using the appropriate constructor from the ***Interval*** class. Then we intersect each interval contained in z with the previously

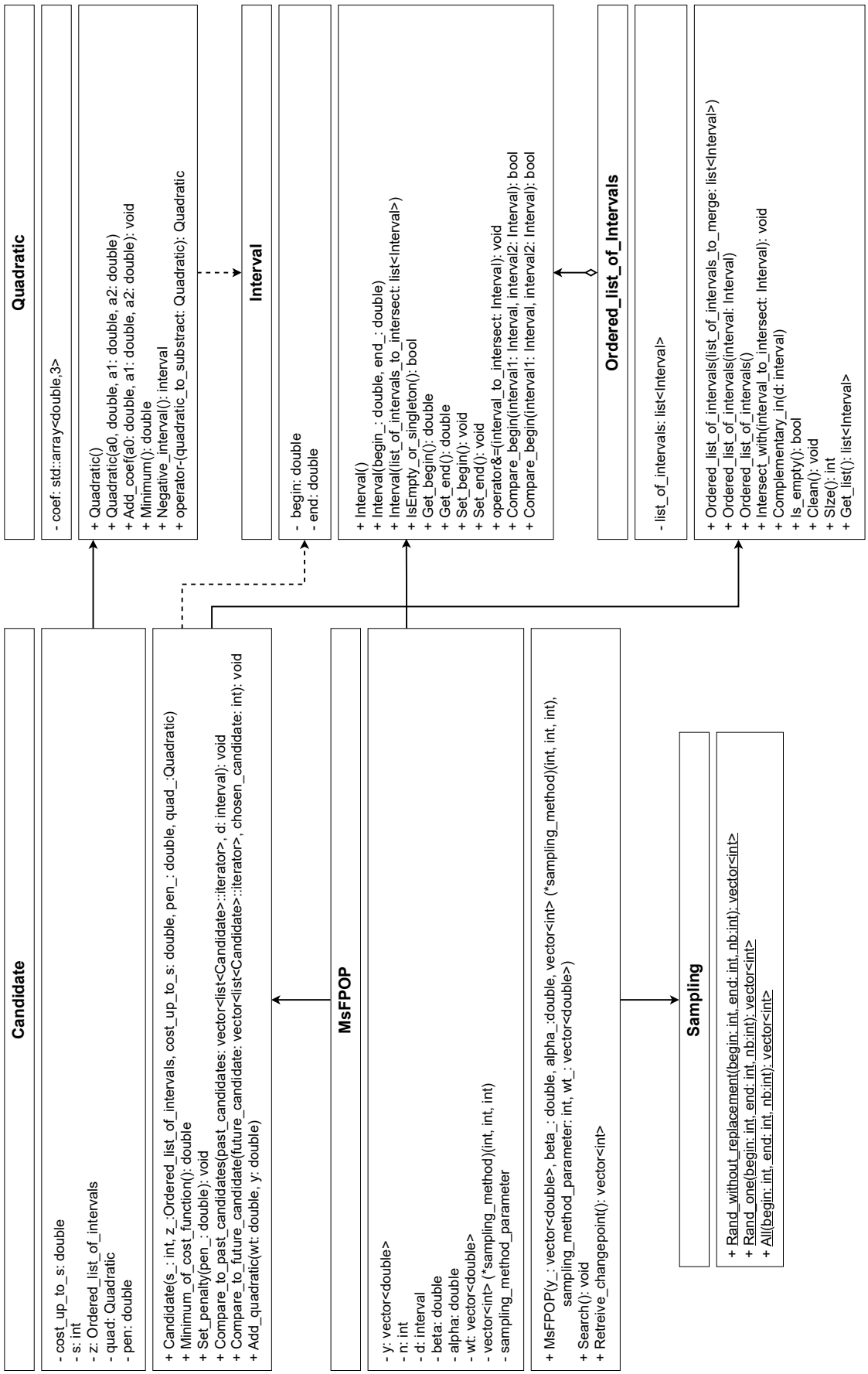


FIGURE B.1 – Class diagram of the MsFPOP project. A solid arrow signifies an association, a dotted arrow indicates a dependency, while a diamond shape represents an aggregation.

instantiated intersection. This step is performed by calling the *Intersect_ with()* method.

Interval. The ***Interval*** class defines a bounded interval in $\mathbb{R}$. The lower and upper bounds are saved in two double type attributes, namely *begin* and *end*. Any interval of the form $[a, b]$ where $a \geq b$ is considered an empty interval. In the case of singletons, this choice is justified by the properties of the piecewise quadratic function. This function is continuous, which implies that the living set of cost functions adjacent to that associated with the singleton intersect at this point. Thus, we can choose to no longer consider the singleton.

This class has three constructors. The first allows instantiating an empty interval. The second permits instantiating an interval with explicitly provided bounds. The last enables instantiating an interval that corresponds to the intersection of intervals saved in an unordered list. To do this, the greatest of the lower bounds and the smallest of the upper bounds are sought in linear time. Several methods allow interaction with an interval :

- *IsEmpty_ or_ singleton()* returns true if the interval is of the form $[a, b]$ such that $a \geq b$, false otherwise ;
- *Get_ begin(), Get_ end(), Set_ begin(...), Set_ end(...)* are the accessors and mutators of the bounds of an interval ;
- *operator&=(...)* interesects two intervals and modifies the bounds of the first interval so that they equal the bounds of the intersection.

Interval also implements two class methods, *Compare_ begin(...)* and *Compare_ end(...)*, which return true when, respectively, the lower bound of the first interval is smaller than the lower bound of the second interval, and when the upper bound of the first interval is smaller than the upper bound of the second interval, false otherwise. These two comparison methods are used by the algorithms of the standard library to manipulate containers of intervals (e.g., search for the minimal upper bound, sorting based on the lower bound).

Ordered_list_of_intervals. The ***Ordered_ list_ of_ intervals*** class defines a non-empty list of intervals that are ordered by their lower bounds. The container *list_ of_ intervals* is of the ***std*::*list*** type and has the properties of a doubly linked list. The ability to remove an element in constant time (linear with the number of elements to remove) makes this container particularly efficient for processing the dynamic living set of changepoint candidates.

This class has three constructors. The first allows instantiating an empty list. The second allows instantiating a list with an explicitly provided interval. The last allows instantiating a sorted list of intervals and empty intersections from a list of possibly unsorted intervals and possibly non-empty intersections. In the latter case, a sorting operation is first performed on the lower bounds of the intervals contained in the provided list. This operation has a complexity of $\mathcal{O}(|list\_of\_intervals\_to\_merge| \times log(|list\_of\_intervals\_to\_merge|))$. Then, the union of sorted intervals is performed. This operation has a complexity of $\mathcal{O}(|list\_of\_intervals\_to\_merge|)$. Several methods allow interaction with the current sorted list of intervals :

- *Intersect_with(...)* updates a sorted list of intervals by intersecting, in place via the overloaded *&=* operator in ***Interval***, each interval contained in this list with *interval_to_intersect*. If an intersection is empty, the concerned interval is removed from the current list in constant time ;
- *Complementary_in(...)* updates a sorted list of intervals with the complement of intervals contained in this list, in *d*. The first step is to call *Intersect_with(d)* on the sorted list of intervals. The current list now only contains intervals strictly included in *d*. This operation has linear time complexity. The operation to find the complement in d also has linear time complexity ;
- *Is_empty()* returns true if the list is empty, false otherwise ;
- *Clean()* removes empty intervals from a sorted list of intervals ;
- *Size()* returns the size of a sorted list of intervals ;
- *Get_list()* returns a sorted list of intervals.

MsFPOP. The ***MsFPOP*** class is the main class in this project. It implements several attributes that allow setting up the segmentation problem. A problem is instantiated using the constructor of this class by specifying a data vector (y), the constants *alpha* and *beta* used in the penalty calculations, a weight vector (wt) associated to the data, a sampling method (*sampling_method*) from those offered by ***Sampling***, and its associated parameter (*sampling_method_parameter*).

The *search()* procedure finds the changepoints. Throughout the procedure, we maintain a list of changepoint candidates and a vector of iterators, each of which points to a changepoint candidate. Thanks to the operator [] implemented by the ***std*::*vector*** class, we can access each changepoint candidate in constant time without needing to traverse this list.

At each step t (see main loop in Algorithm 1 of Liehrmann and Rigaill [2023]), we :

- update the quadratic form by calling *Add_quadratic(...)* on each changepoint candidate with the new data $y[t]$ and associated weight value $wt[t]$ as arguments ;
- update the penalty calculated on the last segment by calling *Set_penalty()* on each changepoint candidate ;
- find the minimum cost (F_t) among all cost functions by calling *Minimum_of_function_cost()* on each changepoint candidate. We save this cost and the associated changepoint candidate ;
- instantiate a new changepoint candidate t via the constructor of ***Candidate***. The quadratic form of its cost function and the penalty calculated on the last segment are null. The constant part of its cost function is equal to $F_t + \alpha$;
- call *Compare_to_past_candidate()* on the recently introduced changepoint candidate ;
- for each changepoint candidate, we sample future changepoint candidate via *sampling_method* and then call *Compare_to_future_candidate()* ;
- remove from the changepoint candidates whose cost function's living set is empty.

The *Retrieve_changepoints()* method recursively finds and returns the list of changepoints that form the best segmentation of the data.

Quadratic. The ***Quadratic*** class defines a second degree polynomial $a0 + a1x + a2x^2$. The coefficients, $a0, a1, and\ a2$, are saved in *coef*, an array of doubles of size 3. This class has two constructors that allow instantiating a quadratic form with either zero coefficients or explicitly provided coefficients. Several methods allow interaction with a quadratic form :

- *Add_coef()* modifies the current quadratic form by explicitly adding the coefficients of another quadratic form ;
- *Minimum()* returns the minimum of the current quadratic form, i.e., $a_0 - \frac{a_1^2}{4a_2}$;
- *Negative_interval()* first calculates the discriminant of a quadratic form ($a_1^2 - 4a_2a_0$). If the discriminant is strictly greater than 0, the method instantiates and returns an ***Interval*** object whose bounds correspond to the ordered roots of the quadratic. Otherwise, the method instantiates and returns an empty ***Interval*** object ;
- *operator-()* overloads the "-" operator. This operator instantiates and returns a quadratic form whose coefficients are equal to the difference between the coefficients of two quadratic upon which this operator is applied.

Sampling. The ***Sampling*** class implements two methods methods for sampling positive integers. These methods are used to sample future changepoint candidates (step 19 in Algorithm 1 of Liehrmann and Rigaill [2023]).

- *Rand_without_replacement(...)* : This function returns a vector containing nb positive integers, randomly sampled without replacement from the range $(begin, end)$. This is possible if $nb < |(begin, end)|$. If not, it returns all positive integers within the range $(begin, end)$.
- *All()* : This function returns a vector containing all positive integers within the range $(begin, end)$.

Chapter C

DiffSegR : An RNA-Seq data driven method for differential expression analysis using changepoint detection

This article was published in the journal *NAR Genomics and Bioinformatics* (doi.org/10.1093/nargab/lqad098).

NAR Genomics and Bioinformatics, 2023, **5**, 1–12
https://doi.org/10.1093/nargab/lqad098
Advance access publication date: 6 November 2023
Methods Article

DiffSegR: an RNA-seq data driven method for differential expression analysis using changepoint detection

Arnaud Liehrmann [1,2,3,*], Etienne Delannoy [1,2], Alexandra Launay-Avon [1,2], Elodie Gilbault [4], Olivier Loudet [4], Benoît Castandet [1,2,*] and Guillem Rigaill [1,2,3,*]

[1]Institute of Plant Sciences Paris-Saclay (IPS2), Université Paris-Saclay, CNRS, INRAE, Université Evry, Gif sur Yvette, 91190, France
[2]Institute of Plant Sciences Paris-Saclay (IPS2), Université Paris Cité, CNRS, INRAE, Gif sur Yvette, 91190, France
[3]Laboratoire de Mathématiques et de Modélisation d'Evry (LaMME), Université d'Evry-Val-d'Essonne, UMR CNRS 8071, ENSIIE, USC INRAE, Evry, 91037, France
[4]Université Paris-Saclay, INRAE, AgroParisTech, Institut Jean-Pierre Bourgin (IJPB), 78000, Versailles, France

[*]To whom correspondence should be addressed. Email: arnaud.liehrmann@universite-paris-saclay.fr
Correspondence may also be addressed to Benoît Castandet. Tel: +33 169157593; Email: benoit.castandet@universite-paris-saclay.fr
Correspondence may also be addressed to Guillem Rigaill. Tel: +33 164853544; Email: guillem.rigaill@inrae.fr

Abstract

To fully understand gene regulation, it is necessary to have a thorough understanding of both the transcriptome and the enzymatic and RNA-binding activities that shape it. While many RNA-Seq-based tools have been developed to analyze the transcriptome, most only consider the abundance of sequencing reads along annotated patterns (such as genes). These annotations are typically incomplete, leading to errors in the differential expression analysis. To address this issue, we present DiffSegR - an R package that enables the discovery of transcriptome-wide expression differences between two biological conditions using RNA-Seq data. DiffSegR does not require prior annotation and uses a multiple changepoints detection algorithm to identify the boundaries of differentially expressed regions in the per-base $\log_2$ fold change. In a few minutes of computation, DiffSegR could rightfully predict the role of chloroplast ribonuclease Mini-III in rRNA maturation and chloroplast ribonuclease PNPase in (3′/5′)-degradation of rRNA, mRNA and tRNA precursors as well as intron accumulation. We believe DiffSegR will benefit biologists working on transcriptomics as it allows access to information from a layer of the transcriptome overlooked by the classical differential expression analysis pipelines widely used today. DiffSegR is available at https://aliehrmann.github.io/DiffSegR/index.html.

Introduction

It has long been recognized that transcriptomes largely surpass genomes in complexity (1). Besides alternative use of transcription initiation sites, most of the transcript diversity can be ascribed to post-transcriptional modifications, including RNA splicing, processing, alternative polyadenylation, editing or base modification (2). Although the advent of the transcriptomics revolution has allowed an unprecedented understanding of this transcript diversity, the combinatorial nature and very large number of variations is still an analytical challenge (3,4). Moreover, because most strategies for RNA-Seq analysis rely on incomplete transcriptomic variant annotations, meaningful variations may currently be overlooked (5). This is a major issue for biological interpretation as illustrated by the crucial role played in disease development by poorly annotated non coding elements like 5′ and 3′ UTRs (6–9).

As a consequence, there is a massive effort to improve transcriptomic annotations with the help of the third generation (long-read) sequencing technologies from Oxford Nanopore Technologies or Pacific Bioscience. Long RNA-Seq reads may cover an entire RNA isoform from start to end, directly illustrating the exon structure, splicing patterns and UTR composition (10–12). They carry the promise to go beyond the limits of full-length transcript assembly, which is notoriously prone to error (13,14). Although such a strategy can double the number of referenced transcripts for a model organism (15), reaching an exhaustive description of a transcriptome is arguably a Sisyphean task (5,16,17).

Because most RNA-Seq experiments aim at identifying RNA processes that vary between two biological conditions (WT versus mutant or control versus stress, for example), an alternative to this issue is to identify portions of the transcriptome that vary between both experimental conditions (differentially expressed regions or DERs) directly from the RNA-Seq data, without relying on annotations and bypassing assembly altogether. This is performed by a class of methods sometimes referred to as identify-then-annotate tools (18). Their gold standard is to be both highly specific (i.e. able to merge adjacent non-DERs) and highly sensitive (i.e. able to discriminate between adjacent DERs, in particular between up and down DERs). To do so, various methods summarized in Figure 1 (19–22) address a well-defined statistical problem known as multiple changepoints detection, or segmentation problem. This has been a long-standing problem in the field of genomic series analysis (23–27). To identify DERs, current identify-then-annotate tools mainly vary in the signal they segment and in the way they segment it (Figure 1).

Here, we introduced DiffSegR, an R package that uses a new strategy for delineating the boundaries of DERs. It segments the per-base $\log_2$ fold change ($\log_2$-FC) using FPOP, a method designed to identify changepoints in the mean of a Gaussian signal (28). Intuitively, the per-base $\log_2$-FC is a

Received: June 22, 2023. **Revised:** September 27, 2023. **Editorial Decision:** October 13, 2023. **Accepted:** October 23, 2023

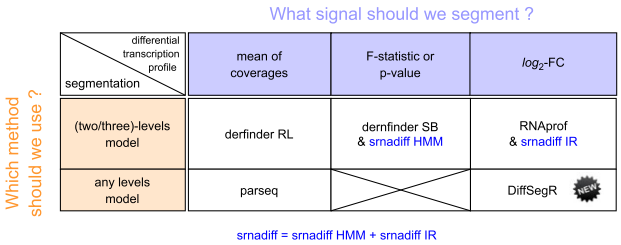


Figure 1. State-of-the-art of Identify-then-annotate methods for detecting differentially expressed regions (DERs) in RNA-Seq data. The methods included in this figure—srnadiff, srnadiff IR, srnadiff HMM (19), derfinder SB, derfinder RL (22), RNAprof (21), parseq (20) and DiffSegR—belong to a class of methods known as identify-then-annotate, which enable the identification of DERs directly from RNA-Seq data without relying on annotations or assembly. To identify DERs, these methods address a well-defined statistical problem known as multiple changepoints detection or segmentation problem. The methods vary in the signal they segment and the way they segment it. For example, srnadiff merges the results of a three-level segmentation model on the per-base $\log_2$ fold change (srnadiff IR) and a two-level segmentation model on the per-base *P*-value (srnadiff HMM). Similarly, derfinder SB and derfinder RL implement a two-level segmentation model on the per-base *F*-statistic (similar to per-base *P*-value) and the mean of coverages, respectively. RNAprof implements a three-level segmentation model on the per-base $\log_2$ fold change. parseq segments the mean of coverages without assuming the number of levels. Finally, DiffSegR introduces a new strategy to identify DERs by segmenting the per-base $\log_2$ fold change without assuming the number of levels. All the methods except parseq assess the found DERs using DESeq2 (29).

measure that scales with the intensity of the transcription differences at each genomic position between the two compared biological conditions. Expression differences are then statistically assessed for each region using the negative binomial generalized linear model of DESeq2 (29) and the outputs can be visualized in IGV (30).

DiffSegR and competitor methods (Figure 1) were compared on two plant RNA-Seq datasets that were previously used in combination with molecular biology techniques to decipher the roles of the chloroplast ribonucleases PNPase and Mini-III (31,32). DiffSegR was the only method able to retrieve all the segments known to differentially accumulate outside of the annotated genic regions (i.e. 3′ and 5′ extensions, anti-sense accumulation). Moreover, it is the only method predicting the overaccumulation of intronic regions on the plastome-scale in the PNPase mutant. Globally, DiffSegR better captures multiple trends of differences within DERs while being more parsimonious in non-DERs than its competitors.

We anticipate DiffSegR will be an important tool in providing an in-depth description of local or regional transcript variations within RNA-seq libraries from two biological conditions, especially when studying RNA processes located outside of the annotated coding sequences, like RNA processing, trimming or splicing.

Materials and methods

DiffSegR segmentation model

Differential transcription profile

DiffSegR builds the coverage profiles indexed on n genomic positions from the BAM files provided by the user. The coverage profile for replicate r of biological condition j is noted $Q_{jr} = \{Q_{ijr}\}_{i=1}^{n} \in \mathbb{N}^n$. By default we propose to compute Q_{ijr} using the geometric mean of the number of 5′ and 3′ end of the reads overlapping position i, denoted $Q_{ijr5'}$ and $Q_{ijr3'}$. Formally:

$$Q_{ijr} + 1 = \left(Q_{ijr5'} + 1\right)^{1/2} \times \left(Q_{ijr3'} + 1\right)^{1/2}. \quad (1)$$

We describe alternative approaches that use either the full length or the 5′ or 3′ end of the reads, and compare them with our geometric mean heuristic in Notes S1–S3, Supplementary Table S9 and Supplementary Figures S40–S42. DiffSegR then builds the differential transcription profile between the biological conditions (named 1 and 2 hereafter) using a $\log_2$-FC per-base transformation because it scales with the intensity of the transcriptional differences between conditions 1 and 2. The $\log_2$-FC at the ith genomic position (denoted Y_i) is given by

$$Y_i = \frac{1}{R_1}\sum_{r_1=1}^{R_1} \log_2(Q_{i1r_1} + 1) - \frac{1}{R_2}\sum_{r_2=1}^{R_2} \log_2(Q_{i2r_2} + 1) \quad (2)$$

where R_1 and R_2 stand for the number of replicates in condition 1 and 2, respectively.

Segmentation model

We consider D changepoints $\tau_1 < \ldots < \tau_D$ within the range 1 and $n - 1$. These changepoints correspond to unknown positions along the genome where a shift in the mean of the per-base $\log_2$-FC (eq: 2) is observed. We adopt the convention that $\tau_0 = 0$ and $\tau_{|\tau|} = n$. These changepoints define $|\tau| = D + 1$ distinct segments. The jth segment includes the data $]\!]\tau_{j-1}, \tau_j]\!] = \{\tau_{j-1} + 1, \ldots, \tau_j\}$. Each segment is premised on the assumption that the Y_i therein are independent and follow the same Gaussian distribution, with a mean μ_j specific to that segment and a common variance σ^2. Expressed mathematically, we have:

$$\forall\, i \in\,]\!]\tau_{j-1}, \tau_j]\!]\; Y_i \;\sim\; \mathcal{N}\left(\mu_j, \sigma^2\right)\; iid. \quad (3)$$

Estimation of the segment

The parameters of the model (eq: 3), including $\tau_1 < \ldots < \tau_D$, can be estimated using penalized maximum likelihood inference. To achieve this, DiffSegR uses the FPOP algorithm (28) (a dynamic programming algorithm that implements functional pruning techniques) which solves the inference problem exactly (see below). For many profiles lengths the computation time of FPOP is log-linear allowing for the segmentation of large data ($10^6 < n < 10^7$) in a matter of seconds. The number of changepoints estimated by FPOP is a decreasing function of the penalty $\lambda\sigma^2 \log(n)$. The constant λ is a hyperparameter that can be adjusted by the user. A good starting point, based on theoretical arguments (33) and simulations (34), is to set $\lambda = 2$. The variance σ^2 is estimated on the data using the unbiased sample variance estimator.

FPOP

Informally, the idea of the FPOP algorithm is to consider the penalized maximum likelihood of the data from observation 1 to t as a function of the parameter (the mean) of the last segment. This idea is referred to as 'functional pruning'. In the Gaussian case, the resulting function is piecewise quadratic. For a new observation at time $t + 1$, it is possible to efficiently update this function (that is, compute the penalized maximum likelihood function from observation 1 to $t + 1$) using a

formula similar to that of the Viterbi algorithm. This formula is applied piece by piece, that is by intervals. At each step, the algorithm searches for the best possible value of the parameter of the last segment to maximize the penalized likelihood.

Normalization

To account for differences in the total number of sequenced reads per sample, we assume that the mean of the coverage μ_{ijr} is composed of a sample-specific size factor s_{jr} and a parameter q_{ijr} proportional to the expected true concentration of transcripts overlapping position i in replicate r of condition j verifying $\mu_{ijr} = s_{jr}\, q_{ijr}$ (29,35,36). As the coverage (eq: 1), the per-base $\log_2$-FC (eq: 2) depends on sample-specific size factors. One can show that the non-normalized and normalized per-base $\log_2$-FC are linked together by an offset denoted ρ such that

$$\rho = \frac{1}{R_1}\sum_{r_1=1}^{R_1}\log_2(s_{1r_1}) - \frac{1}{R_2}\sum_{r_2=1}^{R_2}\log_2(s_{2r_2}).$$

For a given penalty the output of FPOP is shift invariant. That is if the data is shifted by a given value the returned changepoints will be the same. Therefore the segmentation returned by DiffSegR does not depend on the knowledge of the normalization factors. This is a key difference with threshold based methods (e.g. srnadiff IR, srnadiff HMM, RNAprof, derfinder RL, derfinder SB).

We acknowledge that when taking into account the offset to the logarithms (+1) in the per-base $\log_2$-FC, the previous argument is approximately true for large counts but does not hold for small counts.

Overview of the DiffSegR package

DiffSegR is implemented as an R package (www.R-project.org/) and can be found on GitHub (https://aliehrmann.github.io/DiffSegR/index.html) with the installation procedure as well as a vignette with functional examples. The package implements the four steps of a conventional pipeline for identify-then-annotate methods (Figure 2).

Step 1: computing the coverage profiles and the differential transcription profile from BAMs

The *loadData* function builds coverage profiles from BAM files within a locus specified by the user. If the reads are stranded, the function builds one coverage profile per strand for each replicate of both compared biological conditions. By default the heuristic used to compute coverage profiles is the geometric mean of the 5′ and 3′ profiles (eq: 1). Alternative approaches use either the full length or the 5′ or 3′ end of the reads (Note S1). *loadData* then converts the coverage profiles into the per-base $\log_2$-FC (eq: 2) (one per strand) using the reference biological condition specified by the user as the denominator. The function returns the coverage profiles and the differential transcription profile as a list of run-length encoded objects.

Step 2: summarizing the differential transcription landscape

The *segmentation* function uses FPOP (28) on the per-base $\log_2$-FC of both strands to identify the segment's boundaries (or changepoints). The number of returned segments is controlled by the hyperparameter λ specified by the user. The segments are stored as GenomicRanges object and the *segmentation* function finally uses *featurecounts* (44) to assign them the mapped reads from each replicate of each biological condition. By default a read is allowed to be assigned to every segment it overlaps with. The segments and the associated count matrix are returned as a SummarizedExperiment object.

Step 3: differential expression analysis (DEA)

The *dea* function uses DESeq2 (29) to test the difference in average expression between the two compared biological conditions for every segment. The resulting *P*-values are then adjusted using a Benjamini-Hochberg (BH) procedure to control the false discovery rate (FDR), which is a common approach in DEA. However, this approach does not guarantee that the proportion of false discoveries (FDP) will be upper bounded, and there is no statistical guarantee on the number of false discoveries in subsets of segments selected using FDR thresholding. For example, while a widespread practice in DEA is to select a subset of segments whose absolute $\log_2$-FC passes a threshold it can potentially result in an inflated FDR. To address these limitations, the *dea* function can also call a post-hoc inference procedure that provides guarantees on the FDP in arbitrary segment selections (42). Finally, *dea* returns the user-provided SummarizedExperiment object augmented with the outcome of the DEA.

Step 4.A: annotating the differentially expressed regions (DERs)

The *annotateNearest* function annotates the DERs found during the DEA using user specified annotations in the gff3 or gtf format. Seven classes of labels translate the relative positions of the DER and its closest annotation(s): antisense, upstream, downstream, inside, overlapping 3′, overlapping 5′ and overlapping both 5′ and 3′. These labels allow users to easily understand the relationships between the DERs and their nearest annotations, and to analyze the potential functional significance of the DERs in the context of the annotated genomic features.

Step 4.B: visualizing the DERs

The *exportResults* function saves the DERs, not-DERs, segmentation, the mean of coverage profiles from both biological conditions and per-base $\log_2$-FC information, for both strands, in formats readable (bed, gff3) by genome viewers like the Integrative Genome Viewer (IGV) (30). For IGV, *exportResults* also creates a session in xml format that allows loading all tracks in one click. This provides a convenient way to save and visualize the results of the differential expression analysis, allowing a user-friendly exploration and interpretation of the data generated by the DEA. An example of the graphical output obtained with DiffSegR is displayed in Figure 3.

Benchmarking

Data and read mapping

The true positive rate (see *Evaluation metrics*) of DiffSegR and competitors were evaluated on two RNA-Seq datasets comparing *Arabidopsis thaliana* control plants (*col0*) to mutants deficient in the PNPase and Mini-III chloroplast ribonucleases (31,37). We refer to these datasets as *pnp1-1* and *rnc3/4*, respectively. In the *rnc3/4* dataset both conditions contained two replicates with about 19.5 million reads each while in the *pnp1-1* dataset, both conditions contained two replicates with about 18.6 million reads each. DiffSegR ability to work on a bacterial genome was evaluated using a RNA-Seq dataset

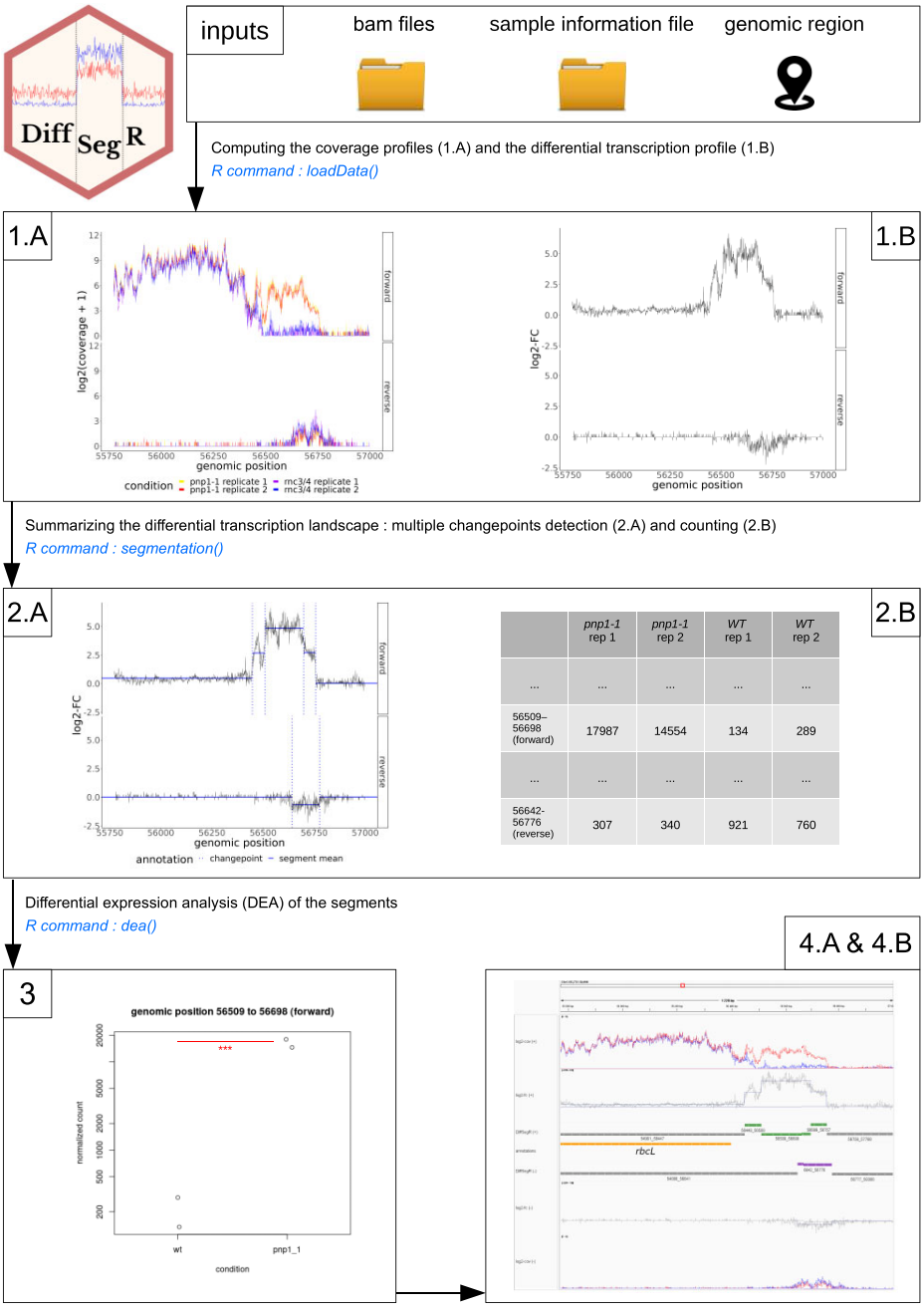


Figure 2. Schematic representation of the DiffSegR pipeline. The DiffSegR pipeline consists of four major steps: **(1)** Computing the coverage profiles and the differential transcription profile from BAMs. The *loadData* function creates coverage profiles from user-specified BAM files and a genomic region. (1.A) It produces one profile per strand for each replicate of both biological conditions. (1.B) The function then calculates the per-base $\log_2$ fold change ($\log_2$-FC) based on the coverage profiles. **(2)** Summarizing the differential transcription landscape. (2.A) The *segmentation* function applies FPOP to the per-base $\log_2$-FC of each strand to identify segment boundaries, known as changepoints. (2.B) Then the *featurecounts* program is used to assign mapped reads to segments, resulting in a count matrix. **(3)** Differential expression analysis (DEA). The *dea* function uses DESeq2 to test the difference in average expression between the two compared biological conditions for each segment. **(4)** Annotating and visualizing the differentially expressed regions (DERs). (4.A) The *annotateNearest* function annotates DERs using user-specified gff3 or gtf format annotations. In parallel, (4.B) the *exportResults* function saves DERs, not-DERs, segmentation, the mean of coverage profiles from both biological conditions, and per-base $\log_2$-FC information in formats compatible with genome viewers like IGV. An IGV session in XML format allows loading all tracks with one click, providing a user-friendly way to visualize and interpret DiffSegR results.

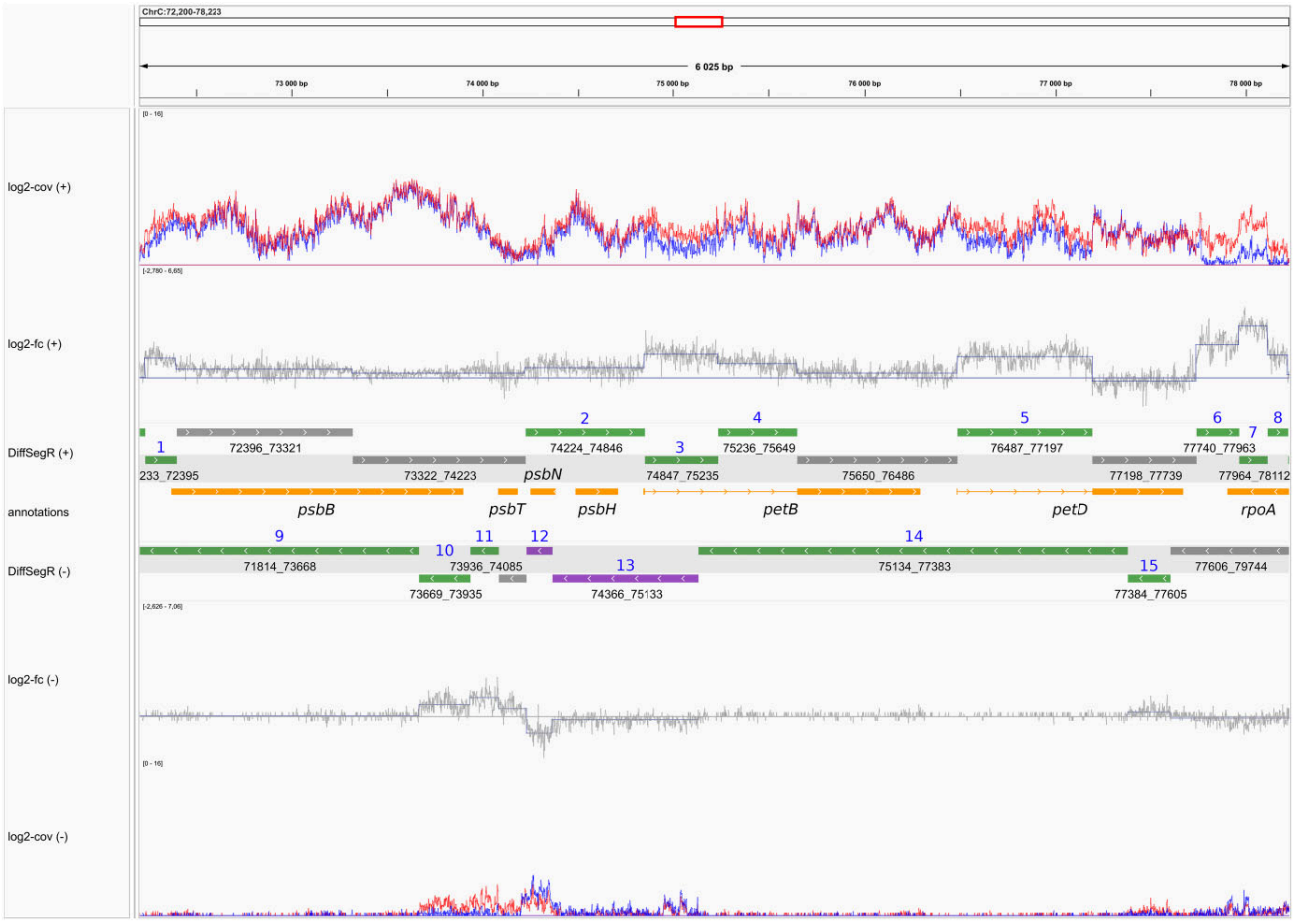


Figure 3. DiffSegR analysis of the *psbB-psbT-psbN-psbH-petB-petD* gene cluster in the *pnp1-1* dataset. The tracks from top to bottom represent: (log_2-Cov (+)) the mean of coverages on the log_2 scale for the forward strand in both biological conditions of interest, with the blue line representing the *WT* condition and the red line representing the *pnp1-1* condition; (log_2-FC (+)) the per-base log_2-FC between *pnp1-1* (numerator) and WT (denominator) for the forward strand. The straight horizontal line represents the zero indicator. When the per-base log_2-FC is above or below the zero indicator line, it suggests up-regulation or down-regulation, respectively, in *pnp1-1* compared to *WT*. The changepoint positions are indicated by vertical blue lines, and the mean of each segment is shown by horizontal blue lines connecting two changepoints; (DiffSegR (+)) the differential expression analysis results for segments identified by DiffSegR on the forward strand are presented as follows: up-regulated regions are depicted in green, down-regulated regions in purple, and non-differentially expressed regions in gray; (annotations) the genes provided by users for interpretations. Symmetrically, the remaining tracks correspond to the same data on the reverse strand. DiffSegR finds 8 up-regulated DERs on the forward strand (IDs 1 to 8), 5 up-regulated DERs on the reverse strand (IDs 9 to 11, 14 and 15), and 2 down-regulated DERs on the reverse strand (IDs 12 and 13). Table 1 provides a summary of the molecular validations published for the DERs identified in the *psbB* gene cluster through DiffSegR analysis. The bedGraph and gff3 files used to generate the tracks and the xml file used to load them in IGV were created using the *exportResults* function of the DiffSegR R package. The session was loaded in IGV 2.12.3 for Linux.

comparing a *Bacillus subtilis* control strain (CCB375 strain) to a mutant deficient for the Rae1 ribonuclease (SSB1002 strain) (38). We refer to this dataset as *Δrae1*. Both conditions contained three replicates with about 14.8 million reads each. The IDEAs dataset used to evaluate the false positive rate (see *Evaluation metrics*) contained ten RNA-Seq replicates of the Col-0 *Arabidopsis thaliana* accession grown in nitrogen deficiency condition with about 32.7 million reads each. The plants were grown at the IJPB Phenoscope platform (https://phenoscope.versailles.inrae.fr/) to ensure maximal homogeneity between the replicates (see the GEO database with the accession number GSE234377 for more details). RNA-Seq datasets were aligned to the *Arabidopsis thaliana* chloroplast genome using the OGE pipeline (https://forgemia.inra.fr/GNet/pipelineoge) (39). This pipeline uses the STAR aligner (40). The BAM files corresponding to the aligned *Bacillus subtilis* RNA-Seq dataset were kindly provided by Ciarán Condon. The alignment was performed using the Bowtie aligner (41). These alignments were then used for the downstream analyses. Because DiffSegR is the only evaluated method able to analyze stranded RNA-Seq reads, the BAM files were then split by strand in order to be used by the competing methods and the results for both strands were finally merged.

Adjusting method parameters

For the purpose of benchmarking DiffSegR against other methods in terms of true positive rate (see below), one or more parameters likely to change the number and/or the positions of the identified changepoints were adjusted.

1. The minimum depth threshold (*minDepth*) is common to derfinder RL and srnadiff. All contiguous positions with mean of coverages above this threshold are kept. For each method, on both datasets, one hundred

minDepth values evenly spaced within the interval [1,6000] were tested. The default *minDepth* value of derfinder RL and srnadiff are 5 and 10, respectively.
2. The minimum $\log_2$-FC threshold (*minLogFC*) is used by srnadiff to keep only contiguous positions with absolute normalized $\log_2$-FC above the threshold. For both methods, on both datasets, one hundred *minLogFC* values evenly spaced within the interval [0.1,7] were tested. The default *minLogFC* value of srnadiff is 0.5.
3. The emission threshold (*emissionThreshold*) is used by srnadiff to define the HMM states. For both methods, on both datasets, one hundred (*emissionThreshold*) values evenly spaced within the interval [0.09, 0.9] were tested. The default *emissionThreshold* value of srnadiff is 0.1.

For all these comparisons and on both datasets, the DiffSegR hyperparameter λ was kept to its default value, λ=2. In other analyses, all parameters from the different methods tested were set to their default values.

Evaluation metrics

At the end of the segmentation process, each method yields a collection of segments that may or may not correspond to genomic regions with differential expression. Differentially expressed regions (DERs) stand for the largest set of segments with a fold change >1.5 (symmetrically $< 2/3$) and a false discovery proportion upper bound set to 5%. Both per-segment fold change and *P*-value are estimated using DESeq2 (29). The post-hoc upper bound is obtained by controlling the joint error rate (JER) at significance level of 5% using the Simes family of thresholds implemented in the R package sanssouci (42,43). Unless specified, all methods were compared using these thresholds. All quality control of the DiffSegR results, including a PCA of counts, a dispersion-mean plot and an histogram of *P*-values are available in supplementary data for *pnp1-1* (Supplementary Figures S1-S3), *rnc3/4* (Supplementary Figures S4–S6) and *Δrae1* (Supplementary Figures S7–S9) datasets. For the comparisons on the *pnp1-1* and *rnc3/4* labeled dataset the error *E* was defined as the total number of labels which are not overlapped by at least one DER. A label is a genomic portion whose corresponding transcript has previously been validated by molecular biology techniques to be differentially accumulated in the mutant compared to WT. The genomic coordinates of the labels can be found in Supplementary Tables S1-S2. The true positive rate is given by $TPR = \frac{N-E}{N}$ where *N* is the total number of labels. In the blank experiment the replicates of the nitrogen deficiency condition from the IDEAs project were resampled in two groups to test several 2 versus 2, 3 versus 3, 4 versus 4 and 5 versus 5 designs. All the DERs identified are supposed to be false positives. The false positive rate is given by $FPR = \frac{\text{number of DERs}}{\text{number of segments}}$.

Results

Foreword

srnadiff merges the results of a two-level segmentation approach on the per-base *P*-value (srnadiff HMM) and a three-level segmentation approach on the per-base $\log_2$-FC (srnadiff IR) (Figure 1). Consequently, for the purposes of the following comparisons, we will use srnadiff as a representative tool of the methods following similar strategies, including derfinder SB and RNAprof. In addition, due to the lengthy process of estimating the parameters of the model implemented in parseq (days) (20), comparing this tool with srnadiff, derfinder RL and DiffSegR is beyond the scope of our study.

Speed and memory comparisons

All the simulations presented here were performed with an Intel Core i7-10810U CPU @ 1.10GHz, 16 Go of RAMs and 10 logical cores. On both chloroplast RNA-Seq datasets, DiffSegR returns results in less than 2 minutes. In comparison, it takes less than 30 s for a standard differential gene expression (DGE) analysis. The identification of segment boundaries using changepoint detection analysis runs in less than a second on both datasets. The slowest step of the DiffSegR pipeline is the construction of the coverage profiles followed later by the segment count table using the featureCounts program and the BAMs files (Supplementary Table S3). Less than 1 Go of RAM is necessary and the peak of memory used is reached at the differential analysis step (Supplementary Table S4).

DiffSegR facilitates the visualization of DERs

DiffSegR was applied to a RNA-Seq dataset comparing control plants (*col0*) to a mutant deficient in the PNPase chloroplast ribonuclease (*pnp1-1*), a major 3′ processing enzyme (37). When dealing with a gene dense genome like the plastome, annotating a DER using the nearest gene can lead to ambiguities. In this case, visualization of the DERs in a genome viewer, as exemplified for the *psbB-psbT-psbN-psbH-petB-petD* gene cluster (Figure 3), is often the simplest solution. In line with previous molecular studies, DiffSegR identifies 15 DERs, 8 on the forward and 7 on the reverse strand, respectively. For example, the overexpressed segment, in 5′ of the *psbB* gene (DER 1 with genomic positions 72233–72395) matches an area previously shown to over accumulate RNA 5′ ends in *pnp1-1* (45) and the segment 2 overlapping *psbH* and antisense to *psbN* (DER 2 with genomic positions 74224 to 74846) corresponds to various 400–700 nt long RNA isoforms previously characterized in *WT* or *pnp1-1* mutants (37,46–48). The published molecular validations corresponding to the DERs identified in the *psbB* gene cluster by DiffSegR are summarized in Table 1.

DiffSegR improves the search for DERs

The ability of DiffSegR and competitor methods derfinder and srnadiff (19,22) to identify DERs was evaluated on two RNA-Seq datasets generated for plants lacking the chloroplast ribonucleases PNPase (see above) and Mini-III (*rnc3/4*) (31,37). In comparison to control plants, these two mutants over accumulate RNA fragments that are mainly located outside of the annotated genic areas and the RNA-Seq data have been extensively validated using molecular techniques (31,32). These validations were used to define 23 labels (17 in *pnp1-1* and 6 in *rnc3/4*) where a DER was expected to be found (list and coordinates of the labels in Supplementary Tables S1-S2). Using its default segmentation hyperparameters (λ=2) DiffSegR identified 434 and 25 DERs in the *pnp1-1* and *rnc3/4* datasets respectively (Supplementary Tables S5-S6; Supplementary Figures S10–S30), including all the predefined labels (TPR = 1). By contrast, srnadiff and derfinder RL identified 16 and 4 labels out of 17 in *pnp1-1* and 4 and 0 labels out of 6 in *rnc3/4* (Table 2). After adjusting the per-base $\log_2$-FC threshold, only srnadiff was also able to reach a TPR of 1 (Supplementary Figure S31-S34). As expected, standard differential gene

Table 1. DERs identified by DiffSegR within the gene cluster *psbB-psbT-psbN-psbH-petB-petD* in *pnp1-1* dataset

Strand	Positions	DiffSegR result	Genomic context	ID	Validation
forward	72233–72395	up	*psbB* 5′ ends	1	(45)
forward	74224–74846	up	*psbH*; antisense to *psbN*	2	(37,46–48)
forward	74847–75235	up	*petB* intron	3	(47)
forward	75236–75649	up	*petB* intron	4	(47)
forward	76487–77196	up	*petD* intron	5	(47)
forward	77740–77963	up	*petD* 3′ ends; antisense to *petD-rpoA* intergenic	6	(37,47)
forward	77964–78112	up	*petD* 3′ ends; antisense to *rpoA*	7	(37,47)
forward	78113–78218	up	*petD* 3′ ends; antisense to *rpoA*	8	(37,47)
reverse	71814–73668	up	*psbN* 3′ ends; antisense to *psbB*	9	NA
reverse	73669–73935	up	*psbN* 3′ ends; antisense to *psbB*	10	NA
reverse	73936–74085	up	*psbN* 3′ ends; antisense to *psbB-psbT* intergenic	11	(37)
reverse	74232–74365	down	*psbN*	12	(47)
reverse	74366–75133	down	*psbN* 5′ ends; antisense to *psbH* and *petB*	13	(37)
reverse	75134–77383	up	*rpoA* 3′ ends; antisense to *petB* and *petD*	14	NA
reverse	77384–77605	up	*rpoA* 3′ ends; antisense to *petD*	15	NA

Most DERs are supported by molecular validations described in the literature. Up is for up-regulated and down for down-regulated.

Table 2. Comparison of the true positive rates (TPRs) for DiffSegR, srnadiff and derfinder RL methods on the *pnp1-1* (17 labels) and *rnc3/4* (6 labels) datasets

Dataset	Method	TPR
pnp1-1	DiffSegR	1 (17/17)
pnp1-1	srnadiff	0.94 (16/17)
pnp1-1	derfinder RL	0.24 (4/17)
rnc3/4	DiffSegR	1 (6/6)
rnc3/4	srnadiff	0.67 (4/6)
rnc3/4	derfinder RL	0 (0/6)

Each method is executed using its default segmentation hyperparameters.

expression (DGE) analysis, which relies on known gene annotations and is considered as a routine research tool (3), was unable to identify labels located outside of these annotations, therefore resulting in an TPR of 0. Because the large number of DERs found by DiffSegR could suggest it has a high FPR, we evaluated and compared it to classical DGE analysis (49) using a RNA-Seq dataset containing 10 replicates of the nitrogen deficiency condition. Any DER identified between subsamples of the replicates was therefore considered a false positive. The empirical cumulative distribution functions (eCDFs) of the FPR for both DiffSegR and the DGE analysis were similar when using the 5 versus 5 designs. For the 2 versus 2 designs, approximately 90% and 80% of the designs resulted in <2.5% of FPR with DiffSegR and traditional DGE respectively (Figure 4). These observations confirm that the FPR is not inflated in the results of DiffSegR (see Supplementary Figure S35 for 3 versus 3 and 4 versus 4 designs).

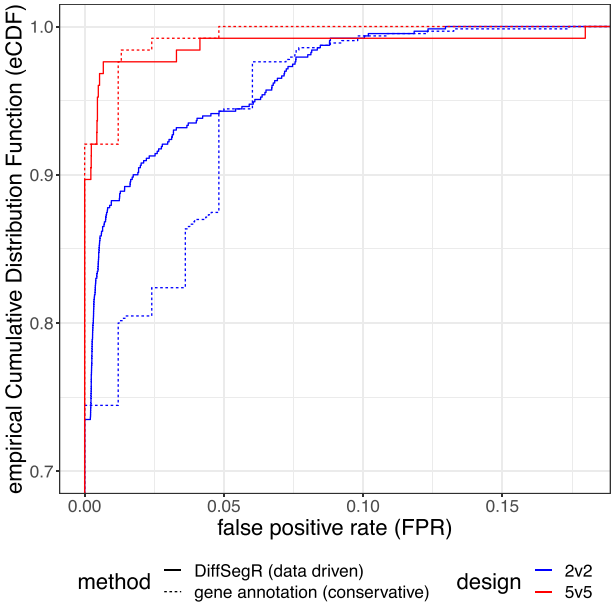


Figure 4. Comparison of the empirical cumulative distribution functions (eCDFs) of the False Positive Rate (FPR) from DiffSegR and the Differential Expression analysis within Gene annotations (DGE). The eCDFs of FPRs from DiffSegR (solid curves) and DGE (dashed curves) methods are compared by re-sampling two groups from 10 biological replicates of the same nitrogen deficiency condition in the IDEAs dataset. The figure displays results for group sizes of 2 (blue curves) and 5 (red curves) (see Supplementary Figure S35 for 3v3 and 4v4 designs). The eCDF represents the proportion of comparisons (y-axis) with fewer false positives than a specified percentage (x-axis). The eCDF analysis demonstrates that the FPR in DiffSegR results is not inflated compared to the widely-used DGE approach.

DiffSegR better captures the differential landscape

Because derfinder RL and srnadiff use a two- or three-level segmentation model they are susceptible to merge in a single DER various contiguous segments having different $\log_2$-FC. As a consequence, distinct DER events stemming from distinct RNA maturation processes could be wrongly associated together (Note S4 and Supplementary Figure S43). In contrast, DiffsegR segments the mean of the per-base $\log_2$-FC without making any assumption on the number of levels. It should therefore be able to distinguish between contiguous DER events, leading to shorter DER than the other methods. We therefore compared the length distribution of DERs identified by DiffSegR, srnadiff and derfinder RL. In agreement with our expectation, the DERs identified by DiffSegR are on average smaller than those identified by its competitors in both the *pnp1-1* and *rnc3/4* datasets (Figure 5). Respective median sizes are equal to 211 and 455 nt for DiffSegR and srnadiff (P-value $< 2.2 \times 10^{-16}$, Mann–Whitney U test) in *pnp1-1*. In *rnc3/4* respective median lengths are equal to 15 and 97 nt (P-value $= 0.0362$, Mann–Whitney U test) (Figure 5A). An identical trend can be observed between DiffSegR and

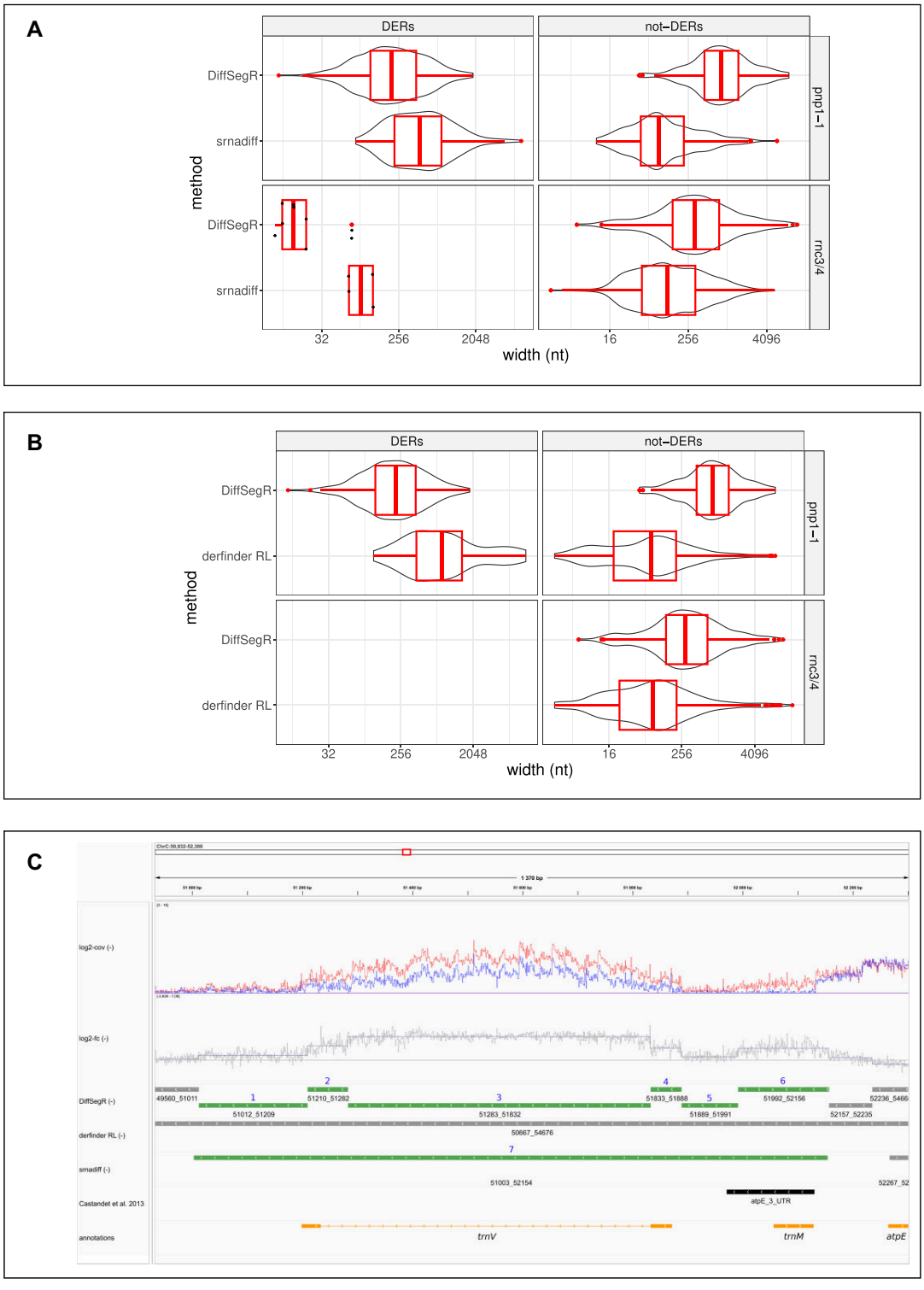


Figure 5. Comparisons of DERs and not-DERs lengths between DiffSegR, derfinder RL and srnadiff on *pnp1-1* and *rnc3/4* datasets. **(A)** The length distribution of DERs and not-DERs identified by DiffSegR and srnadiff are shown using both boxplot and violin plot. Only overlapping (not-)DERs between the compared methods are kept. A (not-)DER of method DiffSegR is considered overlapping either if it covers 90% of a (not-)DER of srnadiff or if 90% of it is covered by a DER of method srnadiff. When there are fewer than 20 overlapping DERs or not-DERs, the violin plot is replaced by a dot plot. **(B)** Similar comparisons were made between DiffSegR and derfinder RL methods. Derfinder does not identify DERs in *rnc3/4*, which explains the lack of overlap between DiffSegR DERs and derfinder RL DERs in this dataset. **(A, B)** In both datasets, DiffSegR not-DERs are on average longer than srnadiff not-DERs and derfinder RL not-DERs. Additionally, DiffSegR DERs are on average smaller compared to srnadiff DERs and derfinder RL DERs (Mann-Whitney U test). **(C)** Comparison of DiffSegR, derfinder RL, and srnadiff analyses for the *trnV* gene and the 3′ ends of *atpE*, located on the reverse strand of the chloroplast genome. The tracks are defined as depicted in Figure 3, and further enhanced by incorporating the results from the derfinder RL and srnadiff analyses. DiffSegR identifies 6 up-regulated DERs (IDs 1–6). derfinder RL fails to detect any DERs within this region. Lastly, srnadiff discovers a singular DER (ID 7).

derfinder RL. In *pnp1-1* respective median sizes are equal to 220 and 826 nt (P-value $< 2.2 \times 10^{-16}$, Mann–Whitney U test). In *rnc3/4*, derfinder fails to detect DERs, accounting for the absence of overlapping DERs between DiffSegR and derfinder RL in this particular dataset (Figure 5.B). We conclude that srnadiff and derfinder RL indeed merge neighboring DERs with different $\log_2$-FC.

Moreover, derfinder RL directly segments the mean of coverages and is therefore susceptible to split regions that are not differentially expressed into distinct segments (Note S5 and Supplementary Figure S44). This is because the shape of the transcriptional signal is strongly influenced by numerous biological and technical factors that are not directly related to *bona fide* transcriptional differences (50). In contrast, DiffSegR uses the per-base $\log_2$-FC that is largely unaffected by the underlying transcriptional coverage. This is because local variations in coverage are reproducible and cancel out when taking the difference of the $\log_2(\log_2\text{-FC})$ (Supplementary Figure S36). As a consequence, we expect DiffSegR to return not-DER longer than derfinder RL. We therefore compared the length distribution of not-DERs identified by DiffSegR, srnadiff and derfinder RL in both *pnp1-1* and *rnc3/4* datasets. Figure 5 shows that the not-DERs identified by DiffSegR are indeed on average longer than those identified by its competitors. Respective median sizes are equal to 833 and 80 nt for DiffSegR and srnadiff (P-value $< 2.2 \times 10^{-16}$, Mann–Whitney U test) in *pnp1-1*. In *rnc3/4* respective median lengths are equal to 294 and 86 nt (P-value $< 2.2 \times 10^{-16}$, Mann–Whitney U test) (Figure 5A). An identical trend can be observed between DiffSegR and derfinder RL. In *pnp1-1* respective median sizes are equal to 833 and 80 nt (P-value $< 2.2 \times 10^{-16}$, Mann–Whitney U test). In the *rnc3/4* dataset, respective median lengths are equal to 327 and 122 nt (P-value $< 2.2 \times 10^{-16}$, Mann–Whitney U test) (Figure 5B). We conclude that both srnadiff and derfinder RL over-segment regions that are not differentially expressed in comparison to DiffSegR.

DiffSegR can be used on sparser genomes

Sparsity refers to the fraction of a genomic region with a null RNA-Seq coverage and is known to cause artifacts in statistical analyses (51). Because the two plant chloroplasts RNA-Seq datasets previously used have a low sparsity ranging from 0.42 to 0.57 we tested DiffSegR on a *Bacillus subtilis* RNA-Seq dataset previously used to decipher the role of the Rae1 ribonuclease (38) and whose sparsity ranged from 0.79 to 0.82 between the different replicates. Using standard differential expression analysis, Leroy *et al.* identified 46 mRNAs and ncRNAs as significantly up-regulated in the *rae1* mutant (q-value < 0.05 & fold change > 1.5) and eventually selected seven of them (*S1025*, *S1024*, *S1026*, *yrzI*, *bmrC*, *bmrD*, *bglC*) as candidates for direct degradation by Rae1. DiffSegR returned significant up-regulated DERs overlapping 45 of the 46 genes identified by Leroy *et al.* including the seven candidates of interests (Supplementary Figures S37–S39). In addition, DiffSegR returned significantly up-regulated DERs overlapping 60 other genes (Supplementary Tables S7 and S8). A striking feature was however the over-representation of very short DERs. The five most abundant ones were indeed 4 (6.5%), 6 (6.4%), 5 (5.9%), 2 (5.6%) and 8 (5.4%) nt long while the five most abundant ones in the *pnp1-1* dataset were 55 (1.7%), 73 (1.7%), 83 (1.1%), 204 (1.1%), 56 (0.8%) nt long.

Discussion

DiffSegR is a straightforward solution to the DERs detection problem

We here introduced DiffSegR, an R package that allows the discovery of transcriptome-wide expression differences between two biological conditions using RNA-Seq data (Figure 2). While standard RNA-Seq differential analyses rely on reference gene annotations and therefore miss potentially meaningful DERs, DiffSegR directly identifies the boundaries of DERs without requiring any annotation. Unlike its competitors, DiffSegR is designed to analyze stranded RNA-Seq reads, therefore allowing the identification of transcriptional differences on both the forward and reverse strands. This is an invaluable asset when considering the pervasiveness of antisense transcripts (52–54). The output generated by DiffSegR can be easily loaded into the Integrative Genomics Viewer (IGV), providing a user-friendly platform for the exploration and interpretation of the results (Figure 3).

Like other methods willing to automatically identify transcription differences along the genome, DiffSegR addresses a well-defined statistical problem known as the multiple changepoints detection or segmentation problem. Among the many algorithmically and statistically well-established methods that have been developed to tackle this problem (55,56), DiffSegR uses FPOP (28). This method relies on a Gaussian model to detect changes in the mean of a signal. The computation time of FPOP is log-linear in the signal length, making it time efficient (Supplementary Table S3). FPOP is statistically grounded (33,57), and has been shown to be effective in numerous simulations (28,55) and genomic applications (26,27,58). Another advantage of FPOP is that it only has one parameter (the penalty), therefore simplifying calibration and interpretation.

A key feature of DiffSegR is the use of the per-base $\log_2$-FC signal for segmentation analysis, a strategy that carries three main advantages. First, it scales with the intensity of the difference up to a normalization constant. Second, it discriminates between up-regulated and down-regulated DERs and third, it is largely insensitive to local variations in coverage as they are reproducible (Supplementary Figure S36) and cancel out when taking the difference of the logs ($\log_2$-FC). Moreover, in contrast to the two-level (DER and not-DER or expressed and not-expressed) and three-level (up-regulated DER, down-regulated DER, not-DER) segmentation models used by other approaches (Figure 1), FPOP does not make any assumptions on the number of levels in the $\log_2$-FC and can effectively distinguish between adjacent DERs that involves distinct RNA maturation processes. As a consequence DiffSegR detects fewer changes in non-differential regions but detects more segments in DERs than its competitors (Figure 5). This suggests that DiffSegR is able to effectively summarize the data, providing a detailed and accurate representation of the differential landscape while being more selective in its analysis of not-DERs.

DiffSegR accurately captures the differential landscape

DiffSegR finds all the extended 3′ and 5′ ends of transcripts, as well as accumulated antisense RNA, in RNA-Seq labeled datasets *pnp1-1* and *rnc3/4*. These labels were previously verified through molecular techniques, and DiffSegR was able to identify them with its default settings, while none of the competitors tested were able to do so. However, the use of the

same dataset twice in DiffSegR (and its competitors), a procedure so-called double dipping, first for segmentation and then for differential analysis may result in an inflated false positive rate (59–61). We therefore verified that the FPR of DiffSegR is similar to standard DGE analysis using a blank experiment (Figure 4). A possible explanation to the observed robustness is the fact that DiffSegR uses different aspects of the data in its two steps: while the segmentation uses the per-base $\log_2$-FC, the DEA relies on normalized counts, per-segment $\log_2$-FC, and dispersion. The last three parameters are estimated by DESeq2.

We are therefore confident that the numerous DERs identified outside of the predefined labels in the two chloroplastic RNA-Seq datasets represent *bona fide* DERs. For example, 387 out of the 434 DERs identified in the *pnp1-1* RNA-Seq experiment did not overlap labels. While an exhaustive molecular validation of these 387 segments is beyond the scope of this study, numerous evidences suggest they are accurate. Specifically, DiffSegR identifies 72 DERs overlapping all the 25 plastid introns in the PNPase mutant, a feature previously shown to reflect a lack of intron degradation following splicing in the mutant (47). Neither srnadiff nor derfinder RL were able to capture this feature entirely. Another example suggesting that DiffSegR does not over-segment the differential transcription profile is displayed for genomic area 51012–52156 in Figure 5.C. While it is not differentially expressed according to derfinder RL, srnadiff considers it as a single DER (DER 7 with genomic positions 51003–52154) and DiffSegR identifies 6 contiguous different DERs within it. The multiplicity of DERs identified by DiffSegR seems to better reflect the shape of the $\log_2$-FC and is also consistent with the known roles of the PNPase in transcript 3′ end maturation (DER 1 with genomic positions 51012–51209 and DER 6 with genomic positions 51992–52156 for *trnV* and *atpE*, respectively) or the degradation of tRNA 5′ precursor (DER 5 with genomic positions 51889–51991 for *trnV*) (32). Finally, both *trnV* exons over accumulate (DERs 2 and 4 with genomic positions 51210–51282 and 51833–51888, respectively) in the mutant, along with the corresponding intron (DER 3 with genomic positions 51283–51832). The segmentation in three different DERs is, once again, an accurate interpretation of the two different biological mechanisms targeting tRNAs and introns in the mutant (47,62).

Larger genomes with more zeroes

DiffSegR is also effective and powerful on genomes larger and more complex than the chloroplast. It effectively identified the two RNA locations that have been shown to be degraded by the Rae1 endoribonuclease in *Bacillus subtilis* (38,63). This illustrates one of the big advantages of DiffSegR, it can be easily used to narrow down the number of genomic regions worth investigating. From the 4.2 Gb *Bacillus* genome it identified 1833 regions (Supplementary Table S7) that contained the two known cleavage sites, a number that is compatible with the workforce of most research teams. It is however true that the segmentation model used by DiffSegR may result in an over segmentation in profiles containing many base pairs with a null coverage. This could be problematic when addressing even larger genomes, like nuclear ones, and prevent interpretability of the results.

A straightforward solution would be to apply DiffSegR to smaller portions of the genome, only keeping the ones with sufficient coverage. This however comes with issues of its own as (i) identifying those genomic portions is a segmentation problem itself, multiplying the genomic areas complexifies selection, and (ii) this leads to a *triple-dipping* problem as the data is used three times (identification of the genomic area, segmentation within the genomic area and differential expression analysis). Alternative strategies would be to integrate more advanced segmentation methods already available. More specifically, we believe it could be useful to (i) weight the base pair according to its coverage (using a weighted version of FPOP, (64)), (ii) consider full length reads at the prize of modeling auto-correlation (65), and (iii) model the discrete nature of the data using a negative binomial model (66).

Conclusion

In conclusion, DiffSegR is a powerful tool that provides researchers with a systematic and accurate way to discover expression differences between two conditions using RNA-Seq data, without the need for prior annotations. Because it is designed to compare two conditions, we believe that DiffSegR has the potential to change the way researchers approach differential expression analysis, especially considering the wealth of RNA-Seq based strategies aimed at capturing specific events (67). For example, it has already been used on RNA immunoprecipitation sequencing data to study translation initiation in plant mitochondria (68). We anticipate it could similarly be used to find newly transcribed RNAs compared to mature RNA control in nascent RNA analysis (69), to find differences in ribosome bound RNA in translatome analysis (70) or to discriminate structured (double-stranded RNA) from unstructured RNAs in structurome analysis (71). We expect that the use of DiffSegR will lead to new discoveries and insights in the field of transcriptomic.

Data availability

Software availability

The latest version of the DIffSegR R package is available at https://aliehrmann.github.io/DiffSegR/index.html and https://zenodo.org/doi/10.5281/zenodo.10017833. The package includes a Vignette which shows on a minimal example how to use the main functions.

Data availability

- Raw sequences for the *rnc3/4* dataset have been retrieved from the BioProject database with the accession number PRJNA268035.
- Raw sequences for the *pnp1-1* dataset have been retrieved from the SRA database with the accession number SRA046998.
- Raw sequences for the nitrogen deficiency condition from the IDEAs dataset are available at GEO database with the accession number GSE234377.
- Raw sequences for the *Δrae1* dataset can be accessed from the GEO database with the number GSE93894.

Reproducibility

The scripts used to generate the figures/tables from this manuscript and figures/tables from the Supplementary Materials are available at https://github.com/aLiehrmann/DiffSegR_paper and https://zenodo.org/doi/10.5281/zenodo.10017833.

Supplementary data

Supplementary Data are available at NARGAB Online.

Acknowledgements

The authors would like to thank Ciarán Condon for providing the sequencing data about the *B.subtilis Δrae1* mutant and extensive discussions about the analyses. This work has benefited from the support of IJPB's Plant Observatory technological platforms. We also thank Amber Hotto for help proofreading the manuscript.

Funding

Agence Nationale de la Recherche [ANR-20-CE20-0004 JOAQUIN to B.C.]; the IDEAS experiment was funded by an ATIGE grant from Génopole; A.L. was supported by a PhD fellowship from the French ministère de l'enseignement supérieur et de la recherche; the IPS2 and IJPB benefit from the support of Saclay Plant Sciences-SPS [ANR-17-EUR-0007].

Conflict of interest statement

None declared.

References

1. Gilbert,W. (1978) Why genes in pieces?*Nature*, **271**, 501.
2. Djebali,S., Davis,C.A., Merkel,A., Dobin,A., Lassmann,T., Mortazavi,A., Tanzer,A., Lagarde,J., Lin,W., Schlesinger,F., *et al.* (2012) Landscape of transcription in human cells. *Nature*, **489**, 101–108.
3. Stark,R., Grzelak,M. and Hadfield,J. (2019) RNA sequencing: the teenage years. *Nat. Rev. Genet.*, **20**, 631–656.
4. Mendes Soares,L.M. and Valcárcel,J. (2006) The expanding transcriptome: the genome as the 'Book of Sand'. *EMBO J.*, **25**, 923–931.
5. Morillon,A. and Gautheret,D. (2019) Bridging the gap between reference and real transcriptomes. *Genome Biol.*, **20**, 112.
6. Whiffin,N., Karczewski,K.J., Zhang,X., Chothani,S., Smith,M.J., Evans,D.G., Roberts,A.M., Quaife,N.M., Schafer,S., Rackham,O., *et al.* (2020) Characterising the loss-of-function impact of 5′ untranslated region variants in 15,708 individuals. *Nat. Commun.*, **11**, 2523.
7. Griesemer,D., Xue,J.R., Reilly,S.K., Ulirsch,J.C., Kukreja,K., Davis,J.R., Kanai,M., Yang,D.K., Butts,J.C., Guney,M.H., *et al.* (2021) Genome-wide functional screen of 3'UTR variants uncovers causal variants for human disease and evolution. *Cell*, **184**, 5247–5260.
8. Chan,J.J., Tabatabaeian,H. and Tay,Y. (2022) 3′UTR heterogeneity and cancer progression. *Trends Cell Biol.*, **33**, 568–582.
9. Zhang,Y., Liu,L., Qiu,Q., Zhou,Q., Ding,J., Lu,Y. and Liu,P. (2021) Alternative polyadenylation: methods, mechanism, function, and role in cancer. *J. Exp. Clin. Cancer Res.*, **40**, 51.
10. Rhoads,A. and Au,K.F. (2015) PacBio Sequencing and its applications. *Genomics. Proteomics Bioinformatics*, **13**, 278–289.
11. Wang,Y., Zhao,Y., Bollas,A., Wang,Y. and Au,K.F. (2021) Nanopore sequencing technology, bioinformatics and applications. *Nat. Biotechnol.*, **39**, 1348–1365.
12. Weirather,J.L., de Cesare,M., Wang,Y., Piazza,P., Sebastiano,V., Wang,X.-J., Buck,D. and Au,K.F. (2017) Comprehensive comparison of Pacific Biosciences and Oxford Nanopore Technologies and their applications to transcriptome analysis. *F1000Research*, **6**, 100.
13. Steijger,T., Abril,J.F., Engström,P.G., Kokocinski,F., Hubbard,T.J., Guigó,R., Harrow,J. and Bertone,P. (2013) Assessment of transcript reconstruction methods for RNA-seq. *Nat. Methods*, **10**, 1177–1184.
14. Mehmood,A., Laiho,A., Venäläinen,M.S., McGlinchey,A.J., Wang,N. and Elo,L.L. (2020) Systematic evaluation of differential splicing tools for RNA-seq studies. *Brief. Bioinform.*, **21**, 2052–2065.
15. Zhang,R., Kuo,R., Coulter,M., Calixto,C.P.G., Entizne,J.C., Guo,W., Marquez,Y., Milne,L., Riegler,S., Matsui,A., *et al.* (2022) A high-resolution single-molecule sequencing-based arabidopsis transcriptome using novel methods of iso-seq analysis. *Genome Biol.*, **23**, 149.
16. Nellore,A., Jaffe,A.E., Fortin,J.-P., Alquicira-Hernández,J., Collado-Torres,L., Wang,S., Phillips,R.A. III, Karbhari,N., Hansen,K.D., Langmead,B., *et al.* (2016) Human splicing diversity and the extent of unannotated splice junctions across human RNA-seq samples on the Sequence Read Archive. *Genome Biol.*, **17**, 266.
17. Deveson,I.W., Brunck,M.E., Blackburn,J., Tseng,E., Hon,T., Clark,T.A., Clark,M.B., Crawford,J., Dinger,M.E., Nielsen,L.K., *et al.* (2018) Universal alternative splicing of noncoding exons. *Cell Syst.*, **6**, 245–255.
18. Frazee,A.C., Sabunciyan,S., Hansen,K.D., Irizarry,R.A. and Leek,J.T. (2014) Differential expression analysis of RNA-seq data at single-base resolution. *Biostatistics*, **15**, 413–426.
19. Zytnicki,M. and González,I. (2021) Finding differentially expressed sRNA-seq regions with srnadiff. *PLoS One*, **16**, e0256196.
20. Mirauta,B., Nicolas,P. and Richard,H. (2014) Parseq: reconstruction of microbial transcription landscape from RNA-seq read counts using state-space models. *Bioinformatics*, **30**, 1409–1416.
21. Tran,V.D.T., Souiai,O., Romero-Barrios,N., Crespi,M. and Gautheret,D. (2016) Detection of generic differential RNA processing events from RNA-seq data. *RNA Biol*, **13**, 59–67.
22. Collado-Torres,L., Nellore,A., Frazee,A.C., Wilks,C., Love,M.I., Langmead,B., Irizarry,R.A., Leek,J.T. and Jaffe,A.E. (2017) Flexible expressed region analysis for RNA-seq with derfinder. *Nucleic. Acids. Res.*, **45**, e9.
23. Willenbrock,H. and Fridlyand,J. (2005) A comparison study: applying segmentation to array CGH data for downstream analyses. *Bioinformatics*, **21**, 4084–4091.
24. Picard,F., Robin,S., Lebarbier,E. and Daudin,J.-J. (2007) A segmentation/clustering model for the analysis of array CGH data. *Biometrics*, **63**, 758–766.
25. Hocking,T.D., Rigaill,G. and Bourque,G. (2015) PeakSeg: constrained optimal segmentation and supervised penalty learning for peak detection in count data. In: *32nd International Conference on Machine Learning, ICML 2015*. PMLR, Vol. **1**, pp. 324–332.
26. Liehrmann,A., Rigaill,G. and Hocking,T.D. (2021) Increased peak detection accuracy in over-dispersed ChIP-seq data with supervised segmentation models. *BMC Bioinf.*, **22**, 323.
27. Hocking,T.D., Rigaill,G., Fearnhead,P. and Bourque,G. (2020) Constrained dynamic programming and supervised penalty learning algorithms for peak detection in genomic data. *J. Mach. Learn. Res.*, **21**, 1–40.
28. Maidstone,R., Hocking,T., Rigaill,G. and Fearnhead,P. (2017) On optimal multiple changepoint algorithms for large data. *Stat. Comput.*, **27**, 519–533.
29. Love,M.I., Huber,W. and Anders,S. (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.*, **15**, 550.
30. Robinson,J.T., Thorvaldsdóttir,H., Winckler,W., Guttman,M., Lander,E.S., Getz,G. and Mesirov,J.P. (2011) Integrative genomics viewer. *Nat. Biotechnol.*, **29**, 24–26.
31. Hotto,A.M., Castandet,B., Gilet,L., Higdon,A., Condon,C. and Stern,D.B. (2015) Arabidopsis chloroplast mini-ribonuclease III participates in rRNA maturation and intron recycling. *Plant Cell*, **27**, 724–740.

32. Castandet,B., Hotto,A.M., Fei,Z. and Stern,D.B. (2013) Strand-specific RNA sequencing uncovers chloroplast ribonuclease functions. *FEBS Lett.*, **587**, 3096–3101.
33. Yao,Y.-C. and Au,S.T. (1989) Least-squares estimation of a step function. *Sankhyā Indian J. Stat. Ser. A*, **51**, 370–381.
34. Fearnhead,P. and Rigaill,G. (2020) Relating and comparing methods for detecting changes in mean. *Stat*, **9**, e291.
35. Robinson,M.D., McCarthy,D.J. and Smyth,G.K. (2009) edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**, 139–140.
36. Law,C.W., Chen,Y., Shi,W. and Smyth,G.K. (2014) voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol.*, **15**, R29.
37. Hotto,A.M., Schmitz,R.J., Fei,Z., Ecker,J.R. and Stern,D.B. (2011) Unexpected diversity of chloroplast noncoding RNAs as revealed by deep sequencing of the Arabidopsis transcriptome. *G3 Genes Genomes Genetics*, **1**, 559–570.
38. Leroy,M., Piton,J., Gilet,L., Pellegrini,O., Proux,C., Coppée,J., Figaro,S. and Condon,C. (2017) Rae1/YacP, a new endoribonuclease involved in ribosome-dependent mRNA decay in *Bacillus subtilis*. *EMBO J.*, **36**, 1167–1181.
39. Baudry,K., Delannoy,E. and Colas des Francs-Small,C. (2022) Analysis of the plant mitochondrial transcriptome. *Methods Mol. Biol.*, **2363**, 235–262.
40. Dobin,A., Davis,C.A., Schlesinger,F., Drenkow,J., Zaleski,C., Jha,S., Batut,P., Chaisson,M. and Gingeras,T.R. (2013) STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, **29**, 15–21.
41. Langmead,B., Trapnell,C., Pop,M. and Salzberg,S.L. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.*, **10**, R25.
42. Blanchard,G., Neuvial,P. and Roquain,E. (2020) Post hoc confidence bounds on false positives using reference families. *Ann. Stat.*, **48**, 1281–1303.
43. Neuvial,P., Blanchard,G., Durand,G., Roquain,E. and Enjalbert-Courrech,N. (2022) sanssouci: post hoc multiple testing inference. R package version 0.12.8, https://sanssouci-org.github.io/sanssouci/index.html(30 October 2023, date last accessed).
44. Liao,Y., Smyth,G.K. and Shi,W. (2014) featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics*, **30**, 923–930.
45. Castandet,B., Germain,A., Hotto,A.M. and Stern,D.B. (2019) Systematic sequencing of chloroplast transcript termini from Arabidopsis thaliana reveals >200 transcription initiation sites and the extensive imprints of RNA-binding proteins and secondary structures. *Nucleic Acids Res.*, **47**, 11889–11905.
46. Felder,S., Meierhoff,K., Sane,A.P., Meurer,J., Driemel,C., Plücken,H., Klaff,P., Stein,B., Bechtold,N. and Westhoff,P. (2001) The nucleus-encoded HCF107 gene of Arabidopsis provides a link between intercistronic RNA processing and the accumulation of translation-competent psbH transcripts in chloroplasts. *Plant Cell*, **13**, 2127–2141.
47. Germain,A., Herlich,S., Larom,S., Kim,S.H., Schuster,G. and Stern,D.B. (2011) Mutational analysis of Arabidopsis chloroplast polynucleotide phosphorylase reveals roles for both RNase PH core domains in polyadenylation, RNA 3′-end maturation and intron degradation. *Plant J.*, **67**, 381–394.
48. Guilcher,M., Liehrmann,A., Seyman,C., Blein,T., Rigaill,G., Castandet,B. and Delannoy,E. (2021) Full length transcriptome highlights the coordination of plastid transcript processing. *Int. J. Mol. Sci.*, **22**, 11297.
49. Van den Berge,K., Hembach,K.M., Soneson,C., Tiberi,S., Clement,L., Love,M.I., Patro,R. and Robinson,M.D. (2019) RNA sequencing data: hitchhiker's guide to expression analysis. *Annu. Rev. Biomed. Data Sci.*, **2**, 139–173.
50. Lahens,N.F., Kavakli,I.H., Zhang,R., Hayer,K., Black,M.B., Dueck,H., Pizarro,A., Kim,J., Irizarry,R., Thomas,R.S., *et al.* (2014) IVT-seq reveals extreme bias in RNA sequencing. *Genome Biol.*, **15**, R86.
51. Silverman,J.D., Roche,K., Mukherjee,S. and David,L.A. (2020) Naught all zeros in sequence count data are the same. *Comput. Struct. Biotechnol. J.*, **18**, 2789–2798.
52. Reis,R.S. and Poirier,Y. (2021) Making sense of the natural antisense transcript puzzle. *Trends Plant Sci.*, **26**, 1104–1115.
53. Tan-Wong,S.M., Dhir,S. and Proudfoot,N.J. (2019) R-loops promote antisense transcription across the mammalian genome. *Mol. Cell*, **76**, 600–616.
54. Wade,J.T. and Grainger,D.C. (2014) Pervasive transcription: illuminating the dark matter of bacterial transcriptomes. *Nat. Rev. Microbiol.*, **12**, 647–653.
55. Fearnhead,P. and Rigaill,G. (2019) Changepoint detection in the presence of outliers. *J. Am. Stat. Assoc.*, **114**, 169–183.
56. Truong,C., Oudre,L. and Vayatis,N. (2020) Selective review of offline change point detection methods. *Signal Process.*, **167**, 107299.
57. Garreau,D. and Arlot,S. (2018) Consistent change-point detection with kernels. *Electron. J. Stat.*, **12**, 4440–4486.
58. Hocking,T.D., Rigaill,G., Fearnhead,P. and Bourque,G. (2022) Generalized functional pruning optimal partitioning (GFPOP) for constrained changepoint detection in genomic data. *J. Stat. Softw.*, **101**, 1–31.
59. Gao,L.L., Bien,J. and Witten,D. (2022) Selective inference for hierarchical clustering. *J. Am. Stat. Assoc.*, https://doi.org/10.1080/01621459.2022.2116331.
60. Neufeld,A.C., Gao,L.L. and Witten,D.M. (2022) Tree-values: selective inference for regression trees. *J. Mach. Learn. Res.*, **23**, 1–43.
61. Zhao,S., Witten,D. and Shojaie,A. (2021) In defense of the indefensible: a very naïve approach to high-dimensional inference. *Stat. Sci.*, **36**, 562–577.
62. Walter,M., Kilian,J. and Kudla,J. (2002) PNPase activity determines the efficiency of mRNA 3′-end processing, the degradation of tRNA and the extent of polyadenylation in chloroplasts. *EMBO J.*, **21**, 6905–6914.
63. Deves,V., Trinquier,A., Gilet,L., Alharake,J., Condon,C. and Braun,F. (2023) Shut down of multidrug transporter bmrCD mRNA expression mediated by the ribosome associated endoribonuclease Rae1 cleavage in a new cryptic ORF. *RNA*, **29**, 1108–1116.
64. Rigaill,G. (2022) fpopw: weighted segmentation using functional pruning and optimal partioning. https://cran.r-project.org/package=fpopw(30 October 2023, date last accessed).
65. Romano,G., Rigaill,G., Runge,V. and Fearnhead,P. (2022) Detecting abrupt changes in the presence of local fluctuations and autocorrelated noise. *J. Am. Stat. Assoc.*, **117**, 2147–2162.
66. Runge,V., Hocking,T.D., Romano,G., Afghah,F., Fearnhead,P. and Rigaill,G. (2023) gfpop: an R package for univariate graph-constrained change-point detection. *J. Stat. Softw.*, **106**, 1–39.
67. Reuter,J.A., Spacek,D.V. and Snyder,M.P. (2015) High-throughput sequencing technologies. *Mol. Cell*, **58**, 586–597.
68. Tran,H.C., Schmitt,V., Lama,S., Wang,C., Launay-Avon,A., Bernfur,K., Sultan,K., Khan,K., Brunaud,V., Liehrmann,A., *et al.* (2023) An mTRAN-mRNA interaction mediates mitochondrial translation initiation in plants. *Science*, **381**, eadg0995.
69. Wissink,E.M., Vihervaara,A., Tippens,N.D. and Lis,J.T. (2019) Nascent RNA analyses: tracking transcription and its regulation. *Nat. Rev. Genet.*, **20**, 705–723.
70. Calviello,L. and Ohler,U. (2017) Beyond read-counts: ribo-seq data analysis to understand the functions of the transcriptome. *Trends Genet.*, **33**, 728–744.
71. Kertesz,M., Wan,Y., Mazor,E., Rinn,J.L., Nutter,R.C., Chang,H.Y. and Segal,E. (2010) Genome-wide measurement of RNA secondary structure in yeast. *Nature*, **467**, 103–107.

Received: June 22, 2023. **Revised:** September 27, 2023. **Editorial Decision:** October 13, 2023. **Accepted:** October 23, 2023

Supplementary Materials for DiffSegR: An RNA-Seq data driven method for differential expression analysis using changepoint detection

Arnaud Liehrmann [*1,2,3], Etienne Delannoy [1,2], Alexandra Launay-Avon [1,2], Elodie Gilbault [4], Olivier Loudet [4], Benoît Castandet [*1,2] and Guillem Rigaill [*1,2,3]

[1] Institute of Plant Sciences Paris-Saclay (IPS2), Université Paris-Saclay, CNRS, INRAE, Université Evry, Gif sur Yvette, 91190, France

[2] Institute of Plant Sciences Paris-Saclay (IPS2), Université Paris Cité, CNRS, INRAE, Gif sur Yvette, 91190, France

[3] Laboratoire de Mathématiques et de Modélisation d'Evry (LaMME), Université d'Evry-Val-d'Essonne, UMR CNRS 8071, ENSIIE, USC INRAE, Evry, 91037, France

[4] Université Paris-Saclay, INRAE, AgroParisTech, Institut Jean-Pierre Bourgin (IJPB), 78000, Versailles, France

* To whom correspondence should be addressed

Arnaud Liehrmann arnaud.liehrmann@universite-paris-saclay.fr

Benoît Castandet benoit.castandet@universite-paris-saclay.fr

Guillem Rigaill guillem.rigaill@inrae.fr

The supplementary file includes:

Figures S1 to S39;

Notes S1 to S5 containing the supplementary Table S9 and the supplementary Figures S40 to S44;

Supplementary Figures S1 to S39

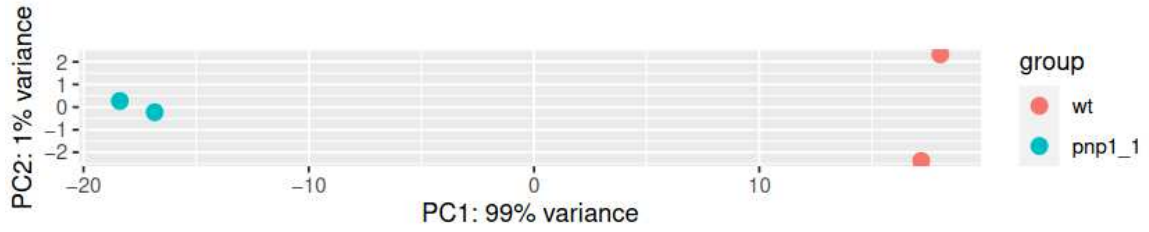


Figure S1: PCA of transformed counts from *pnp1-1* RNA-Seq experiment analyzed with DiffSegR. The biological replicates cluster well by condition on PC1.

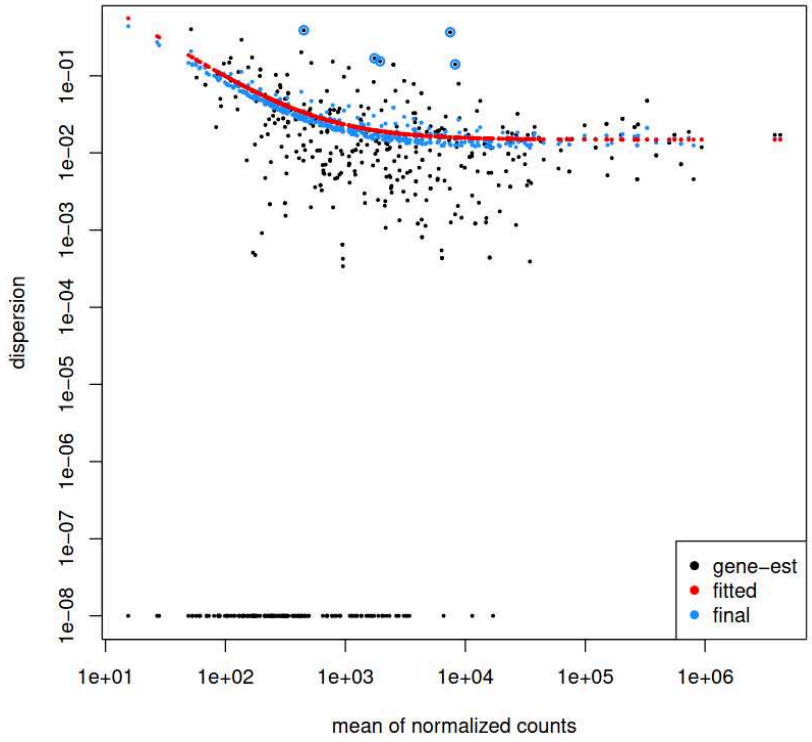


Figure S2: Dispersion-mean plot from *pnp1-1* RNA-Seq experiment analyzed with DiffSegR. The plot shows a characteristic dispersion-mean trend for RNA-Seq data.

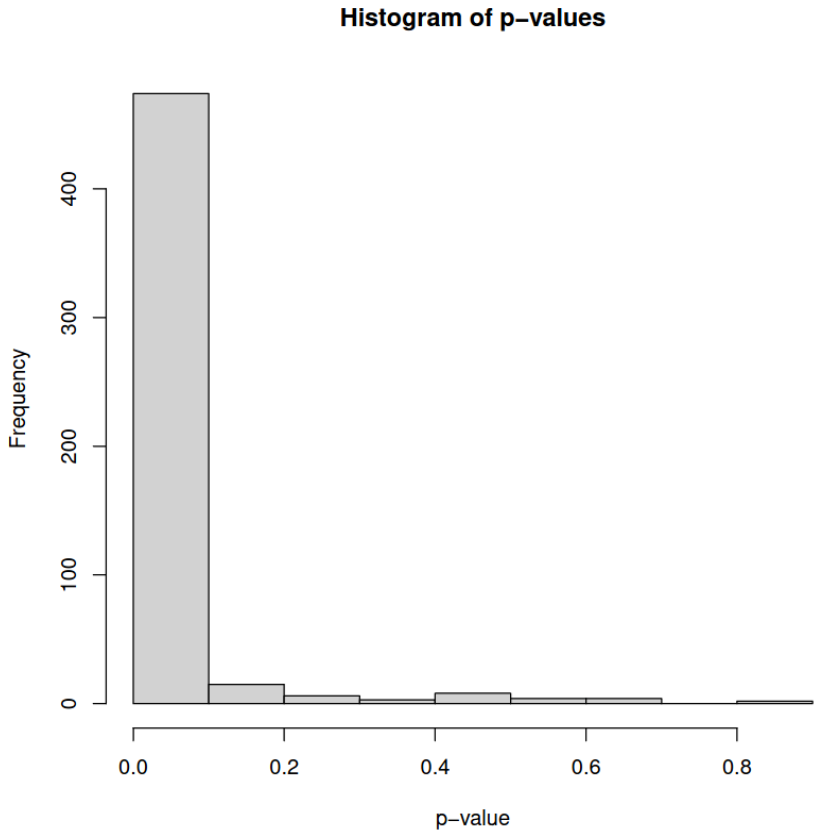


Figure S3: Histogram of p-values from *pnp1-1* RNA-Seq experiment analyzed with DiffSegR. The histogram does not show oddity.

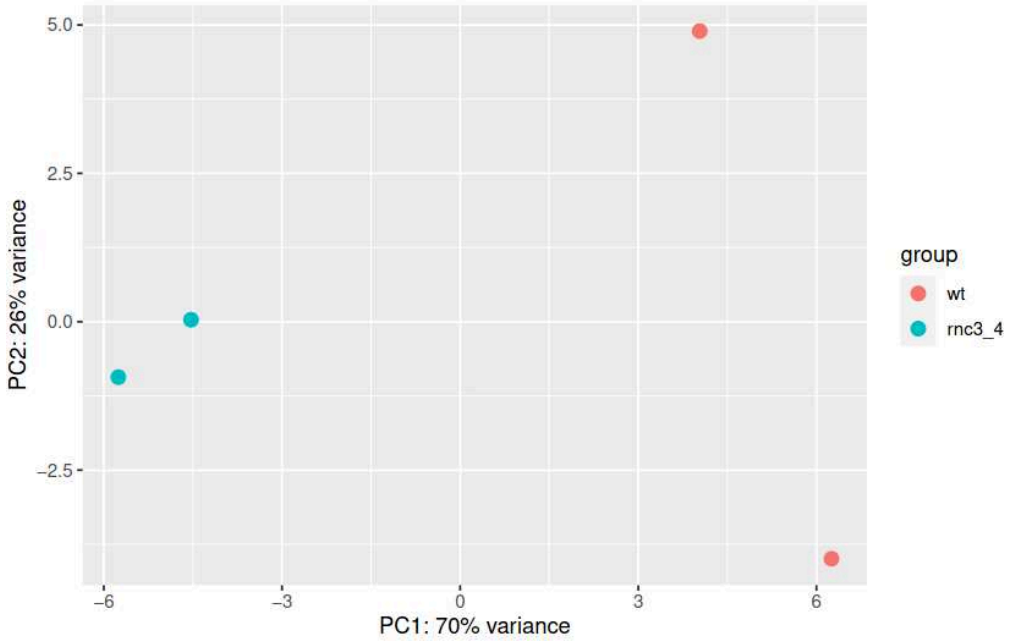


Figure S4: PCA of transformed counts from *rnc3/4* RNA-Seq experiment analyzed with DiffSegR. The biological replicates cluster well by condition on PC1.

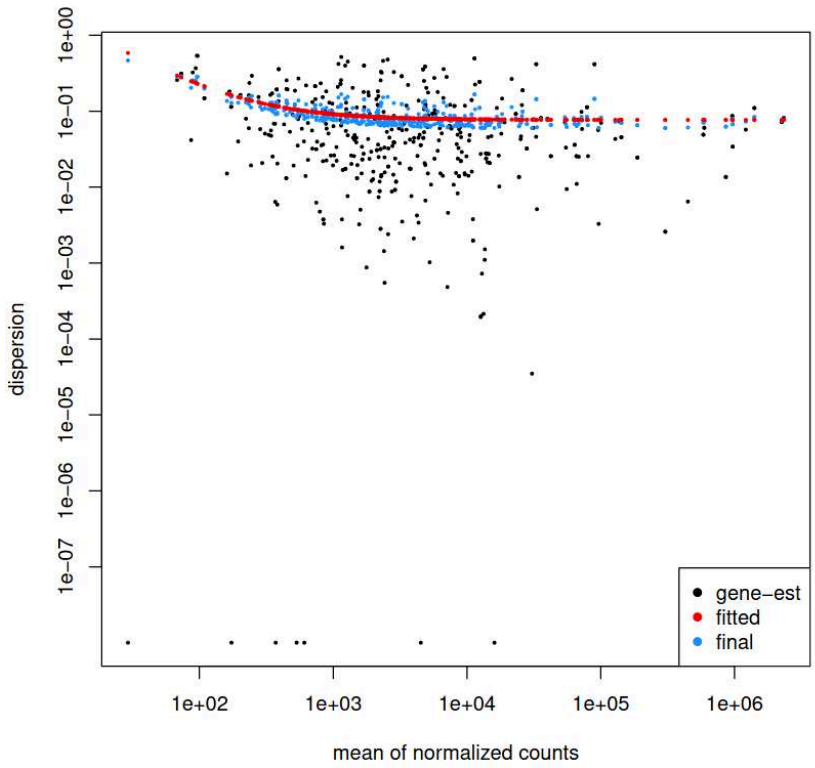


Figure S5: Dispersion-mean plot from *rnc3/4* RNA-Seq experiment analyzed with DiffSegR. The plot shows a characteristic dispersion-mean trend for RNA-Seq data.

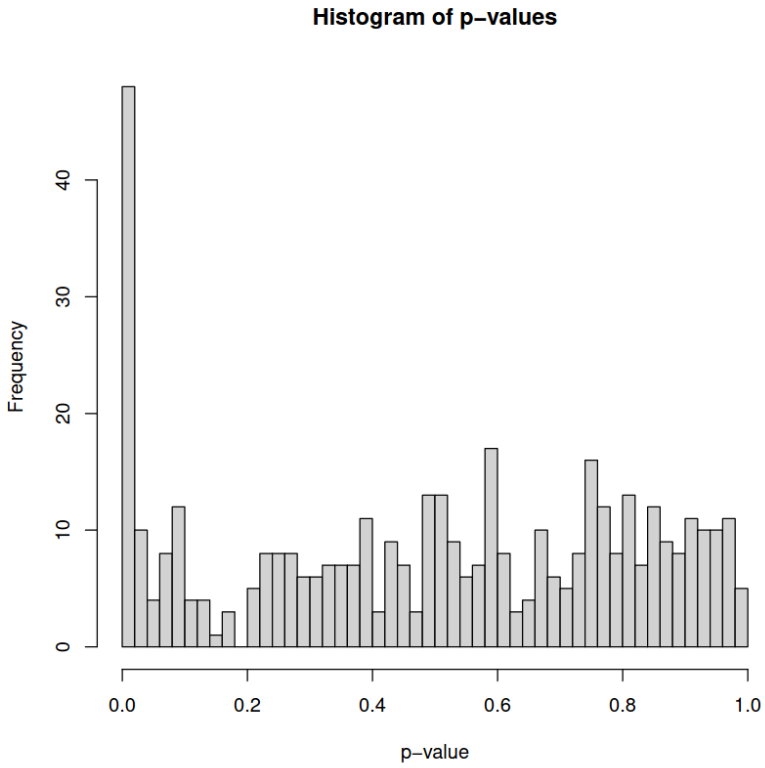


Figure S6: Histogram of p-values from *rnc3/4* RNA-Seq experiment analyzed with DiffSegR. The histogram does not show oddity.

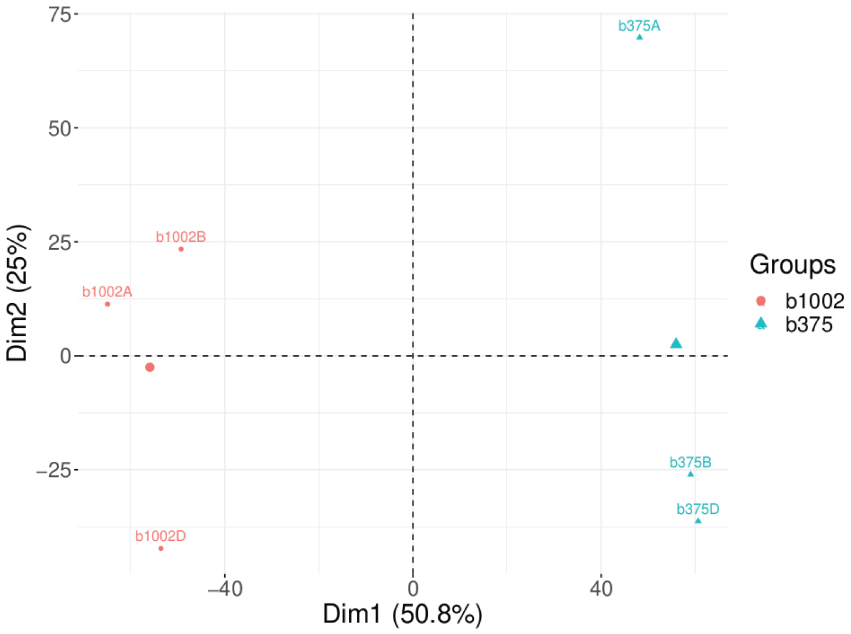


Figure S7: PCA of transformed counts from Δ*rae1* RNA-Seq experiment analyzed with DiffSegR. The biological replicates cluster well by condition on Dim1.

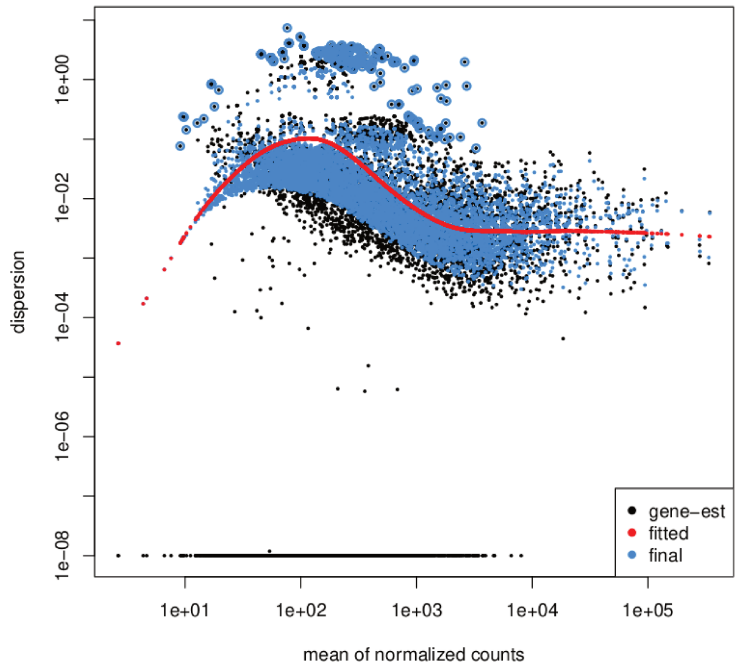


Figure S8: Dispersion-mean plot from △*rae1* RNA-Seq experiment analyzed with DiffSegR. The dispersion of small counts is negatively biased. Small counts can be filtered out during the preprocessing step.

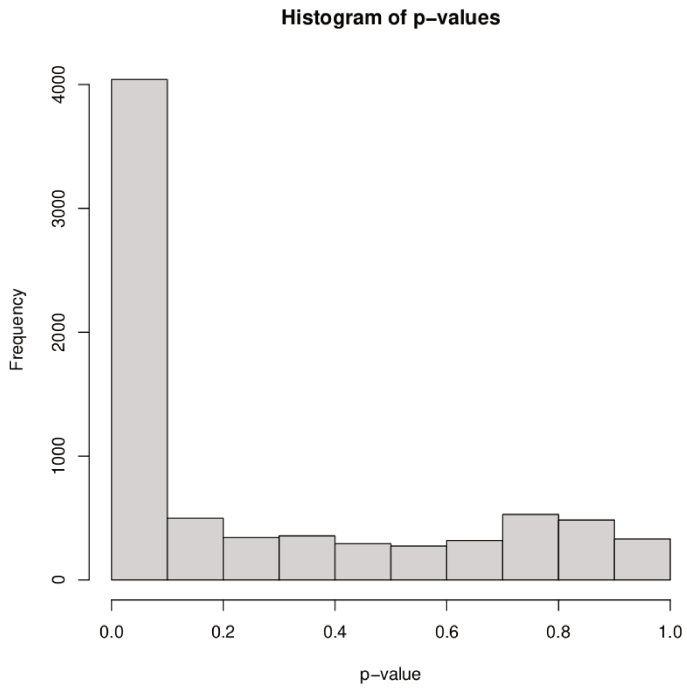


Figure S9: Histogram of p-values from △*rae1* RNA-Seq experiment analyzed with DiffSegR. The histogram does not show oddity.

Figure S10: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 1 to 714 on the reverse strand in the *pnp1-1* dataset. The tracks from top to bottom represent: (log2-Cov (-)) the mean of coverages on the log2 scale for the reverse strand in both biological conditions of interest, with the blue line representing the WT condition and the red line representing the *pnp1-1* condition; (log2-FC (-)) the per-base log2-FC between *pnp1-1* (numerator) and WT (denominator) for the reverse strand. The straight horizontal line represents the zero indicator. When the per-base log2-FC is above or below the zero indicator line, it suggests up-regulation or down-regulation, respectively, in *pnp1-1* compared to WT. The changepoint positions are indicated by vertical blue lines, and the mean of each segment is shown by horizontal blue lines connecting two changepoints; (DiffSegR (-)) the differential expression analysis results for segments identified by DiffSegR on the reverse strand are presented as follows: up-regulated regions are depicted in green, down-regulated regions in purple, and non-differentially expressed regions (non-DERs) in gray. (derfinder RL (-)) the derfinder RL results in the same format as the previous track; (srnadiff (-)) the srnadiff results in the same format as the previous track. (Castandet et al. 2013) the labels of differentially accumulated RNAs in *pnp1-1* compared to WT based on molecular biology validations described in Castandet et al. 2013; (annotations) the genes annotations. The bedGraph and gff3 files used to generate the tracks and the xml file used to load them in IGV were created using the *exportResults* function of the DiffSegR R package. The session was loaded in IGV 2.12.3 for Linux.

Figure S11: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 27,146 to 27,446 on the forward strand in the *pnp1-1* dataset. The tracks are similar to those described in Figure S10.

Figure S12: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 34,994 to 36,500 on the forward strand in the *pnp1-1* dataset. The tracks are similar to those described in Figure S10.

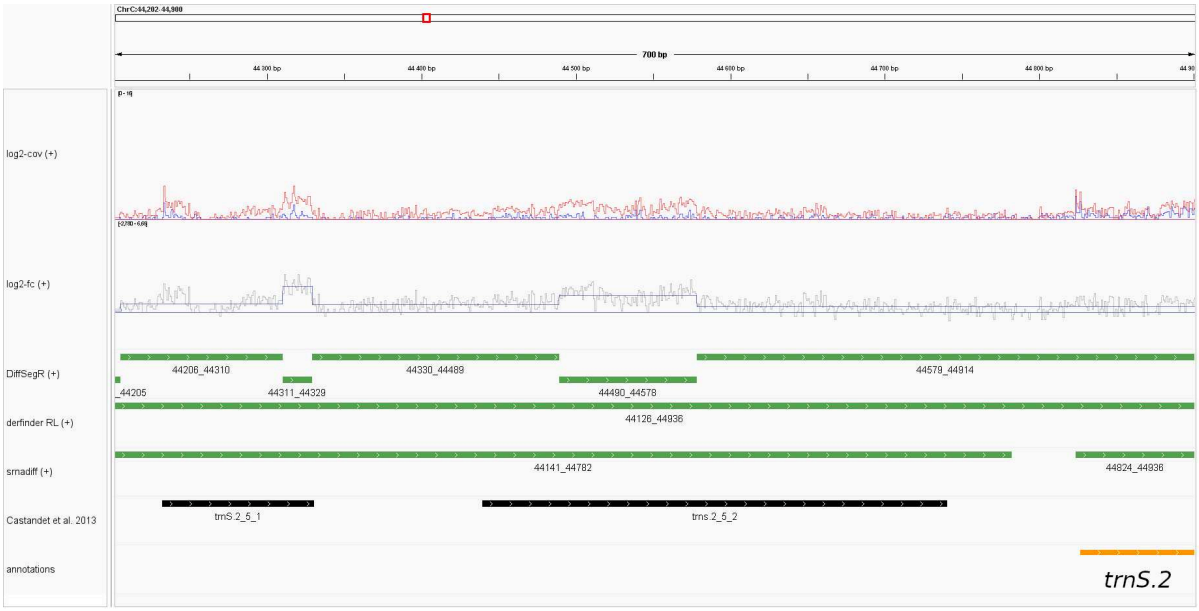


Figure S13: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 44,202 to 44,900 on the forward strand in the *pnp1-1* dataset. The tracks are similar to those described in Figure S10.

Figure S14: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 47,520 to 48,370 on the forward strand in the *pnp1-1* dataset. The tracks are similar to those described in Figure S10.

Figure S15: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 48,406 to 48,762 on the reverse strand in the *pnp1-1* dataset. The tracks are similar to those described in Figure S10.

Figure S16: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 51,958 to 52,350 on the reverse strand in the *pnp1-1* dataset. The tracks are similar to those described in Figure S10.

Figure S17: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 56,300 to 56,800 on the forward strand in the *pnp1-1* dataset. The tracks are similar to those described in Figure S10.

Figure S18: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 62,476 to 63,189 on the forward strand in the *pnp1-1* dataset. The tracks are similar to those described in Figure S10.

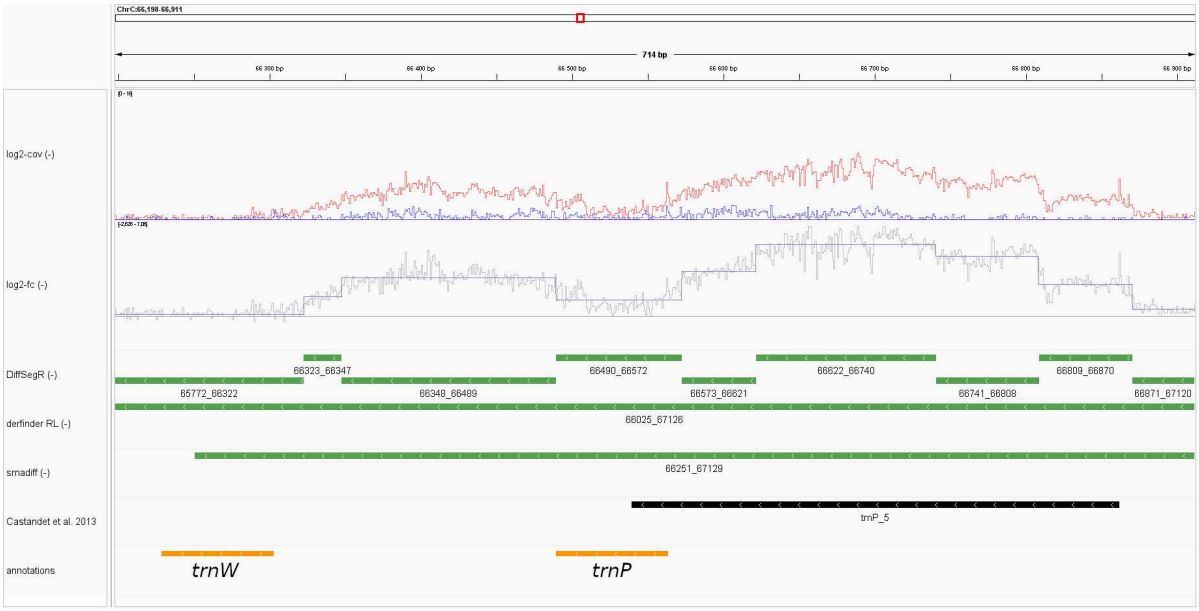


Figure S19: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 66,198 to 66,911 on the reverse strand in the *pnp1-1* dataset. The tracks are similar to those described in Figure S10.

Figure S20: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 77,589 to 77,990 on the forward strand in the *pnp1-1* dataset. The tracks are similar to those described in Figure S10.

Figure S21: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 94,236 to 95,089 on the reverse strand in the *pnp1-1* dataset. The tracks are similar to those described in Figure S10.

Figure S22: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 107,213 to 107,926 on the forward strand in the *pnp1-1* dataset. The tracks are similar to those described in Figure S10.

Figure S23: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 108,978 to 109,691 on the reverse strand in the *pnp1-1* dataset. The tracks are similar to those described in Figure S10.

Figure S24: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 115,261 to 115,974 on the reverse strand in the *pnp1-1* dataset. The tracks are similar to those described in Figure S10.

Figure S25: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 104,561 to 104,917 on the forward strand in the *rnc3/4* dataset. The tracks are similar to those described in Figure S10.

Figure S26: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 107,549 to 107,905 on the forward strand in the *rnc3/4* dataset. The tracks are similar to those described in Figure S10.

Figure S27: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 107,587 to 107,943 on the reverse strand in the *rnc3/4* dataset. The tracks are similar to those described in Figure S10.

Figure S28: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 130,736 to 131,092 on the forward strand in the *rnc3/4* dataset. The tracks are similar to those described in Figure S10.

Figure S29: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 130,742 to 131,098 on the reverse strand in the *rnc3/4* dataset. The tracks are similar to those described in Figure S10.

Figure S30: Comparison of DiffSegR, derfinder RL, and srnadiff analyses of chloroplast genomic positions 133,869 to 134,046 on the reverse strand in the *rnc3/4* dataset. The tracks are similar to those described in Figure S10.

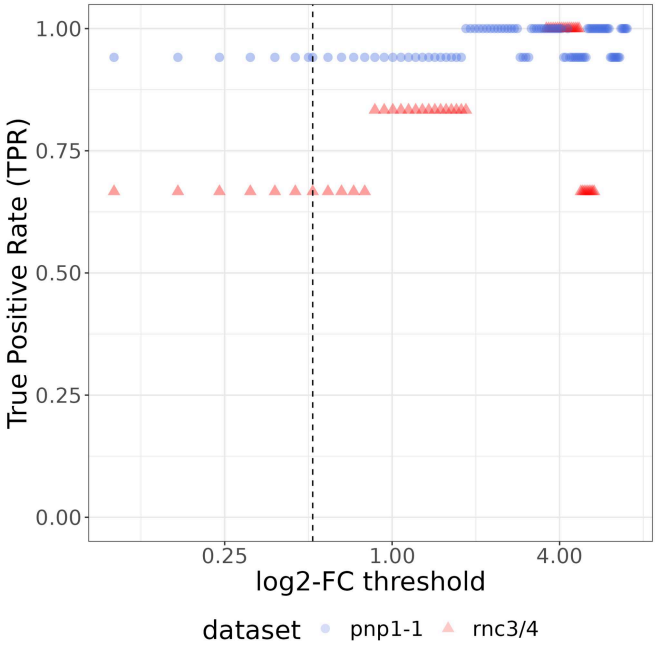


Figure S31 : The true positive rate (TPR) of srnadiff on *rnc3/4* and *pnp1-1* labeled datasets as a function of user-defined log2-FC threshold. The black vertical line represents the default log2-FC threshold value (0.5).

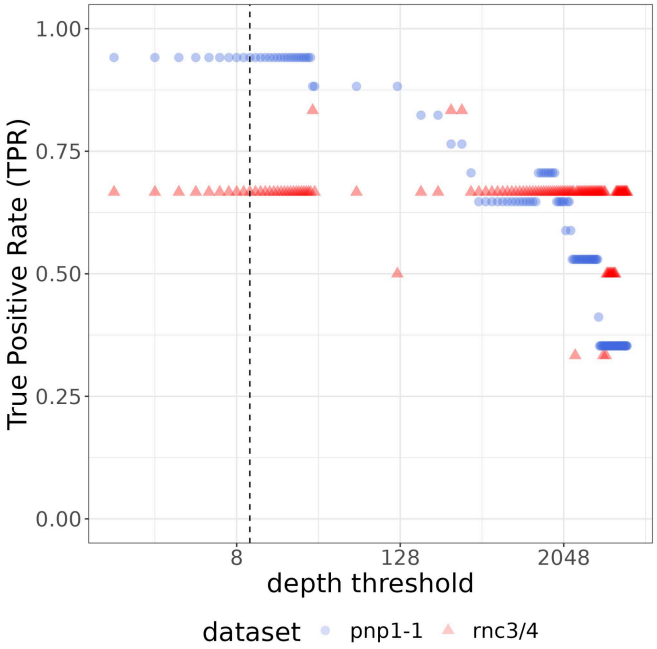


Figure S32 : The true positive rate (TPR) of srnadiff on *rnc3/4* and *pnp1-1* labeled datasets as a function of user-defined depth threshold. The black vertical line represents the default depth threshold value (10).

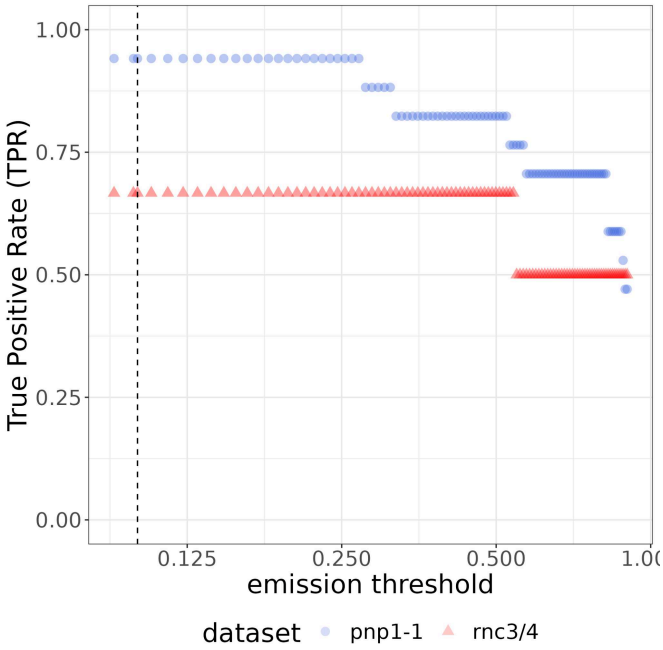


Figure S33 : The true positive rate (TPR) of srnadiff on *rnc3/4* and *pnp1-1* labeled datasets as a function of user-defined emission threshold. The black vertical line represents the default emission threshold value (0.1).

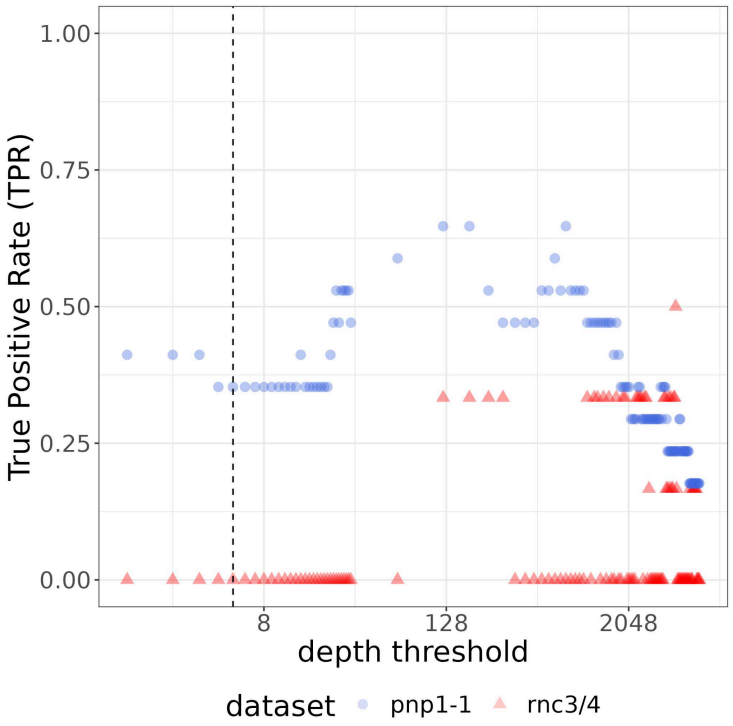


Figure S34 : The true positive rate (TPR) of derfinder RL on *rnc3/4* and *pnp1-1* labeled datasets as a function of user-defined depth threshold. The black vertical line represents the default depth threshold value (5).

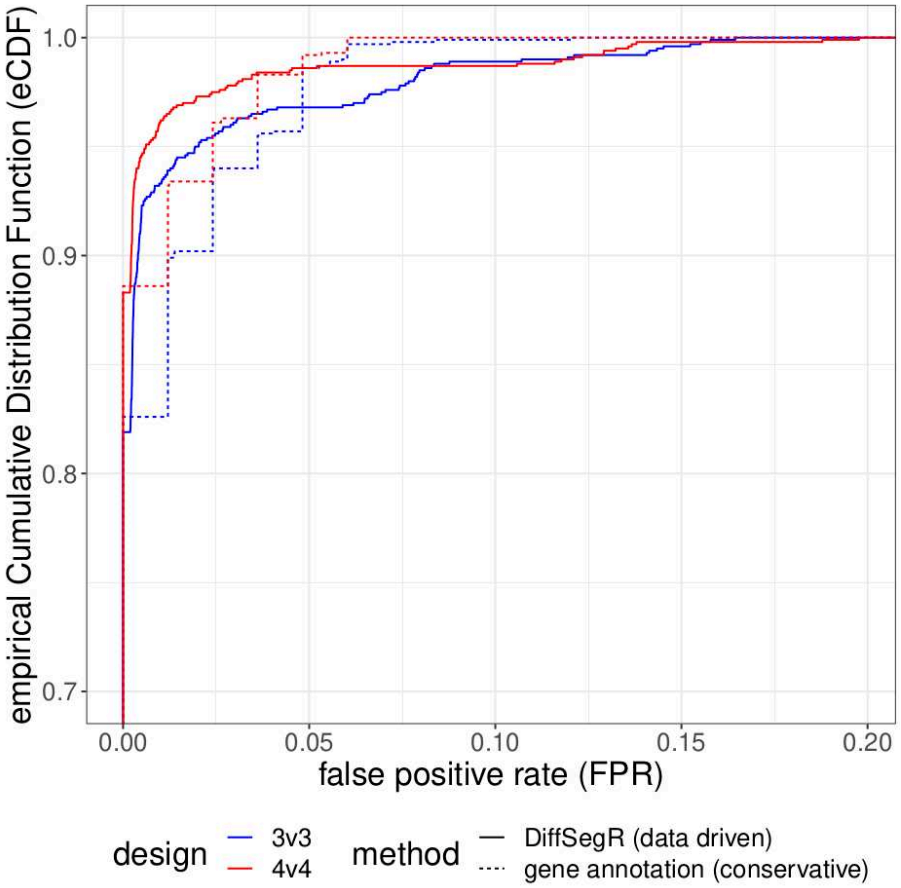


Figure S35: Comparison of the empirical cumulative distribution functions (eCDFs) of the False Positive Rate (FPR) from DiffSegR and the Differential Expression analysis within Gene annotations (DGE). The eCDFs of FPRs from DiffSegR (solid curves) and DGE (dashed curves) methods are compared by re-sampling two groups from 10 biological replicates of the same nitrogen deficiency condition in the IDEAs dataset. The figure displays results for group sizes of 3 (blue curves) and 4 (red curves). The eCDF represents the proportion of comparisons (y-axis) with fewer false positives than a specified percentage (x-axis). The eCDF analysis demonstrates that the FPR in DiffSegR results is not inflated compared to the widely-used DGE approach.

Figure S36 : Overview of coverage profiles overlapping *psbA* gene in *pnp1-1* dataset. Coverage profiles exhibit local variations highly reproducible from one sample to another. Local variations could be caused by technical factors that shape the coverage, e.g. 5'/3' bias, PCR bias, GC bias, non-random priming.

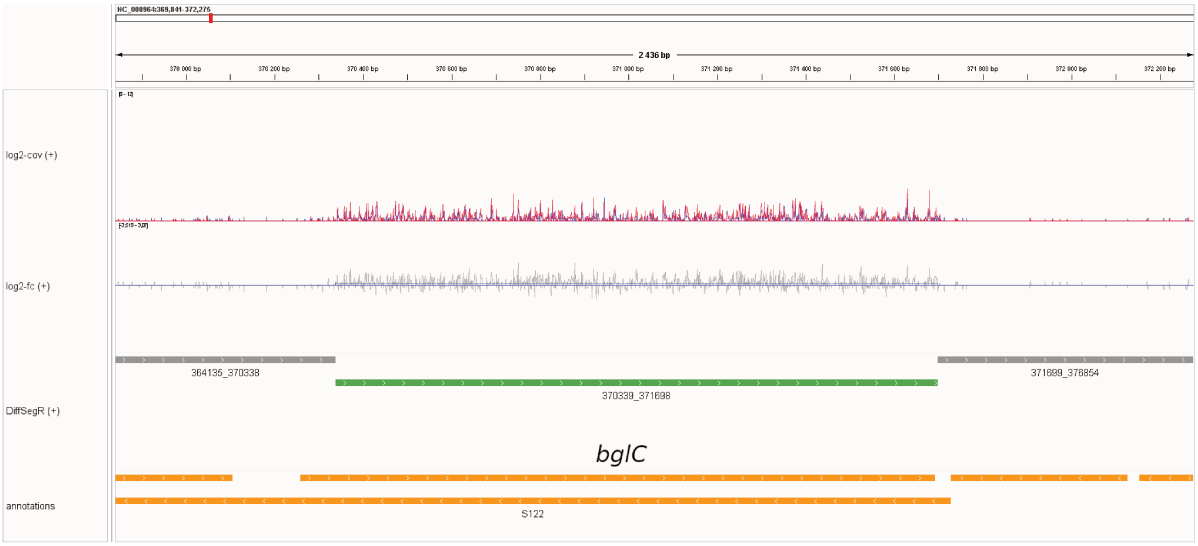


Figure S37: DiffSegR analysis genomic positions 369,841 to 372,275 on the forward strand in the Δ*rae1* dataset. The tracks are similar to those described in Figure S10.

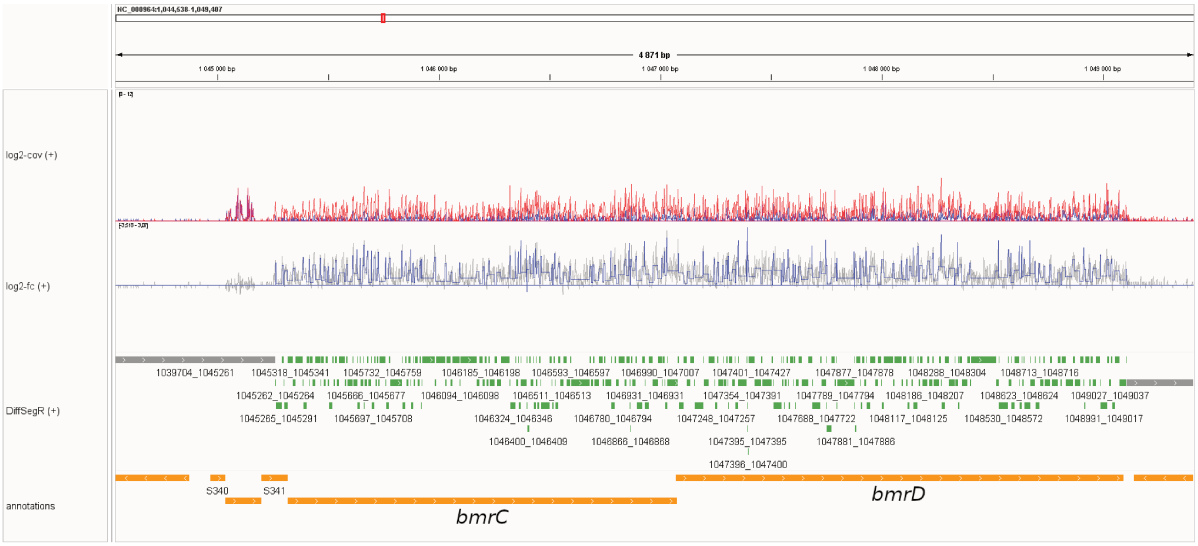


Figure S38: DiffSegR analysis of genomic positions 1,044,538 to 1,049,407 on the forward strand in the Δ*rae1* dataset. The tracks are similar to those described in Figure S10.

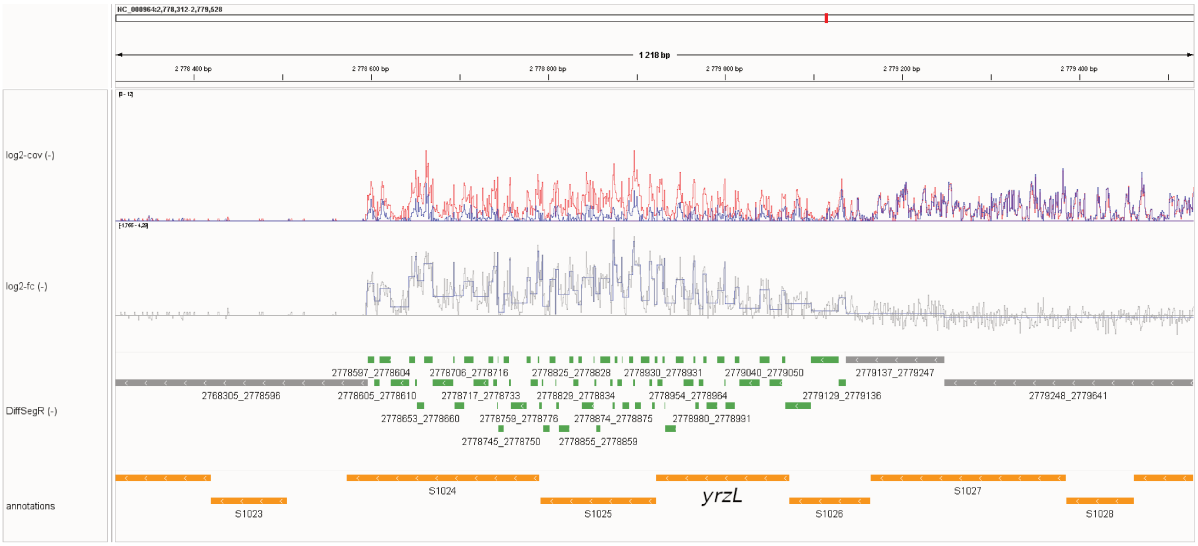


Figure S39: DiffSegR analysis of genomic positions 2,778,312 to 372,275 on the reverse strand in the Δ*rae1* dataset. The tracks are similar to those described in Figure S10.

Notes S1 to S5 with Supplementary Table S9 and Supplementary Figures S40 to S44

Note S1: Coverage profile Heuristics

We propose four heuristics to compute Q_{jr}. The first one takes advantage of the full length reads (Figure S40.A), the second only the 5' ends (Figure S40.B), the third only the 3' ends (Figure S40.C). In previous heuristics, we count the number of elements overlapping each genomic position. For the last heuristic, we compute the geometric mean of the second and third heuristics (as described in the main manuscript).

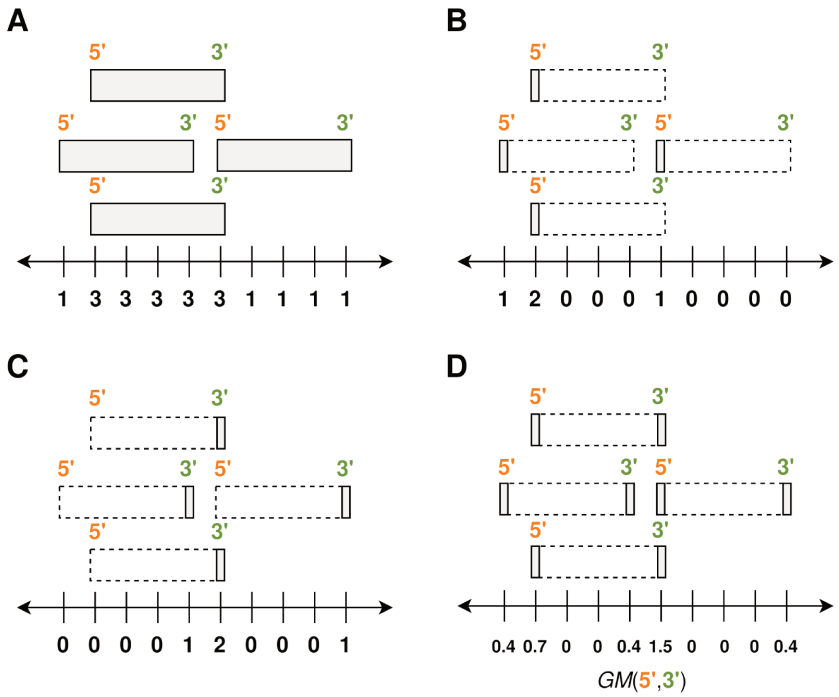


Figure S40: Four heuristics used to compute $Q_{j,r}$ from genome mapped reads. In **A** (full length reads), **B** (5' ends) and **C** (3' ends), at each genomic position, elements in gray are summed up to obtain the per-base coverage. In **D** (average) we compute the geometric mean of **B** and **C**.

Note S2: Full length reads based coverage is more auto-correlated

From a biological perspective, transcription at a particular base is likely to be dependent on transcription at the previous base, but counting a read at all the bases it covers further increases the auto-correlation between counts at neighboring bases. Visually it smooths the profile and gives the false impression that there is little variability. The per-base log2-FC being a transformation of coverages, it is also affected by the auto-correlation. We aimed for a balance between biological consistency and statistical difficulty.

For coverage heuristics, we computed the lag-1 to lag-5 auto-correlation function (ACF) on the per-base log2-FC in both *rnc3/4* and *pnp1-1* datasets. As expected, in *rnc3/4* the lag-5 ACF $\in [0, 1]$ is larger on the per-base lof2-FC calculated on full lengths reads based coverage (0.967) than those calculated on the 5' ends (0.341), the 3' ends (0.379) or the average profile of both (0.483). We observe the same tendency in *pnp1-1* with a lag-5 ACF of respectively 0.989, 0.690, 0.704 and 0.801. Results from lag-1 to lag-4 are available in Table S8. The segmentation model implemented in DiffSegR assumes that counts at every base pair are independent but still have a certain level robust to auto-correlation, certainly not 0.967 or 0.989.

Table S9: The per-base log2-FC is affected by the auto-correlation. As empirical confirmation, we computed the lag-1 to lag-5 auto-correlation function (ACF) on the log2-FC per-base in both *rnc3/4* and *pnp1-1* datasets.

dataset & ACF lag / coverage type	*pnp1-1*				Δ*rae1*			
	lag-1	lag-2	lag-3	lag-4	lag-1	lag-2	lag-3	lag-4
full length	0.998	0.996	0.994	0.992	0.994	0.988	0.981	0.975
5' ends	0.726	0.713	0.704	0.696	0.416	0.396	0.373	0.355
3' ends	0.726	0.719	0.712	0.709	0.433	0.416	0.402	0.389
geometric mean	0.820	0.812	0.807	0.804	0.532	0.518	0.504	0.491

Note S3: Flanking expressed regions are biased in 3' or 5’ end of reads based coverages

The reads in our libraries have lengths longer than 75 nt. This means that the coverage profiles computed using the 5' end of the reads do not adequately cover the 3' end of the expressed regions, and vice versa for the coverage profiles calculated using the 3' end of the reads. Taking the geometric mean of the two profiles partially resolves this issue (Figures S41-42).

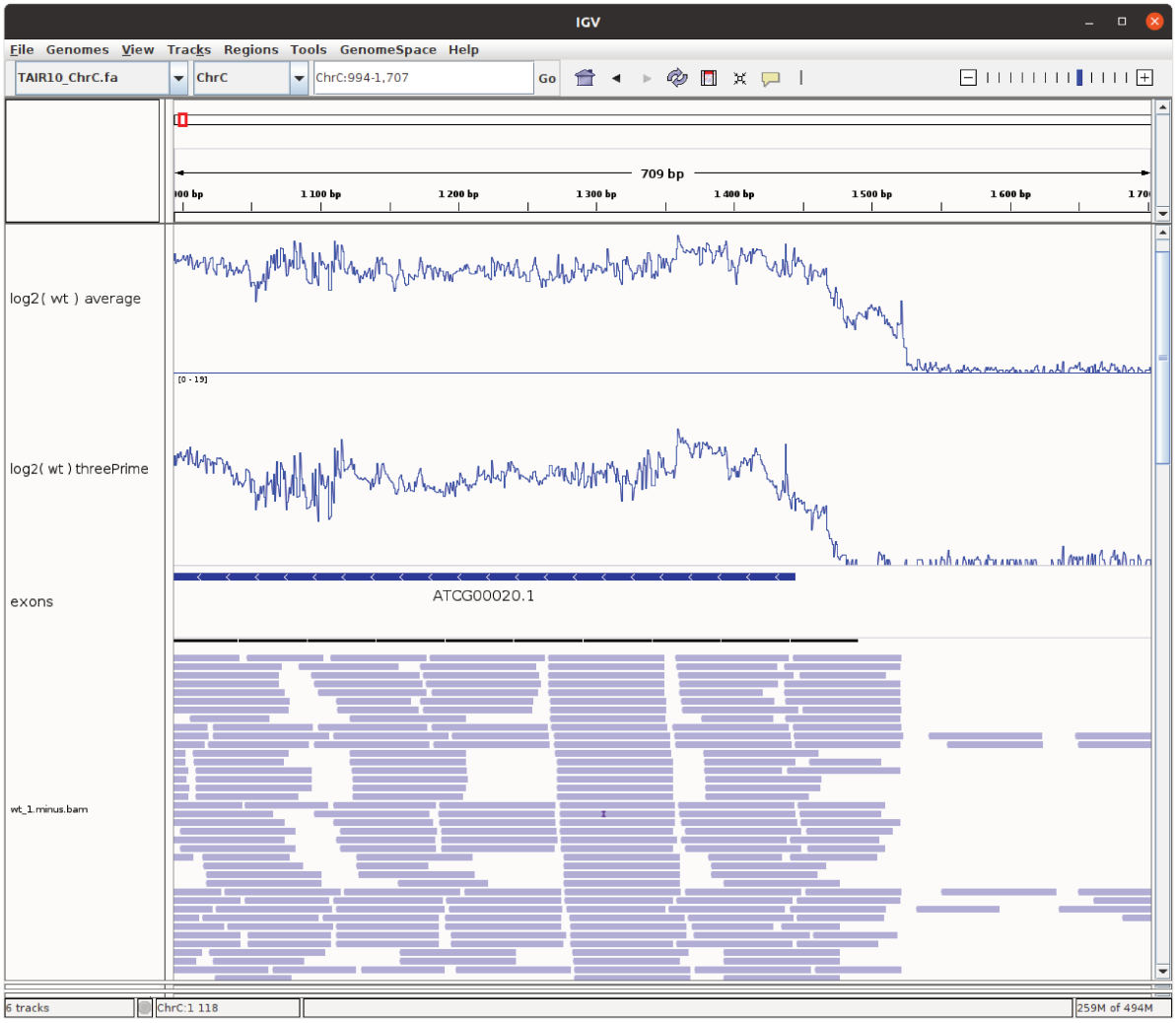


Figure S41: 3' ends of the expressed regions are poorly covered in the coverage profiles calculated on 5' ends of reads. Overview of the positions 214 to 927 on the reverse strand of the chloroplast genome. The expressed region corresponds to 3' ends of the *psbA* gene.

The first track stands for the coverage, on logarithmic scale, based on the average of 5' and 3' ends of the *WT* condition in *pnp1-1* dataset The second track stands for the coverage profile calculated on the 5' ends of reads, also on a logarithmic scale, of the same condition. The third track stands for exons boundaries. The last track stands for a sample of mapped reads.

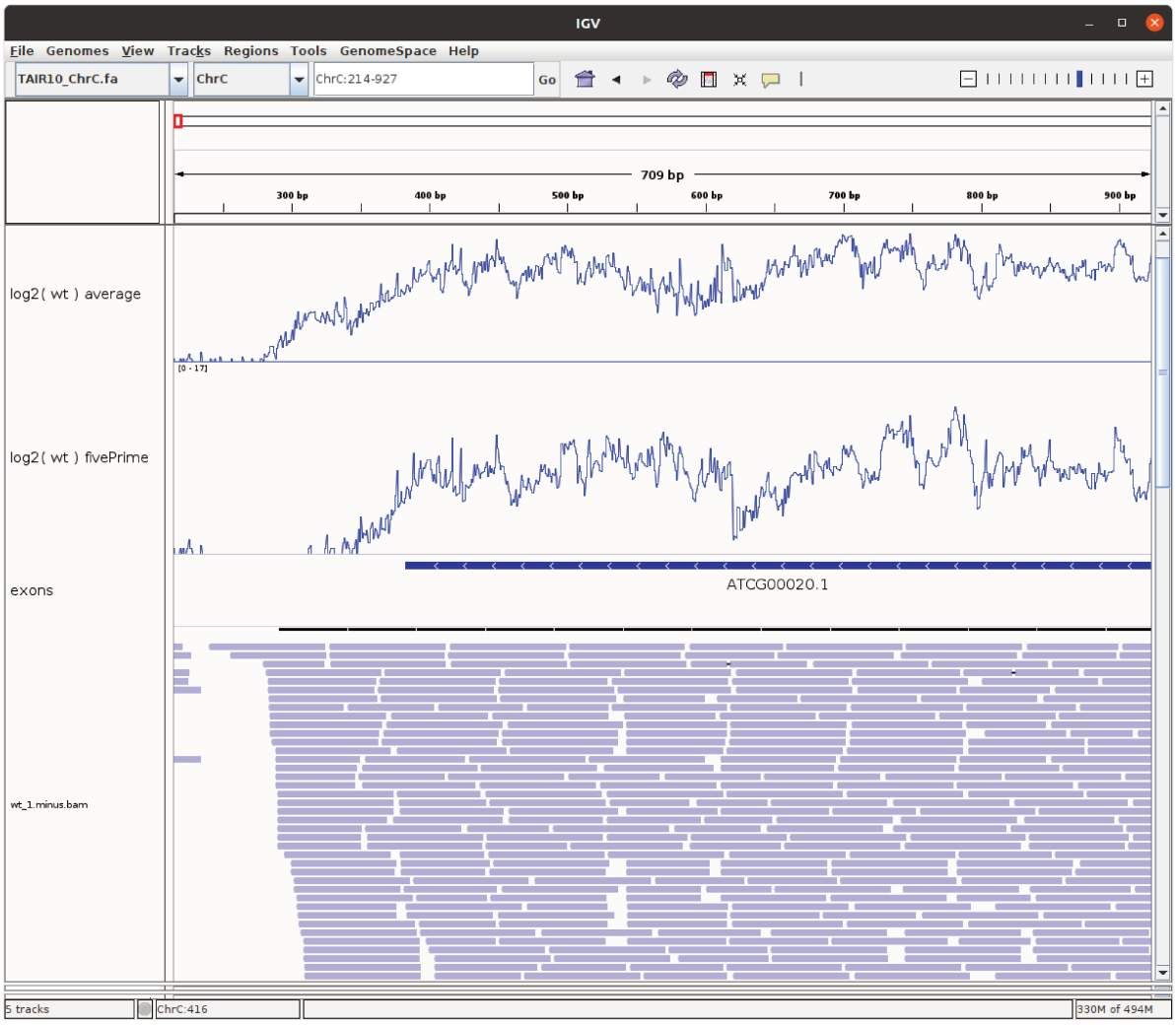


Figure S42: 5' ends of the expressed regions are poorly covered in the coverage profiles calculated on 3' ends of reads. Overview of the positions 994 to 1,707 on the reverse strand of the chloroplast genome. The expressed region corresponds to 5' ends of the *psbA* gene. The first track stands for the coverage, on logarithmic scale, based on the average of 5' and 3' ends of the *WT* condition in the *pnp1-1* dataset. The second track stands for the coverage profile calculated on the 5' ends of reads, also on a logarithmic scale, of the same condition. The third track stands for exons boundaries. The last track stands for a sample of mapped reads.

Note S4: Segmenting in two or three levels merge neighboring differential regions with different log2-FC

In the following paragraph we use a theoretical example to explain the limits of the segmentation models used by state-of-the-art methods to recover differentially expressed regions.

Apart from parseq, which segments the mean of coverages, several other tools (derfinder SB, derfinder RL, srnadiff HMM) use a two-level segmentation with differentially expressed (DE) and non-DE levels, or expressed and not-expressed levels. srnadiff IR uses a three-level segmentation with down-regulated, up-regulated, and non-DE levels. However, we argue that this can be detrimental to biological interpretation. For example, if a gene is up-regulated in condition 2 compared to condition 1 and has an intron retention, a two or three-level segmentation will result in a single large region that effectively combines the up-regulation and intron retention (Figure S43). Additionally, two-level segmentation can merge differential regions with opposite signs of log2-FC, which can also reduce statistical power. A segmentation model that does not make assumptions about the number of levels in the per-base log2-FC should be able to discriminate adjacent differently expressed regulatory events, and this is what we tested in DiffSegR.

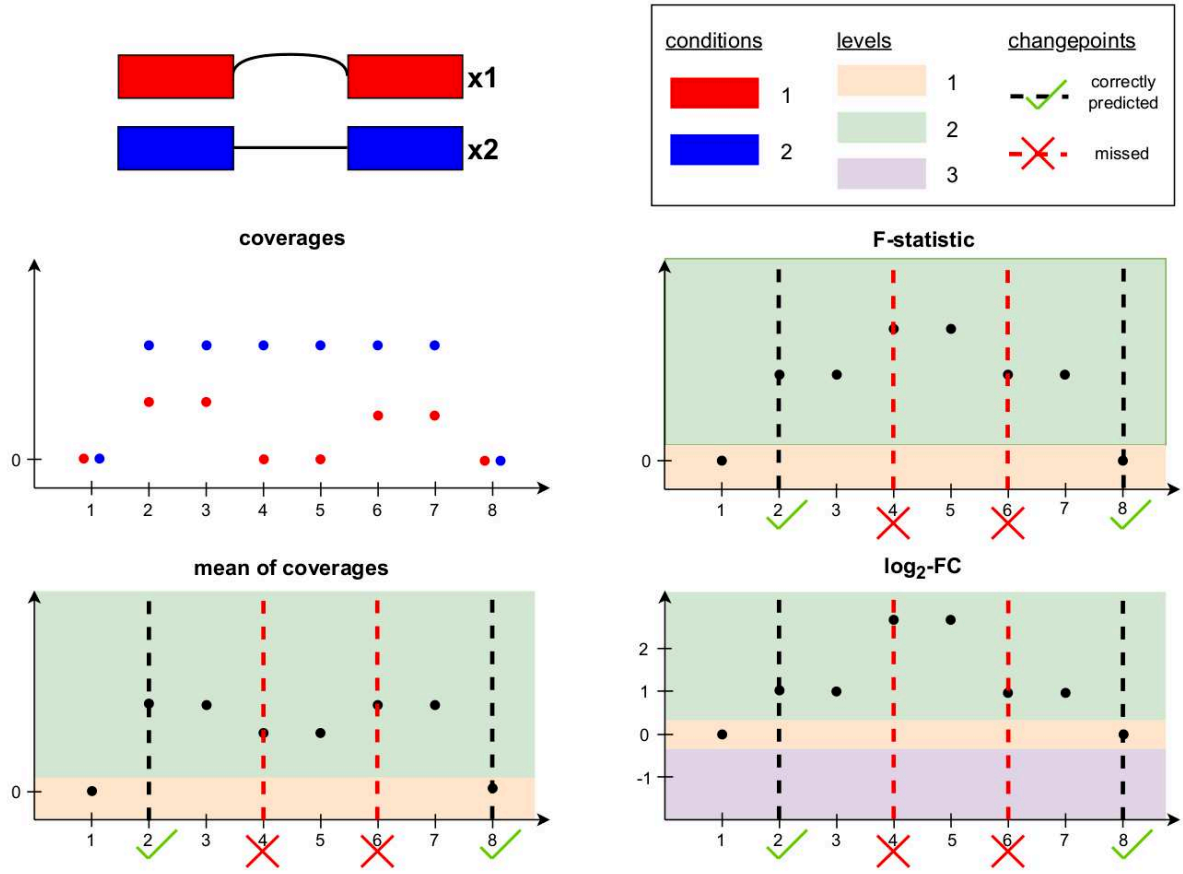


Figure S43: Segmenting in two or three levels merge neighboring differential regions with different per-base log2-FC. In this second example the gene is two times more expressed in condition 1 than in condition 2, and in condition 2 the gene also undergoes intron retention. As a result, the log2-FC is higher within the intron than within the exons. Segmenting the mean of coverages in two levels (expressed and not-expressed) merges the up-regulation and intron retention. Segmenting the F-statistic in two levels (differentially expressed (DE) and non-DE) or the per-base log2-FC in three levels (up-regulated, down-regulated, non-DE) also results in the merging of these two events.

Note S5: Additional changes in the per-base coverage and F-statistic

In the following paragraph we use a theoretical example to explain the limits of the signals used by state-of-the-art methods to recover differentially expressed regions. These limits are assessed on real data in the *DiffSegR better captures the differential landscape* section.

There are various factors that can influence the coverage of a DNA sequence, including the per-base expression, 5'/3' bias, PCR bias, GC bias, and non-random priming. This can result in significant variation in coverage from one base to another. Therefore, it is possible that changes in coverage may not always align with differences in transcription between two biological conditions.

Ignoring any normalization issue, consider bases that follow each other with (non-differential scenario) respectively 1 and 40 counts in both condition 1 and 2 ; (differential scenario) respectively 1 and 40 counts in condition 1 & 3 and 120 counts in condition 2.

There is a noticeable change in coverage between the two bases, yet the log2-FC remains constant and equal to 0 in the non-differential scenario and $\log(3)$ in the differential scenario. These additional changes likely lose statistical power and (in the absence of a post-processing step) make it more difficult to interpret the results biologically, as a single biological regulation will be identified as two or more. Note that, with respect to these two examples, the F-statistic is better behaved yet not perfect. In the first scenario the F-statistics should be equal to its expected value under the null and we should not detect any change between the two positions. However in the differential scenario as power depends on the underlying counts it is likely that we will detect a change between position 1 and 2. The non-differential scenario is illustrated in Figure S44 by the positions 6, 7 and 8. The differential scenario is illustrated in Figure S44 by positions 2 and 3. Given that our primary goal is to identify differences in transcription between two biological conditions, we believe that directly segmenting the per-base log2-FC is a better option, and this is what we tested in DiffSegR.

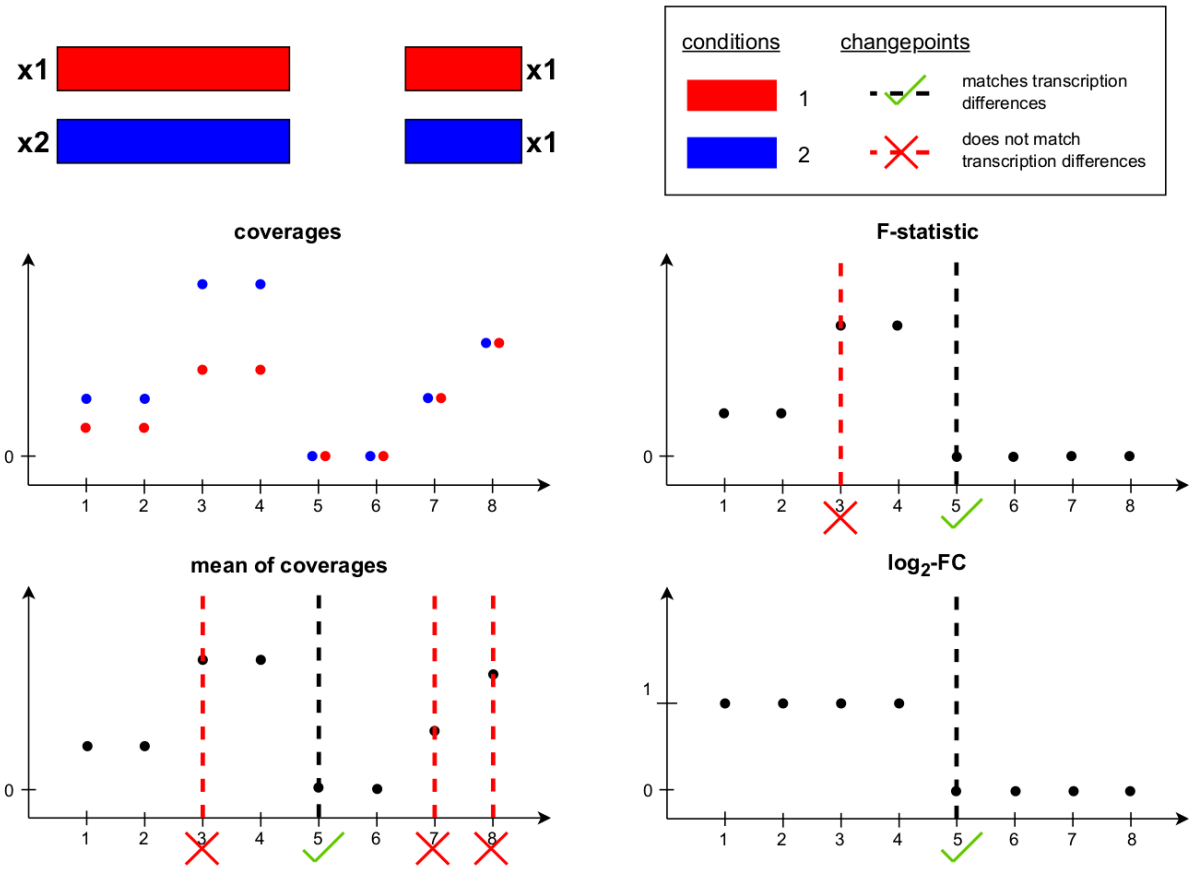


Figure S44: Additional changes in the per-base coverage and F-statistic. The first gene is twice as highly expressed in condition 2 compared to condition 1, while the second gene has the same level of expression in both conditions. Segmentation of the mean of coverages, the F-statistic, and the per-base log2-FC results in 4, 2, and 1 changes, respectively. The changes between positions 2 to 3 and 7 to 8 are caused by coverage bias, while the changes between positions 4 to 5 and 6 to 7 mark the end of transcription of the first gene and the start of transcription of the second gene. The change between positions 4 to 5 is the only one that also corresponds to a difference in transcription between the two conditions.

Chapter D

Coordination of RNA events

D.1 Full Length Transcriptome Highlights the Coordination of Plastid Transcript Processing

This article was published in the journal *International Journal of Molecular Sciences* (doi.org/10.3390/ijms222011297).

International Journal of
Molecular Sciences

MDPI

Article

Full Length Transcriptome Highlights the Coordination of Plastid Transcript Processing

Marine Guilcher [1,2], Arnaud Liehrmann [1,2,3], Chloé Seyman [3], Thomas Blein [1,2], Guillem Rigaill [1,2,3], Benoit Castandet [1,2] and Etienne Delannoy [1,2,*]

1 Institute of Plant Sciences Paris-Saclay (IPS2), Université Paris-Saclay, CNRS, INRAE, Université Evry, 91405 Orsay, France; marine.guilcher@universite-paris-saclay.fr (M.G.); arnaud.lieh@gmail.com (A.L.); thomas.blein@cnrs.fr (T.B.); guillem.rigaill@inrae.fr (G.R.); benoit.castandet@universite-paris-saclay.fr (B.C.)
2 Institute of Plant Sciences Paris-Saclay (IPS2), Université de Paris, CNRS, INRAE, 91405 Orsay, France
3 Laboratoire de Mathématiques et de Modélisation d'Evry (LaMME), Université d'Evry-Val-d'Essonne, UMR CNRS 8071, ENSIIE, USC INRAE, 91000 Evry, France; chloeseyman@gmail.com
* Correspondence: etienne.delannoy@inrae.fr

Abstract: Plastid gene expression involves many post-transcriptional maturation steps resulting in a complex transcriptome composed of multiple isoforms. Although short-read RNA-Seq has considerably improved our understanding of the molecular mechanisms controlling these processes, it is unable to sequence full-length transcripts. This information is crucial, however, when it comes to understanding the interplay between the various steps of plastid gene expression. Here, we describe a protocol to study the plastid transcriptome using nanopore sequencing. In the leaf of *Arabidopsis thaliana*, with about 1.5 million strand-specific reads mapped to the chloroplast genome, we could recapitulate most of the complexity of the plastid transcriptome (polygenic transcripts, multiple isoforms associated with post-transcriptional processing) using virtual Northern blots. Even if the transcripts longer than about 2500 nucleotides were missing, the study of the co-occurrence of editing and splicing events identified 42 pairs of events that were not occurring independently. This study also highlighted a preferential chronology of maturation events with splicing happening after most sites were edited.

Keywords: *Arabidopsis thaliana*; plastid; co-maturation; post-transcriptional; nanopore

Citation: Guilcher, M.; Liehrmann, A.; Seyman, C.; Blein, T.; Rigaill, G.; Castandet, B.; Delannoy, E. Full Length Transcriptome Highlights the Coordination of Plastid Transcript Processing. *Int. J. Mol. Sci.* **2021**, *22*, 11297. https://doi.org/10.3390/ijms222011297

Academic Editors: Mamoru Sugita and Sarath Chandra Janga

Received: 29 August 2021
Accepted: 11 October 2021
Published: 19 October 2021

1. Introduction

Plastids are derived from the endosymbiosis between photosynthetic organisms and an ancestral Eukaryote. Although most of the initial symbiont genes have been transferred to the nucleus during the course of evolution, plastids of land plants and other photosynthetic Eukaryotes still maintain a small but essential genome. It mainly encodes subunits of each of the photosynthetic complexes (Photosystem I and II, cytochrome b6/f, ATP synthase and Rubisco) and some of the plastid gene expression (PGE) machinery [1]. Most of the proteins involved in PGE are, however, encoded in the nucleus and need to be targeted back to plastids. As a consequence, PGE retains characteristics from both eukaryotes and bacterial systems, resulting in a sophisticated interplay between nucleus and plastid encoded factors [2–4].

A striking feature of PGE is the importance and complexity of the post-transcriptional maturation steps. In addition to the intron removal by RNA splicing [5] and the specific conversion of cytosines into uridines by RNA editing [6], complete maturation also requires intergenic cleavage of the multigenic transcripts and the generation of 5′ and 3′ ends through RNA processing [7,8]. Most of the RNA binding proteins (RBP) or ribonucleases known to be involved in PGE are localized in a membraneless structure surrounding the plastome—the nucleoid [9]. This close association between RNA maturation factors might be an explanation for the multiple pleiotropic effects observed in chloroplast mutants [7].

Various investigations, both in vitro and in organellar gene expression mutant plants, have indeed revealed situations where the different maturation events can influence each other. For example, intron removal is a prerequisite for editing in the *ndhA* second exon [10] and *atpF* splicing is severely reduced in the *aef1* mutant in which the editing of *atpF_12707* is abolished [11]. *Arabidopsis thaliana* chloroplast RNA editing is affected in a mutant deficient for the exoribonuclease PNPase [12] while correct processing of the potato mitochondrial tRNA Phe requires RNA editing [13]. Editing sites can even influence each other. For example, in *A. thaliana*, editing of mitochondrial *ccmB_17869* by MEF19 depends on the editing of *ccmB_17884* by MEF37 [14]. Similarly, in *Physcomitrium patens,* editing of the mitochondrial *ccmFc-C103* by PpPPR_65 controls editing of *ccmFc-C122* by PpPPR_71 [15,16].

These dependencies are usually explained according to two models. First, one maturation event can modify the RNA secondary structure necessary for the second maturation. Second, the proteins responsible for the maturation can interact with each other or, more directly, target several maturation events. Most studies, however, only focused on a limited set of transcripts or RNA maturation events precluding any general conclusions. This illustrates the urgent need for the development of global approaches capable of simultaneously studying all the RNA maturation processes, at the transcriptomic level. This issue has recently been tackled by the increasing use of Illumina-based RNA-Seq strategies to study PGE from transcription to translation [17–23].

Although this has considerably increased the power and sensitivity of PGE analyses, it is ill-suited to study the potential coordination between maturation steps. The short reads used by Illumina technology (the maximum insert size of Illumina TruSeq RNA libraries reaches around 350 base-pairs) make it impossible to monitor the co-occurrence of these events on single RNA transcripts that can be several kilobases long. An alternative would be to take advantage of other sequencing technologies such as PacBio or Oxford Nanopore. They theoretically allow the sequencing of full-length cDNAs or RNA and should therefore overcome the current technical limitations [24]. A major issue, however, is that most of the available library preparation protocols only capture polyadenylated RNA transcripts, therefore excluding plastid transcripts. A recent protocol analyzing chromatin-bound transcripts also captures non-polyadenylated transcripts but was not applied to the analysis of plastid transcripts [25,26].

In this work, we describe the analysis of the *A. thaliana* plastid transcriptome by sequencing full-length non-polyadenylated and polyadenylated cDNAs using the Oxford Nanopore technology (ONT). This analysis identified all known post-transcriptional maturation events and provided an overview of their coordination in normal growth conditions.

2. Results

2.1. A Protocol to Sequence the Full Length Plastid Transcriptome

The library synthesis protocol is derived from the Switching Mechanism at the 5′ end of RNA Transcript (SMART) technology developed to synthesize full-length cDNAs [27]. Because polyadenylation of chloroplastic RNAs acts as a degradation signal [28], we, however, had to first start with the ligation of an RNA adapter (modified from Hotto et al. [29]) at the 3′ end of the RNAs to allow the priming of the reverse transcription and an rRNA depletion before completing the cDNA synthesis. The cDNAs are then incorporated into an ONT sequencing library and sequenced. Sampling RNA from leaves of 5 week-old col-0 *A. thaliana* plants grown in long-day conditions at 20 °C, we mapped between 1.55 million and 2.69 million stranded reads (mapping rate between 98.5% and 99.8%) to the *A. thaliana* genome including between 10% and 40% to the plastid genome and between 0.3% and 0.8% to the mitochondrial genome. The median error rate was between 4% and 4.4%. The rRNA depletion was very efficient with less than 0.1% of reads mapping to rRNA loci. More than 99.5% of the reads mapped to the annotated nuclear genes corresponding to the sense orientation, a proportion similar to Illumina stranded RNA-Seq. Most of the reads (99%) were between 195 and 2141 nucleotides (nt) long with a median size of 852 nt and a maximum size of 4805 nt. In *A. thaliana*, 7261 genes are producing transcripts longer

than 2141 nt and more than 390 genes (including the plastid *ycf2* gene) are producing transcripts longer than 4800 nt. Based on the whole transcriptome, the 3′ to 5′ transcript coverage was better with our protocol than for similar samples analyzed using Illumina sequencing for transcripts below 1500 nt (22,853 genes; Figure S1). For transcripts above 1500 nt (17,985 genes), the Illumina sequencing performed better and a moderate 3′-5′ bias can be observed. These results confirm that our nanopore reads were mostly full-length and stranded but that the longer transcripts are missing from the sequencing libraries.

2.2. A Representative Picture of the Plastid Transcriptome

With at least 275,000 reads mapped on the plastid genome for each biological replicate, the coverage is deep enough to have a good representation of the plastid transcriptome. To verify that the sequencing data are correctly capturing the plastid transcriptome, we looked at the complex transcriptional profile of the *psbB* to *petD* genomic region (Figure 1).

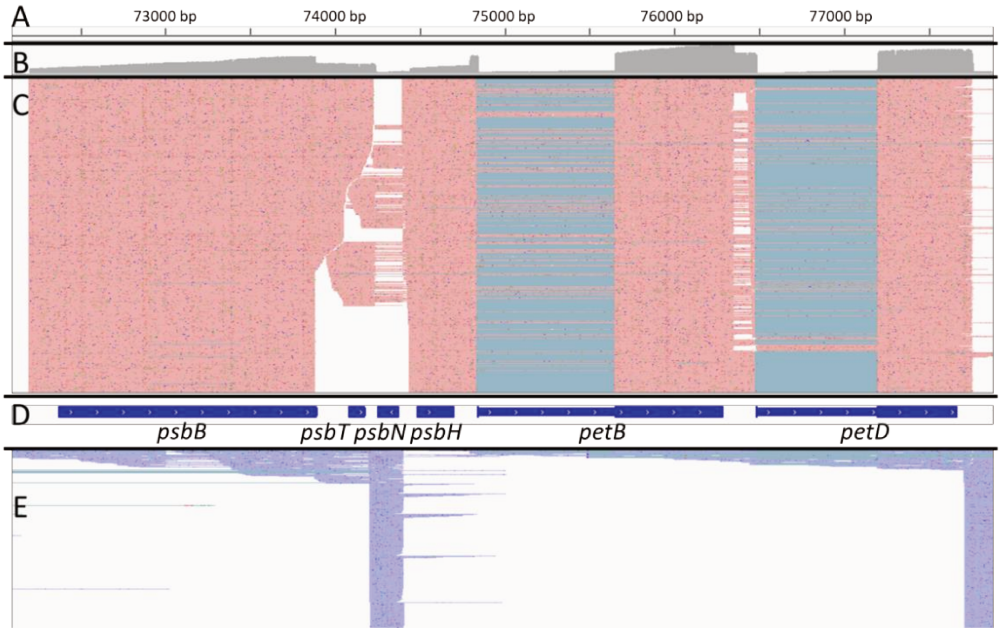


Figure 1. The complexity of the *psbB-petD* locus. Screenshots of Integrative Genomics Viewer (IGV) displaying nanopore reads mapping to the *psbB-petD* locus. (**A**) plastid genomic position. (**B**) coverage track displaying the number of reads at each nucleotide. (**C**) screenshot of reads mapping on the Watson strand. Matching bases are shown in red. Split reads are joined by blue lines. (**D**) Annotation of the locus. Introns are shown as thinner segments. (**E**) screenshot of reads mapping on the Crick strand. Matching bases are shown in purple. Split reads are joined by blue lines.

Following transcription, transcripts from this multigenic locus are processed into multiple poly- or monocistronic isoforms on both genomic strands [30,31]. A rapid overview of the reads showed the transcription of *psbN* on the Crick strand while *psbB*, *psbT*, *psbH*, *petB* and *petD* were transcribed from the Watson strand as expected. The spliced *petD* and *petB* transcripts were also found. Taking advantage of long-read sequencing, it is possible to emulate Northern blots by selecting reads which map on specific positions and plotting the distribution of the read lengths. Felder et al. [30] studied the involvement of HCF107 in the processing of the *psbB* to *petD* locus with an extensive use of Northern blots, allowing a comparison of the two methods. We therefore generated virtual Northern blots for *psbN*, *psbH*, *petB* and *petD* (Figure 2) using virtual probes equivalent to the probes used for Figure 4C,E,H,I of Felder et al. [30].

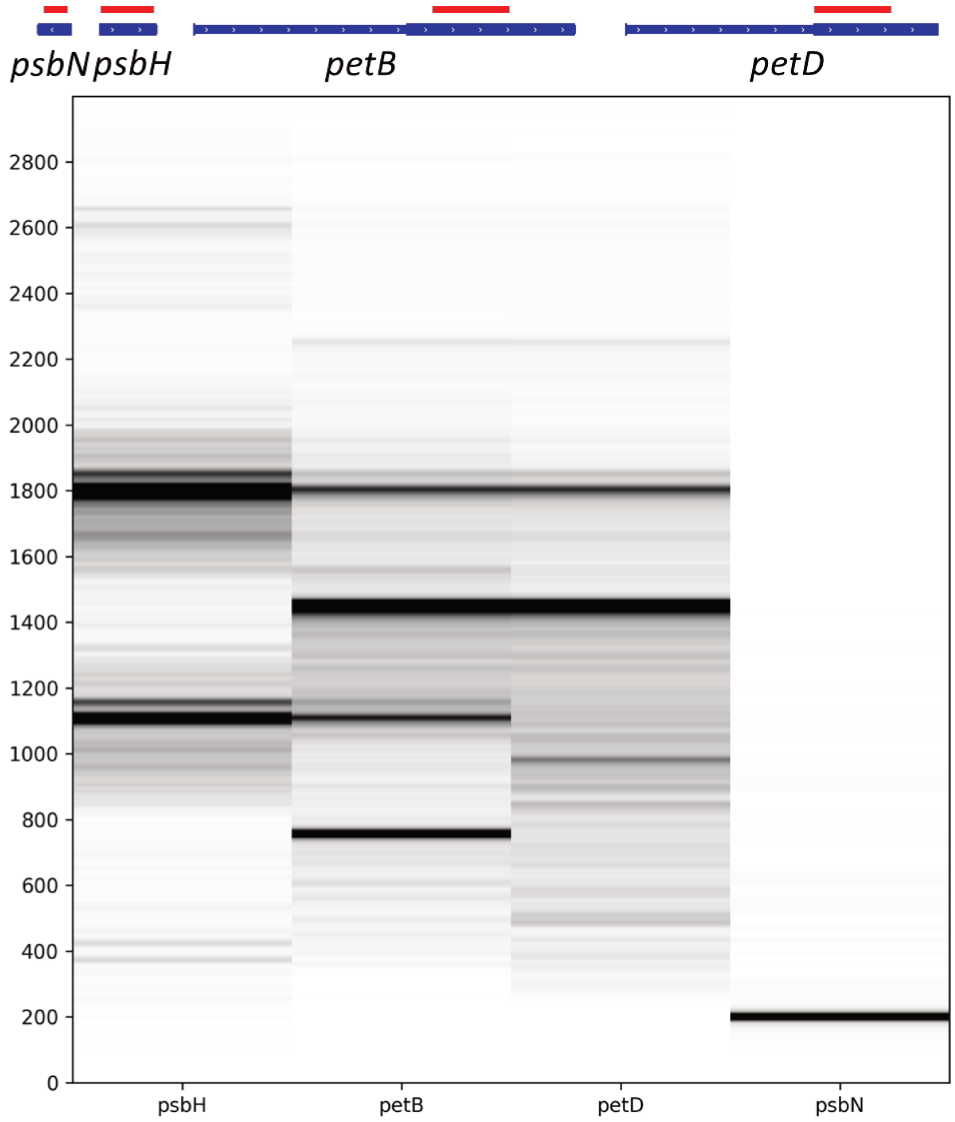


Figure 2. Virtual Northern blots derived from the nanopore sequencing. Northern blots were emulated from nanopore reads mapping to the sequences of *psbN*, *psbH*, the second exon of *petB*, or the second exon of *petD* shown in red on the genomic map displayed above. The size (in nt) is shown on the left.

Reads mapping to *psbN* were almost exclusively 200 nt long which is compatible with the signal detected by a classic Northern blot. Reads mapping to *psbH* showed two major isoforms around 1100 nucleotides (nt) and 1800 nt but also two minor isoforms around 370 nt and 2600 nt. This profile is also compatible with the regular Northern blot. However, Felder et al. [30] also detected larger isoforms at 3300, 4100, 4900, and 5600 nt that were not captured in our sequencing libraries. The virtual Northern blot for *petB* showed four major isoforms at 750, 1100, 1450, and 1800 nt. A faint isoform may be present at 2250 nt. These isoforms were also detected by Felder et al. [30] who found additional isoforms at 2600, 3300, 4100, 4900, and 5600 nt. Finally, for *petD*, we found two major isoforms around 1450 and 1800 nt and minor isoforms around 990 and 2225 nt. We missed the larger isoforms detected by Felder et al. [30] but also a 1200 nt isoform described as an unspliced *petD* transcript which seemed to be replaced by our 990 nt isoform. The detection of sharp "bands" of the expected size in our virtual Northern blots confirms that the majority of the nanopore reads correspond to full-length cDNAs but this result also confirms that transcripts longer than 2–2.5 kb are under-represented in our sequencing libraries.

In these complex loci, it is sometimes difficult to identify all the bands on a regular Northern blot. For example, Felder et al. did not associate the 2200 nt transcript of their *petB* and *petD* Northern blots to a particular isoform.

Our sequencing showed that this transcript is most likely a polycistronic intermediate containing an unspliced *petB* with a spliced *petD* (Figure 3A). For *petD*, we detected a minor isoform around 990 nt. The associated transcripts corresponded to two distinct isoforms (Figure 3B). The first one corresponded to spliced *petD* transcripts but with 5′ ends within the second *petB* exon. The second one had a 5′ end in the *petD* intron at position 76,780 and included the second *petD* exon. Position 76,780 was identified as a transcription start site and multiple 5′ ends were mapped in this area [20]. Similarly, because of their poor resolution, regular Northern blots can miss isoforms of similar sizes. Our virtual Northern blot for *psbH* showed that the four peaks are double peaks: the main isoforms are each associated with isoforms which are 50 nt longer. When mapping these isoforms, we could show that the short and long isoforms are associated with different 5′ ends, the long one around the genomic position 74,393 and the short one around 74,441 (Figure 3C). According to Castandet et al. [20], position 74,441 corresponds to the major processed extremity of *psbH* while position 74,393 is a transcription start site.

Even if our nanopore reads showed no or only moderate 3′ to 5′ bias in general (Figure S1), some plastid transcripts showed 3′ to 5′ but also 5′ to 3′ biases (Figure 4).

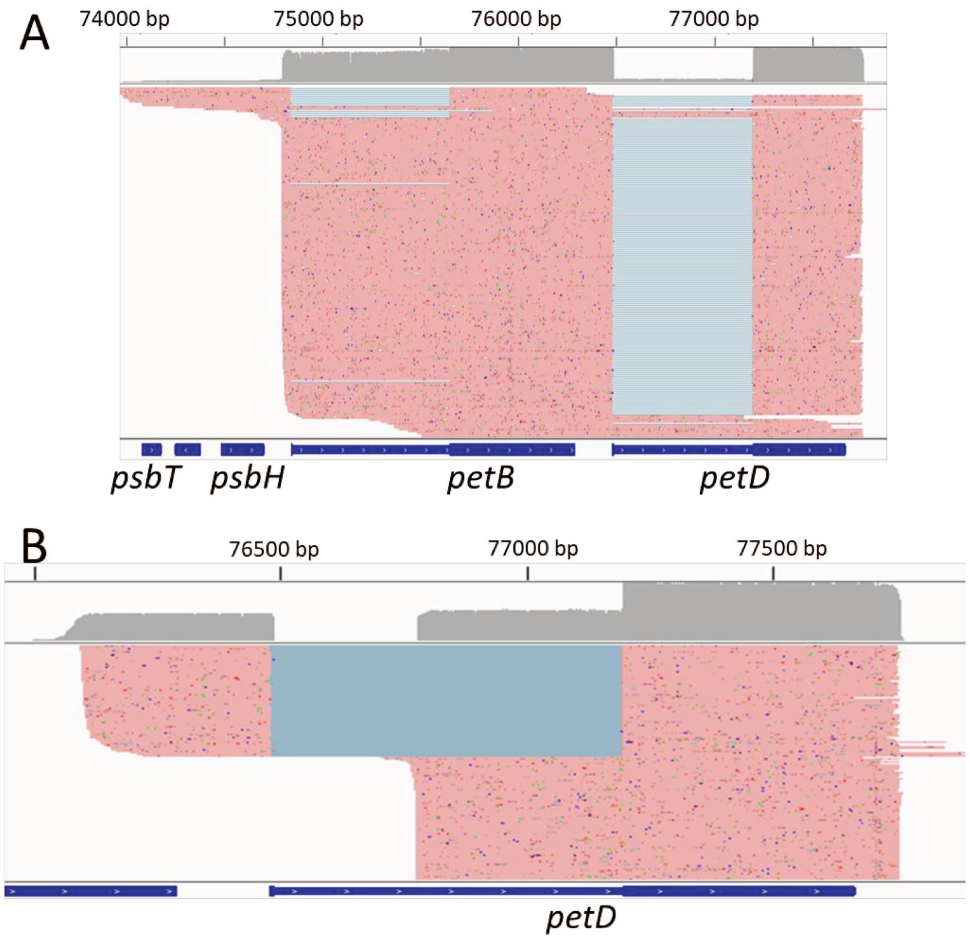


Figure 3. *Cont.*

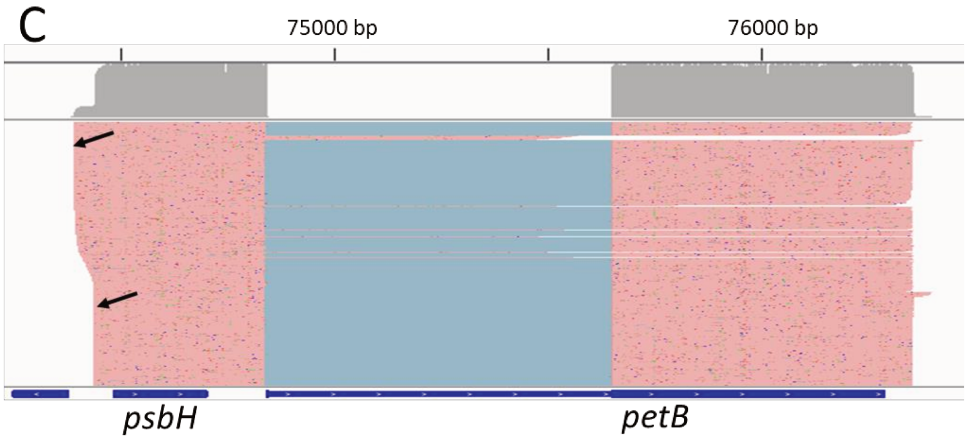


Figure 3. Identification of transcripts isoforms. Screenshots of IGV displaying the reads corresponding to various virtual Northern blot isoforms. Matching bases are shown in red. Split reads are joined by blue lines. Other colors indicate mismatches and indels. (**A**) Reads corresponding to the 2200 nt isoform of the *petB* and *petD* virtual Northern blots. (**B**) Reads corresponding to the 990 nt isoform of the *petD* virtual Northern blot. (**C**) Reads corresponding to the 1100–1150 isoform of the *psbH* virtual Northern blot. The two 5′ ends are shown by black arrows.

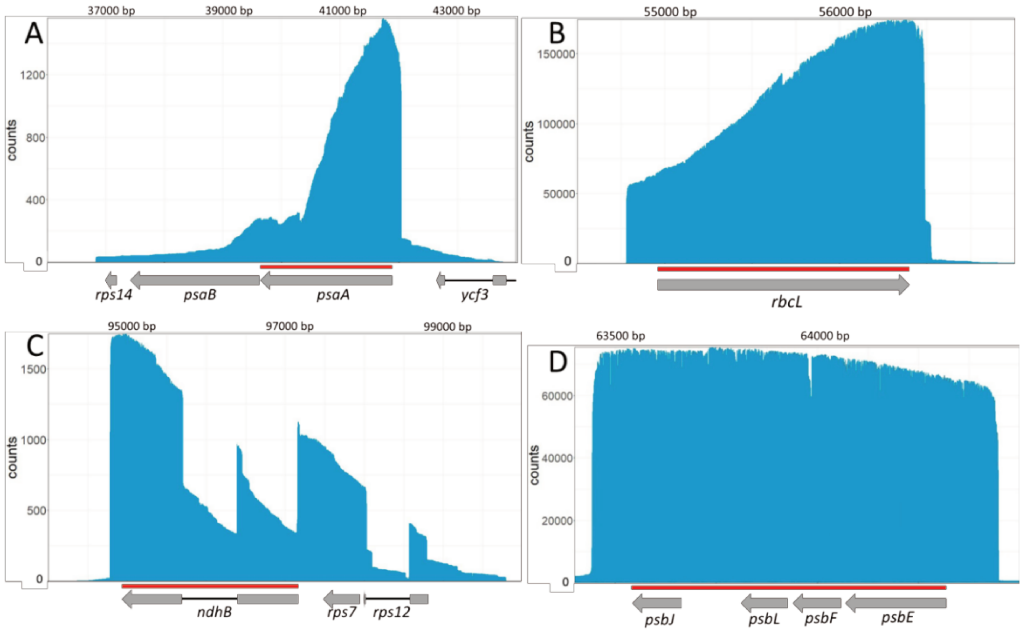


Figure 4. Examples of coverage biases in plastid transcripts. For each panel, the coverage at single-nucleotide resolution by strand-specific reads overlapping at least partially the genomic regions shown in red is shown. At the top, the genomic positions are shown while the coding sequences associated with these regions are shown below as gray arrows or boxes. Introns are represented as black lines. (**A**): *psaA* transcripts. (**B**): *rbcL* transcripts. (**C**): *ndhB* transcripts. (**D**): *psbE-psbJ* transcripts.

In *ndhB*, the coverage at the 3′ end was only about 20% of the coverage at the 5′ end. This bias could be technical but it was not the same for other transcripts (for example *rbcL* or the polycistronic *psbE-F-L-J* transcript). For *psaA*, we observed a strong 5′ to 3′ bias. It may illustrate the pattern of transcript degradation of the 5300 nt long *psaA-psaB-rps14* transcript [32] which is absent of our sequencing libraries.

Finally, post-transcriptional maturation events can be quantitatively analyzed. Known editing sites could be detected with rates comparable to leaf datasets (Tables 1 and S1) previously published by Guillaumot et al. [33] (Pearson correlation = 0.97; p-value < 2.2×10^{-16}) and Ruwe et al. [12] (Pearson correlation = 0.94; p-value < 2.2×10^{-16}).

Table 1. Quantification of known editing and splicing events.

Name	Type	Maturation Rate	Maturation Rate (Guillaumot et al., 2017)	Maturation Rate (Ruwe et al., 2013)
int_RPS16	splicing	4%	4%	NA
int_ATPF	splicing	89%	82%	NA
int_RPOC1	splicing	64%	19%	NA
int_YCF3_i2	splicing	79%	42%	NA
int_YCF3_i1	splicing	63%	45%	NA
int_CLP_i2	splicing	60%	71%	NA
int_CLP_i1	splicing	69%	62%	NA
int_PETB	splicing	91%	58%	NA
int_PETD	splicing	97%	62%	NA
int_RPL16	splicing	69%	12%	NA
int_RPL2.1	splicing	66%	52%	NA
int_NDHB.1	splicing	68%	55%	NA
int_RPS12C	splicing	92%	81%	NA
int_NDHA	splicing	68%	27%	NA
matK_2931	editing	53%	79%	93%
atpF_12707	editing	89%	91%	95%
atpH_UTR_13210	editing	5%	3%	4%
rpoC1_21806	editing	33%	21%	15%
rpoB_23898	editing	87%	82%	85%
rpoB_25779	editing	64%	83%	86%
rpoB_25992	editing	69%	76%	94%
psbZ_35800	editing	93%	90%	95%
rps14_37092	editing	89%	93%	94%
rps14_37161	editing	92%	97%	96%
ycf3_i2_43350	editing	16%	10%	12%
rps4_UTR_45095	editing	6%	3%	10%
ndhK_ndhJ_49209	editing	4%	4%	6%
accD_57868	editing	90%	95%	99%
accD_58642	editing	76%	75%	83%
psbF_63985	editing	90%	98%	98%
psbE_64109	editing	95%	100%	100%
petL_65716	editing	79%	91%	86%
rps18_UTR_68453	editing	3%	4%	NA
rps12_69553	editing	21%	26%	27%
clpP_69942	editing	82%	72%	81%
rpoA_78691	editing	78%	76%	91%
rpl23_86055	editing	34%	74%	75%
ycf2_as_91535	editing	3%	4%	NA
ndhB_UTR_94622	editing	8%	0%	NA
ndhB_94999	editing	88%	93%	94%
ndhB_95225	editing	95%	98%	99%
ndhB_95608	editing	87%	84%	80%
ndhB_95644	editing	78%	87%	81%
ndhB_95650	editing	88%	91%	84%
ndhB_96419	editing	75%	94%	92%
ndhB_96439	editing	6%	4%	6%
ndhB_96457	editing	6%	3%	5%
ndhB_96579	editing	90%	89%	90%
ndhB_96698	editing	81%	88%	82%
ndhB_97016	editing	94%	94%	95%
ndhF_112349	editing	85%	93%	96%
ndhD_116281	editing	76%	83%	92%
ndhD_116290	editing	77%	84%	90%
ndhD_116494	editing	88%	90%	93%
ndhD_116785	editing	94%	97%	98%
ndhD_117166	editing	35%	33%	45%
ndhG_118858	editing	69%	78%	85%

NA: Not Analyzed. The genomic position of each site and the corresponding nomenclature of Rüdinger et al. [34] are given in Table S1.

It should be noted that the analysis of poorly edited sites by nanopore sequencing must be done carefully because of the relatively high error rate of this technology. For example, using the same pipeline as Guillaumot et al., we detected 123 plastid C to U transitions with a rate higher than 10% but only 44 of them were also detected by Guillaumot et al. [33] using Illumina sequencing. Similarly, intron splicing efficiency could be measured (Tables 1 and S1) and it varied from 4% to 97% depending on the intron. Most values are higher (by 22 points on average) than the efficiencies measured by Guillaumot et al. [33] using Illumina. This bias could be explained by an underrepresentation of long (unspliced) transcripts compared to short (spliced) ones. However, this bias is not linked to the unspliced transcript length, the intron length or the unspliced/spliced size ratio (Figure S2). An alternative, but not exclusive, explanation is that the abundance of unspliced transcripts is difficult to estimate with Illumina sequencing.

2.3. Some Post-Transcriptional Events Are Coordinated and Ordered

Because editing and splicing events are well defined (a single genomic position, either processed or not), it is easy to statistically analyze the possible coordination between these events.

Although 1596 co-occurring events could theoretically be expected with 14 splicing and 43 editing events analyzed, only 138 co-occurrences were detected at least once. This is, however, expected as all events are not found on a single transcript (Table S2). Out of these 138 pairs of maturation events, 42 were not found to occur independently (Figure 5). Conversely, we did not detect any complete dependency (when one maturation event is absolutely required for another maturation event to occur). We observed partial dependencies between splicing events (*clpP* introns, *petD* and *petB* introns), editing and splicing events (in the *atpF*, *clpP* and *ndhB* transcripts) and between editing events (in the *rps14*, *ndhD* and *ndhB* transcripts). This partial dependency also occurred between different genes (*petD* and *petB*; *psbE* and *psbF*) belonging to polycistronic transcripts. Some sites of coordinated events like *ndhD_116290* and *ndhD_116281*, *ndhB_95650* and *ndhB_95644 or ndhB_95419* and the *ndhB* intron could be very close but the others were separated by more than 100 nt.

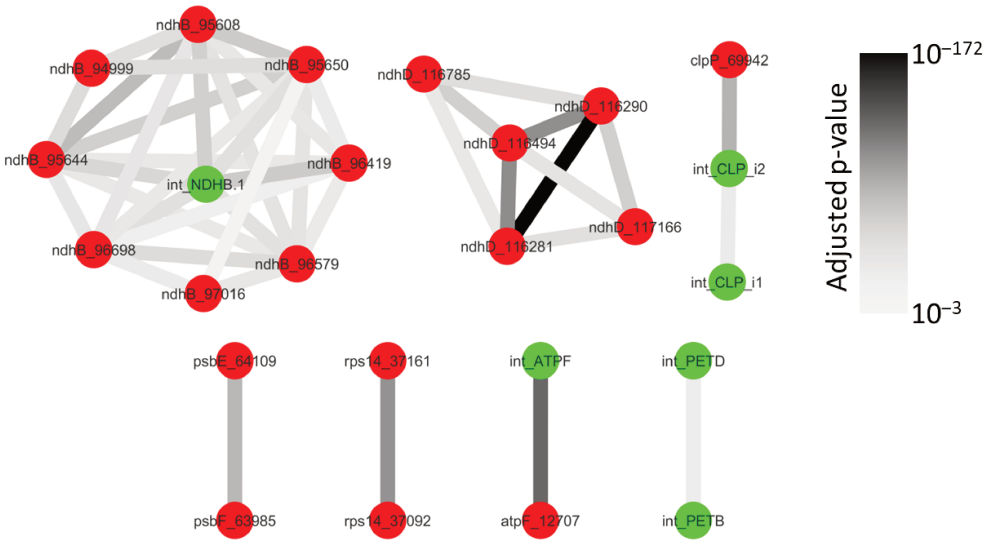


Figure 5. Network of splicing and editing coordination. Splicing events are shown in green and editing events in red. Dependent events are joined by an edge. The darkness of the edge is proportional to the adjusted *p*-value of the Exact Fisher test for the pool of the three replicates.

A more detailed analysis shows that maturation intermediates (TF (True-False) and FT (False-True) columns of Table S2) were always less frequent than expected for independent events for the 42 pairs of dependent events. This means that when one site was processed

the second one was more processed than expected randomly. In other words, there was co-maturation but no incompatibility, the maturation of one site increased the rate/speed of maturation of the other one. Furthermore, comparing the abundance of the intermediates of maturation offers the opportunity to find a preferred order of the maturation events (Figure 6).

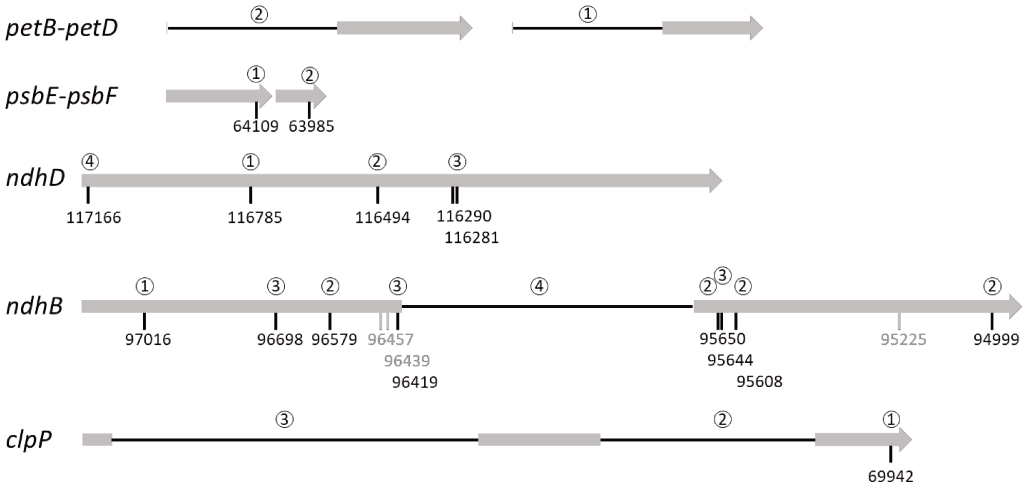


Figure 6. Proposed chronology of maturation events. Exons are shown as grey bars and introns as black lines. The editing sites are indicated by their genomic position. Grey editing sites are processed independently and thus are not included in the chronology. The preferred order of the maturation events is indicated by the numbers above the editing sites or introns.

This analysis suggests that RNA editing at *psbE_64109* generally occurred before RNA editing at *psbF_63985*, and that the splicing of *petD* preferentially occurred before the splicing of *petB*. The maturation of *clpP* generally started with RNA editing at *clpP_69942* followed by the splicing of the second intron and finished by the splicing of the first intron. For *ndhD*, the maturation preferentially started with RNA editing at *ndhD_116785* followed by *ndhD_116494* then both *ndhD_116290* and *ndhD_116281* to finish with *ndhD_117166*, the editing site creating the start codon of *ndhD*. For the *ndhB* transcript, the chronology of the maturation seemed more convoluted as three sites (96,457, 96,439 and 95,225) were edited independently of the other maturation events. RNA editing at *ndhB_97016* seemed to occur first followed by editing at the four sites 96,579, 95,650, 95,608 and 94,999. The maturation of *ndhB* ended with RNA editing at sites 96,698, 96,419, 95,644 and, probably slightly later, its splicing. To confirm the order deduced from the co-occurrence analysis for transcripts requiring more than three maturation events (i.e., *ndhD* and *ndhB*), we identified the reads covering all the maturation events and counted the frequency of the various intermediates (Table S3). Out of 413 intermediate reads, 311 (75%) were compatible with the proposed chronology of *ndhD* maturation. For *ndhB*, only 63 intermediate reads were identified. This number is too small to estimate the frequency of the 4096 possible intermediates (12 maturation events) but 35 (57%) were compatible with the proposed chronology. The reads corresponding to alternative maturation chronologies are probably the result of sequencing errors. Given the nanopore sequencing error rate at around 4% and the fact that this analysis considered five positions in *ndhD* and 12 in *ndhB*, only 81.6% (0.96^5) of the *ndhD* reads and 62% (0.96^{12}) of the *ndhB* reads are expected to be error-free at these positions.

This preferred chronology could theoretically be the result of kinetic differences between the different maturation events. For example, looking at two independent events, the one happening at a higher rate will likely occur first. This simple explanation is, however, incompatible with the observations. In particular, the decrease of the observed vs. expected TF and FT counts (columns delta_TF and delta_FT in Table S2) is not homogenous between TF and FT for most pairs of events. This shows that the positive effect (e.g.,

enhancement) provided by one maturation event to the other is not symmetrical. This asymmetry is involved in the chronology and could reinforce (at least in this case) any putative effects caused by a difference in processing rates. Finally, because of the number of maturation events jointly monitored for the *ndhB* (12 events) and *ndhD* transcripts (5 events, Table S3), the observation of a preferred chronology of maturation is extremely unlikely to be explained only by differences in maturation speed. We conclude that the observed preferred chronology of maturation is due, at least partly, to interactions between the processing events.

3. Discussion

Our protocol generates mostly full-length and stranded reads but transcripts longer than 2000–2500 nt are clearly under-represented. This bias is common to nuclear and plastid transcripts and several pieces of evidence (data not shown) strongly suggest that it is associated with the initial RNA-RNA ligation at the 3′ end of transcripts. It has indeed been described that the ligation step was sensitive to secondary structures at the 3′ end [35]. Maybe the denaturation step preceding the ligation step was not sufficient for long transcripts.

Following transcription, plastid transcripts undergo a complex array of modifications and maturation and the recent massive use of RNA-Seq based strategies has led to an unprecedented knowledge about its different steps. What is sorely lacking, however, is a global understanding of the interplay between RNA editing, splicing and processing.

Initially thought to be mainly independent [36,37], there are now more and more pieces of evidence for crosstalk between the different maturation steps [10,38–41]. Most of these results, however, have been obtained from experiments based on Sanger sequencing of a cDNA of interest, therefore limiting any potential generalization. Taking advantage of the development of nanopore sequencing, we systematically studied the link between individual RNA splicing and RNA editing events, at the plastome level.

Our results show that co-maturation of several sites tends to occur even when located far apart on their cognate transcript. This implies that all of the actors of these different processing events are grouped or co-localized, likely in the nucleoid [9].

Looking at specific links, splicing of the *atpF* intron and RNA editing at the *atpF_12707* site are clearly not independent (Figure 5). This was expected as AEF1, the PPR protein responsible for *atpF_12707* editing in *A. thaliana*, also facilitates *atpF* splicing [11]. Similarly, *clpP* intron 2 and *ndhB* splicing is enhanced by RNA editing in the cognate transcripts (Figure 6). Earlier studies have shown that some unspliced or unprocessed transcripts can already be fully edited [36,37] and this was interpreted as evidence that RNA editing is an early process, mainly occurring before splicing. Although RNA editing can be a prerequisite for splicing when it restores sequences or structures within the intron [42,43], this is an unlikely explanation here as the sites are located far from the identified splicing key elements [44]. A possibility put forward by Yap et al. [11] is that the binding of the RNA editing factor itself could have an indirect effect on splicing through the modification of RNA secondary structure or accessibility.

In agreement with the idea that RNA editing is an early maturation step, we only found marginal evidence that specific RNA editing sites could be influenced by splicing (Figure 5). This result is, however, probably dependent on our experimental model, *A. thaliana*. In various plants, *ndhA* intron removal was shown to be necessary for a *ndhA* editing site located close to the 3′ splice site. In this case, splicing is thought to create the RNA sequence necessary for the recognition of the RNA editing site [10], a site that is absent in *A. thaliana*. A similar situation has been described in *P. patens* mitochondria where *atp9* splicing is necessary to one editing site on the same transcript [15]. As shown for *clpP*, the splicing of one intron can also influence the splicing of another intron located on the same transcript (Figures 5 and 6). Experiments with intron deletions in tobacco have previously shown that the second intron in the *ycf3* transcript needs to be spliced before the first intron. In this case, splicing of the first intron was hypothesized to create a sequence masking essential

structural elements of the second intron [45]. Although *A. thaliana ycf3* structure is similar to tobacco, our analysis did not confirm such dependence in this transcript.

The dependence between RNA editing sites themselves has long been debated. For example, in vitro results on short fragments of the mitochondrial *atp4* RNA suggested that editing of individual sites did not influence others while *in organello* experiments with longer *cox2* transcripts showed a pattern of dependencies [46,47]. The identification of distal elements able to enhance RNA editing was also a strong argument against complete stochastic independence of the editing site recognition [41,48]. Our results show that both cases exist in the chloroplast. Editing site *ndhD_117166* generally requires earlier editing of the four other *ndhD* sites and *ndhB_97016* editing strongly influences editing at *ndhB_96698* and *ndhB_96579* sites. On the other hand editing at *ndhB_95225* seems autonomous and barely influences any other editing site (Figure 6).

Editing and splicing of organellar transcripts are required to get mRNA translated into functional proteins as editing often restores conserved amino acids [49] and splicing preserves the translation frame. However, the study of the translational landscape of *A. thaliana* mitochondria [50] or maize chloroplasts [21] showed that ribosomes were associated with partially edited transcripts and a small fraction of ribosomes were even associated with intronic sequences. Earlier chloroplast polysome purification experiments also showed that transcripts of the *psbB* gene cluster containing the *petB* or *petD* intron could still be translated for other genes [51]. This suggests that partially mature (especially partially edited) transcripts can access the organelle translation machinery. In addition to the dependence of some maturations events, our results showed that they could be ordered (Figure 6). In this chronology, splicing events seemed to occur later than editing events: the splicing of *ndhB* occurred after editing at most sites and splicing of *clpP* occurred after its editing. Even if the chronology was not clear from our results, Yap et al. also showed that *atpF* editing probably occurs before its splicing [11]. In addition, events located at the 5′ end of the transcripts tended to be later than the others. That is clearly the case for *clpP* and *ndhD* associated transcripts. In *ndhD*, RNA editing at *ndhD_117166* was generally the last maturation event and is required to create the start codon and thus to allow the translation of the transcript. This succession of the maturation events where splicing and 5′ end events tend to be last could be a way to ensure the complete (or at least a better) maturation of the transcripts before initiating their translation. Although there is currently no known underlying mechanism to support this hypothesis, it could at least explain why partially edited RNA editing sites are generally more edited in ribosome-associated RNAs than on the steady-state pool of transcripts [21,50]. In addition, it could also explain why sites restoring cryptic start codons have variable but often lower editing rates [49,52].

Despite the modest size of the dataset and its rather simple analysis, the results presented in this study highlight the potential of long-read RNA-Seq for the analysis of plastid and mitochondrial transcriptomes. Even if the molecular protocol still needs improvements to capture the longest transcripts, it provides access to the full complexity of this transcriptome and already showed numerous links between splicing and editing. For analytical reasons, we did not include the analysis of processing in this study but nanopore RNA-Seq is suited for this type of analysis (Figure 3) and we are developing the required bioinformatical and statistical tools. A potential improvement of our strategy would be to directly sequence the chloroplastic RNAs, without performing any cDNA synthesis. This would give access to the various epitranscriptomics marks [53] that are now known to be pervasive in chloroplastic RNAs [54] and whose interactions have, for example, been shown to be important in human diseases [55]. With this complete toolbox, we anticipate it will be possible to explore the impact of growth conditions and/or mutants or compare the nucleoid- or polysome-associated transcriptome to further decipher the molecular mechanisms controlling plastid but also mitochondrial gene expression.

4. Materials and Methods

4.1. Plant Growth and RNA Extraction

Col-0 plants were grown in soil in growth chambers with 16 h of light per day at 20 °C for 5 weeks. Fifteen minutes before the onset of lights, 2 adult leaves were flash-frozen in liquid nitrogen. Total RNA was extracted using Nucleozol (Macherey-Nagel, Hoerdt, France) followed by a purification with AMPure RNA XP beads (Beckman Coulter, Villepinte, France). Three independent experiments were performed to get three biological replicates.

4.2. Nanopore Sequencing

The step-by-step protocol for the construction of the sequencing library is available online at https://forgemia.inra.fr/guillem.rigaill/nanopore_chloro (accessed on 18 October 2021). Briefly, 10 fmoles of the RNA oligo /5Phos/rNrNrNrNrUrGrArArUrGrCrArArCrArCrUrUrCrUrGrUrArC/3InvdT/ (IDT Technologies, Leuven, Belgium) was ligated to the 3′ end of 100 ng of total RNA using 10 U of T4 RNA ligase 1 (NEB, Evry, France). Ligated RNA was depleted of rRNA using the QIAseq FastSelect -rRNA Plant Kit (QIAGEN, Les Ulis, France) before a full-length cDNA synthesis using the SMARTScribe™ Reverse Transcriptase (Takara, Saint Germain en Laye, France) and the oligos AAGCAGTGGTATCAACGCAGAGTACrGrG + G and AAGCAGTGGTATCAACGCAGAGTACGTACAGAAGTGTTGCATTC (IDT Technologies, Leuven, Belgium). Full-length cDNAs were amplified with the SeqAmp DNA Polymerase (Takara, Saint Germain en Laye, France) using the AAGCAGTGGTATCAACGCAGAGTAC primer and purified with AMPure XP beads (Beckman-Coulter, Villepinte, France). 35 fmoles of amplified cDNAs were converted to a nanopore sequencing library with the PCR barcoding kit (Oxford Nanopore Technologies, Oxford, UK) and then sequenced on an R10.3 MinIon flow-cell (Oxford Nanopore Technologies, Oxford, UK).

4.3. Bioinformatics and Statistical Analyses

The raw data were base-called and demultiplexed with Guppy v5.0.7 (Oxford Nanopore Technologies) using the dna_r10.3_450 bps_hac model. Reads were then oriented using the in-house script "fastq_processing.sh" which uses LAST v1179 [56] and CUTADAPT v2.10 [57] and is available online at https://forgemia.inra.fr/guillem.rigaill/nanopore_chloro (accessed on 18 October 2021). They were mapped on the col-0 genomic sequence with Minimap2 v2.1 [58]. Transcript body coverage and strandedness were measured with the RSeQC v3.0 package [59]. The Illumina samples used to compare were the dyw2_HE replicates 1 to 3 (NCBI GEO accession numbers GSM2677518, GSM2677519 and GSM2677520) from Guillaumot et al. [33]. The plants used for these samples were grown in the same growth chambers and the sequencing libraries were constructed with the Illumina TruSeq stranded total RNA with Ribozero plant kit.

The maturation events analyzed in this study are listed in Table S1. They include the editing sites detected by Ruwe et al. [12] and the introns of protein-coding genes. The tRNA introns were omitted because the mature tRNAs are excluded from the sequencing library during sizing. This information is used to annotate each read for every maturation event according to three modalities: mature site, not mature site, and not read site. The latter allows taking insertions/deletions into account which are frequent in nanopore datasets. For each pair of events jointly observed the following configurations are listed and counted in a contingency table: mature/mature, mature/immature, immature/mature, and immature/immature. The dependency of two events, based on the contingency table, is tested using a Fisher exact test and the p-values were adjusted with an FDR [60]. Only pairs of events characterized by an adjusted p-value < 0.1 in at least 2 of the 3 replicates and an adjusted p-value < 0.005 on the pool of the 3 replicates were considered significant. Commented R scripts to annotate reads, create contingency table, perform Fisher's exact tests and generate the result table are available online at https://forgemia.inra.fr/guillem.rigaill/nanopore_chloro (accessed on 18 October 2021).

The splicing and editing rates were measured from pooled reads of the 3 replicates. Virtual Northern blots were generated by extracting the length of the reads mapping from position 75700 to position 76000 on the Watson strand (*petB*), from position 77200 to position 77500 on the Watson strand (*petD*), from position 74487 to position 74706 on the Watson strand (*psbH*) or from position 74254 to position 74378 on the Crick strand (*psbN*) using samtools [61] and bedtools [62]. The size distributions were normalized by setting the value of the most abundant read length to 100. These distributions were converted into virtual Northern blots with the "vNB.py" python script available online at https://forgemia.inra.fr/guillem.rigaill/nanopore_chloro (accessed on 18 October 2021).

Supplementary Materials: The following are available online at https://www.mdpi.com/article/10.3390/ijms222011297/s1.

Author Contributions: Conceptualization, M.G., B.C. and E.D.; Data curation, A.L., G.R. and E.D.; Formal analysis, A.L., C.S. and G.R.; Funding acquisition, G.R., B.C. and E.D.; Investigation, M.G. and E.D.; Methodology, G.R., B.C. and E.D.; Project administration, E.D.; Resources, E.D.; Software, M.G., A.L., C.S., T.B. and G.R.; Supervision, E.D.; Writing—original draft, M.G., B.C. and E.D.; Writing—review and editing, A.L., G.R., B.C. and E.D. All authors have read and agreed to the published version of the manuscript.

Funding: The IPS2 benefits from the support of Saclay Plant Sciences-SPS (ANR-17-EUR-0007). This work was supported by a grant from the Université Evry-Val d'Essonne to E.D., by the ANR-20-CE20-0004 JOAQUIN to B.C. and by the Evry Genopole to G.R.

Data Availability Statement: The fastq files are available from the NCBI SRA database under the accession number PRJNA748959.

Acknowledgments: We thank Etienne Sandré-Chardonnal for the python script generating the picture of the virtual Northern blot and Amber M Hotto for her comments and proofreading of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Green, B.R. Chloroplast genomes of photosynthetic eukaryotes. *Plant J.* **2011**, *66*, 34–44. [CrossRef] [PubMed]
2. Maier, U.G.; Bozarth, A.; Funk, H.T.; Zauner, S.; Rensing, S.A.; Schmitz-Linneweber, C.; Börner, T.; Tillich, M. Complex chloroplast RNA metabolism: Just debugging the genetic programme? *BMC Biol.* **2008**, *6*, 36. [CrossRef] [PubMed]
3. Stern, D.B.; Goldschmidt-Clermont, M.; Hanson, M.R. Chloroplast RNA Metabolism. *Annu. Rev. Plant Biol.* **2010**, *61*, 125–155. [CrossRef] [PubMed]
4. Barkan, A. Expression of Plastid Genes: Organelle-Specific Elaborations on a Prokaryotic Scaffold. *Plant Physiol.* **2011**, *155*, 1520–1532. [CrossRef] [PubMed]
5. de Longevialle, A.F.; Small, I.D.; Lurin, C. Nuclearly-encoded splicing factors implicated in RNA splicing in higher plant organelles. *Mol. Plant* **2010**, *3*, 691–705. [CrossRef]
6. Sun, T.; Bentolila, S.; Hanson, M.R. The Unexpected Diversity of Plant Organelle RNA Editosomes. *Trends Plant Sci.* **2016**, *21*, 962–973. [CrossRef]
7. Germain, A.; Hotto, A.M.; Barkan, A.; Stern, D.B. RNA processing and decay in plastids. *Wiley Interdiscip. Rev. RNA* **2013**, *4*, 295–316. [CrossRef]
8. MacIntosh, G.C.; Castandet, B. Organellar and Secretory Ribonucleases: Major Players in Plant RNA Homeostasis. *Plant Physiol.* **2020**, *183*, 1438. [CrossRef]
9. Majeran, W.; Friso, G.; Asakura, Y.; Qu, X.; Huang, M.; Ponnala, L.; Watkins, K.P.; Barkan, A.; van Wijk, K.J. Nucleoid-enriched proteomes in developing plastids and chloroplasts from maize leaves: A new conceptual framework for nucleoid functions. *Plant Physiol.* **2012**, *158*, 156–189. [CrossRef]
10. Schmitz-Linneweber, C.; Tillich, M.; Herrmann, R.G.; Maier, R.M. Heterologous, splicing-dependent RNA editing in chloroplasts: Allotetraploidy provides trans-factors. *EMBO J.* **2001**, *20*, 4874–4883. [CrossRef]
11. Yap, A.; Kindgren, P.; Colas Des Francs-Small, C.; Kazama, T.; Tanz, S.K.; Toriyama, K.; Small, I. AEF1/MPR25 is implicated in RNA editing of plastid atpF and mitochondrial nad5, and also promotes atpF splicing in Arabidopsis and rice. *Plant J.* **2015**, *81*, 661–669. [CrossRef]
12. Ruwe, H.; Castandet, B.; Schmitz-Linneweber, C.; Stern, D.B. Arabidopsis chloroplast quantitative editotype. *FEBS Lett.* **2013**, *587*, 1429–1433. [CrossRef]

13. Marechal-Drouard, L.; Cosset, A.; Remacle, C.; Ramamonjisoa, D.; Dietrich, A. A single editing event is a prerequisite for efficient processing of potato mitochondrial phenylalanine tRNA. *Mol. Cell. Biol.* **1996**, *16*, 3504–3510. [CrossRef]
14. Malbert, B.; Burger, M.; Lopez-Obando, M.; Baudry, K.; Launay-Avon, A.; Härtel, B.; Verbitskiy, D.; Jörg, A.; Berthomé, R.; Lurin, C.; et al. The Analysis of the Editing Defects in the dyw2 Mutant Provides New Clues for the Prediction of RNA Targets of Arabidopsis E+-Class PPR Proteins. *Plants* **2020**, *9*, 280. [CrossRef]
15. Ichinose, M.; Sugita, C.; Yagi, Y.; Nakamura, T.; Sugita, M. Two DYW subclass PPR proteins are involved in RNA editing of ccmFc and atp9 transcripts in the moss Physcomitrella patens: First complete set of PPR editing factors in plant mitochondria. *Plant Cell Physiol.* **2013**, *54*, 1907–1916. [CrossRef] [PubMed]
16. Schallenberg-Rüdinger, M.; Kindgren, P.; Zehrmann, A.; Small, I.; Knoop, V. A DYW-protein knockout in Physcomitrella affects two closely spaced mitochondrial editing sites and causes a severe developmental phenotype. *Plant J.* **2013**, *76*, 420–432. [CrossRef] [PubMed]
17. Castandet, B.; Hotto, A.M.; Strickler, S.R.; Stern, D.B. ChloroSeq, an Optimized Chloroplast RNA-Seq Bioinformatic Pipeline, Reveals Remodeling of the Organellar Transcriptome Under Heat Stress. *G3 Genes Genomes Genet.* **2016**, *6*, 2817–2827. [CrossRef] [PubMed]
18. Michel, E.J.S.S.; Hotto, A.M.; Strickler, S.R.; Stern, D.B.; Castandet, B. A Guide to the Chloroplast Transcriptome Analysis Using RNA-Seq. *Methods Mol. Biol.* **2018**, *1829*, 295–313. [CrossRef] [PubMed]
19. Malbert, B.; Rigaill, G.; Brunaud, V.; Lurin, C.; Delannoy, E. Bioinformatic Analysis of Chloroplast Gene Expression and RNA Posttranscriptional Maturations Using RNA Sequencing. In *Methods in Molecular Biology*; Humana Press: New York, NY, USA, 2018; Volume 1829, pp. 279–294.
20. Castandet, B.; Germain, A.; Hotto, A.M.; Stern, D.B. Systematic sequencing of chloroplast transcript termini from Arabidopsis thaliana reveals >200 transcription initiation sites and the extensive imprints of RNA-binding proteins and secondary structures. *Nucleic Acids Res.* **2019**, *47*, 11889–11905. [CrossRef] [PubMed]
21. Chotewutmontri, P.; Barkan, A. Dynamics of Chloroplast Translation during Chloroplast Differentiation in Maize. *PLoS Genet.* **2016**, *12*, e1006106. [CrossRef]
22. Ruwe, H.; Wang, G.; Gusewski, S.; Schmitz-Linneweber, C. Systematic analysis of plant mitochondrial and chloroplast small RNAs suggests organelle-specific mRNA stabilization mechanisms. *Nucleic Acids Res.* **2016**, *44*, 7406–7417. [CrossRef] [PubMed]
23. Zhelyazkova, P.; Sharma, C.M.; Forstner, K.U.; Liere, K.; Vogel, J.; Borner, T. The Primary Transcriptome of Barley Chloroplasts: Numerous Noncoding RNAs and the Dominating Role of the Plastid-Encoded RNA Polymerase. *Plant Cell* **2012**, *24*, 123–136. [CrossRef]
24. Cui, J.; Shen, N.; Lu, Z.; Xu, G.; Wang, Y.; Jin, B. Analysis and comprehensive comparison of PacBio and nanopore-based RNA sequencing of the Arabidopsis transcriptome. *Plant Methods* **2020**, *16*, 85. [CrossRef] [PubMed]
25. Long, Y.; Jia, J.; Mo, W.; Jin, X.; Zhai, J. FLEP-seq: Simultaneous detection of RNA polymerase II position, splicing status, polyadenylation site and poly(A) tail length at genome-wide scale by single-molecule nascent RNA sequencing. *Nat. Protoc.* **2021**, *16*, 4355–4381. [CrossRef] [PubMed]
26. Jia, J.; Long, Y.; Zhang, H.; Li, Z.; Liu, Z.; Zhao, Y.; Lu, D.; Jin, X.; Deng, X.; Xia, R.; et al. Post-transcriptional splicing of nascent RNA contributes to widespread intron retention in plants. *Nat. Plants* **2020**, *6*, 780–788. [CrossRef]
27. Zhu, Y.; Machleder, E.; Chenchik, A.; Li, R.; Siebert, P. Reverse transcriptase template switching: A SMART approach for full-length cDNA library construction. *Biotechniques* **2001**, *30*, 892–897. [CrossRef]
28. Schuster, G.; Stern, D. RNA Polyadenylation and Decay in Mitochondria and Chloroplasts. *Prog. Mol. Biol. Transl. Sci.* **2009**, *85*, 393–422. [CrossRef]
29. Hotto, A.M.; Castandet, B.; Gilet, L.; Higdon, A.; Condon, C.; Stern, D.B. Arabidopsis Chloroplast Mini-Ribonuclease III Participates in rRNA Maturation and Intron Recycling. *Plant Cell* **2015**, *27*, 724–740. [CrossRef]
30. Felder, S.; Meierhoff, K.; Sane, A.P.; Meurer, J.; Driemel, C.; Plücken, H.; Klaff, P.; Stein, B.; Bechtold, N.; Westhoff, P. The Nucleus-Encoded HCF107 Gene of Arabidopsis Provides a Link between Intercistronic RNA Processing and the Accumulation of Translation-Competent psbH Transcripts in Chloroplasts. *Plant Cell* **2001**, *13*, 2127. [CrossRef]
31. Stoppel, R.; Meurer, J. Complex RNA metabolism in the chloroplast: An update on the psbB operon. *Planta* **2013**, *237*, 441–449. [CrossRef]
32. Lezhneva, L.; Meurer, J. The nuclear factor HCF145 affects chloroplast psaA-psaB-rps14 transcript abundance in Arabidopsis thaliana. *Plant J.* **2004**, *38*, 740–753. [CrossRef]
33. Guillaumot, D.; Lopez-Obando, M.; Baudry, K.; Avon, A.; Rigaill, G.; Falcon De Longevialle, A.; Broche, B.; Takenaka, M.; Berthomé, R.; De Jaeger, G.; et al. Two interacting PPR proteins are major Arabidopsis editing factors in plastid and mitochondria. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 8877–8882. [CrossRef] [PubMed]
34. Rüdinger, M.; Funk, H.T.; Rensing, S.A.; Maier, U.G.; Knoop, V. RNA editing: Only eleven sites are present in the Physcomitrella patens mitochondrial transcriptome and a universal nomenclature proposal. *Mol. Genet. Genomics* **2009**, *281*, 473–481. [CrossRef] [PubMed]
35. Zhuang, F.; Fuchs, R.T.; Sun, Z.; Zheng, Y.; Robb, G.B. Structural bias in T4 RNA ligase-mediated 3′-adapter ligation. *Nucleic Acids Res.* **2012**, *40*, e54. [CrossRef] [PubMed]
36. Freyer, R.; Hoch, B.; Neckermann, K.; Maier, R.M.; Kössel, H. RNA editing in maize chloroplasts is a processing step independent of splicing and cleavage to monocistronic mRNAs. *Plant J.* **1993**, *4*, 621–629. [CrossRef] [PubMed]

37. Ruf, S.; Zeltz, P.; Kössel, H. Complete RNA editing of unspliced and dicistronic transcripts of the intron-containing reading frame IRF170 from maize chloroplasts. *Proc. Natl. Acad. Sci. USA* **1994**, *91*, 2295–2299. [CrossRef]
38. Maréchal-Drouard, L.; Kumar, R.; Remacle, C.; Small, I. RNA editing of larch mitochondrial tRNA His precursors is a prerequisite for processing. *Nucleic Acids Res.* **1996**, *24*, 3229–3234. [CrossRef]
39. Tillich, M.; Hardel, S.L.; Kupsch, C.; Armbruster, U.; Delannoy, E.; Gualberto, J.M.; Lehwark, P.; Leister, D.; Small, I.D.; Schmitz-Linneweber, C. Chloroplast ribonucleoprotein CP31A is required for editing and stability of specific chloroplast mRNAs. *Proc. Natl. Acad. Sci. USA* **2009**, *106*, 6002–6007. [CrossRef]
40. Karcher, D.; Bock, R. Site-selective inhibition of plastid RNA editing by heat shock and antibiotics: A role for plastid translation in RNA editing. *Nucleic Acids Res.* **1998**, *26*, 1185–1190. [CrossRef]
41. Takenaka, M.; Neuwirt, J.; Brennicke, A. Complex cis-elements determine an RNA editing site in pea mitochondria. *Nucleic Acids Res.* **2004**, *32*, 4137–4144. [CrossRef]
42. Castandet, B.; Choury, D.; Bégu, D.; Jordana, X.; Araya, A. Intron RNA editing is essential for splicing in plant mitochondria. *Nucleic Acids Res.* **2010**, *38*, 7112–7121. [CrossRef]
43. Farré, J.-C.; Aknin, C.; Araya, A.; Castandet, B. RNA Editing in Mitochondrial Trans-Introns Is Required for Splicing. *PLoS ONE* **2012**, *7*, e52644. [CrossRef]
44. Vogel, J.; Börner, T. Lariat formation and a hydrolytic pathway in plant chloroplast group II intron splicing. *EMBO J.* **2002**, *21*, 3794–3803. [CrossRef]
45. Petersen, K.; Schöttler, M.A.; Karcher, D.; Thiele, W.; Bock, R. Elimination of a group II intron from a plastid gene causes a mutant phenotype. *Nucleic Acids Res.* **2011**, *39*, 5181–5192. [CrossRef]
46. Verbitskiy, D.; Takenaka, M.; Neuwirt, J.; Van Der Merwe, J.A.; Brennicke, A. Partially edited RNAs are intermediates of RNA editing in plant mitochondria. *Plant J.* **2006**, *47*. [CrossRef]
47. Castandet, B.; Araya, A. The RNA editing pattern of cox2 mRNA is affected by point mutations in plant mitochondria. *PLoS ONE* **2011**, *6*, e20867. [CrossRef] [PubMed]
48. Staudinger, M.; Bolle, N.; Kempken, F. Mitochondrial electroporation and in organello RNA editing of chimeric atp6 transcripts. *Mol. Genet. Genom.* **2005**, *273*, 130–136. [CrossRef]
49. Small, I.D.; Schallenberg-Rüdinger, M.; Takenaka, M.; Mireau, H.; Ostersetzer-Biran, O. Plant organellar RNA editing: What 30 years of research has revealed. *Plant J.* **2019**, *101*, 1040–1056. [CrossRef] [PubMed]
50. Planchard, N.; Bertin, P.; Quadrado, M.; Dargel-Graffin, C.; Hatin, I.; Namy, O.; Mireau, H. The translational landscape of Arabidopsis mitochondria. *Nucleic Acids Res.* **2018**, *46*, 6218. [CrossRef] [PubMed]
51. Barkan, A. Proteins encoded by a complex chloroplast transcription unit are each translated from both monocistronic and polycistronic mRNAs. *EMBO J.* **1988**, *7*, 2637–2644. [CrossRef]
52. Li, M.; Xia, L.; Zhang, Y.; Niu, G.; Li, M.; Wang, P.; Zhang, Y.; Sang, J.; Zou, D.; Hu, S.; et al. Plant editosome database: A curated database of RNA editosome in plants. *Nucleic Acids Res.* **2019**, *47*, D170. [CrossRef]
53. Anreiter, I.; Mir, Q.; Simpson, J.T.; Janga, S.C.; Soller, M. New Twists in Detecting mRNA Modification Dynamics. *Trends Biotechnol.* **2021**, *39*, 72–89. [CrossRef] [PubMed]
54. Manavski, N.; Vicente, A.; Chi, W.; Meurer, J. The chloroplast epitranscriptome: Factors, sites, regulation, and detection methods. *Genes* **2021**, *39*, 72–89.
55. Kadumuri, R.V.; Janga, S.C. Epitranscriptomic Code and Its Alterations in Human Disease. *Trends Mol. Med.* **2018**, *24*, 886–903. [CrossRef] [PubMed]
56. Kiełbasa, S.M.; Wan, R.; Sato, K.; Horton, P.; Frith, M.C. Adaptive seeds tame genomic sequence comparison. *Genome Res.* **2011**, *21*, 487–493. [CrossRef] [PubMed]
57. Martin, M. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet. J.* **2011**, *17*, 10–12. [CrossRef]
58. Li, H. Minimap2: Pairwise alignment for nucleotide sequences. *Bioinformatics* **2018**, *34*, 3094–3100. [CrossRef]
59. Wang, L.; Nie, J.; Sicotte, H.; Li, Y.; Eckel-Passow, J.E.; Dasari, S.; Vedell, P.T.; Barman, P.; Wang, L.; Weinshiboum, R.; et al. Measure transcript integrity using RNA-seq data. *BMC Bioinform.* **2016**, *17*, 1–16. [CrossRef]
60. Benjamini, Y.; Hochberg, Y. Controlling The False Discovery Rate—A Practical And Powerful Approach To Multiple Testing. *J. R. Stat. Soc. B* **1995**, *57*, 289–300. [CrossRef]
61. Li, H.; Handsaker, B.; Wysoker, A.; Fennell, T.; Ruan, J.; Homer, N.; Marth, G.; Abecasis, G.; Durbin, R. Subgroup, 1000 Genome Project Data Processing The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **2009**, *25*, 2078. [CrossRef]
62. Quinlan, A.R.; Hall, I.M. BEDTools: A flexible suite of utilities for comparing genomic features. *Bioinformatics* **2010**, *26*, 841. [CrossRef]

D.2 comaturationTrackeR (1^{st} version)

Here below is the bachelor's thesis of Chloé Seyman, which I co-supervised alongside Guillem Rigaill over a three-month period at the Laboratoire de Mathématiques et Modélisation d'Évry. This thesis partially describes the initial version of ComaturationTracker, presented as an R pipeline available at the following link : https://forgemia.inra.fr/guillem.rigaill/nanopore_chloro.

RAPPORT DE STAGE

Analyse et visualisation de données RNA-Seq chloroplastiques Nanopores pour l'analyse de transcrits entiers

Chloé SEYMAN
L3 Sciences de la Vie – Informatique

Sous la direction de Guillem RIGAILL et Arnaud LIEHRMANN

Université d'Evry Val d'Essonne Paris-Saclay
Laboratoire de Mathématiques et de Modélisation d'Evry (LaMME)

Année scolaire 2020-2021

REMERCIEMENTS

Je tiens tout d'abord à remercier Monsieur Guillem RIGAILL pour son encadrement et sa pédagogie tout au long de ce stage, ainsi que pour sa bienveillance. Je remercie également Monsieur Arnaud LIERHMANN pour son aide et sa pédagogie, notamment sur la partie informatique. Je remercie également le LaMME pour m'avoir accueillie dans sa structure et pour m'avoir permise de venir quelques jours sur le site malgré le contexte sanitaire difficile. Je tiens également à remercier l'équipe de l'IPS2, à savoir Monsieur Etienne DELANNOY, Monsieur Benoît CASTANDET et Madame Marine GUILCHER, pour m'avoir permise d'assister aux réunions d'équipe et pour leur bienveillance. Je remercie enfin toute l'équipe du LaMME pour leurs discussions autour des mathématiques et pour leur sympathie.

I - INTRODUCTION

Ce stage a été réalisé dans le cadre de ma troisième année de double licence Sciences de la Vie & Informatique à l'Université d'Evry Val d'Essonne (Paris-Saclay), au Laboratoire de Mathématiques et de Modélisation d'Evry (LaMME). D'une durée de 50 jours, répartis du lundi 26 avril au vendredi 16 juillet 2021, ce stage est réalisé à la fois en présentiel et en distanciel, avec une fréquence d'une journée par semaine au LaMME, dû au contexte sanitaire particulier de Covid-19. A ce stade du stage, seules quatre semaines effectives ont été réalisées, dû aux examens décalés au mois de mai et aux cours de biologie qui n'étaient pas terminés. Ce rapport résume donc le travail effectué sur 19 jours effectifs de stage.

Le sujet du stage portait sur l'étude du transcriptome et plus précisément de la co-maturation d'ARNm chloroplastiques d'Arabidopsis Thaliana à partir de données long-reads (Nanopores), par le biais d'analyses bio-informatiques, réalisées principalement sous Rstudio. Ce sujet de recherche est le fruit d'une collaboration entre Guillem Rigaill et Arnaud Liehrmann, avec Etienne Delannoy, Benoît Castandet et Marine Guilcher, trois biologistes spécialistes du chloroplaste travaillant à l'Institut des Sciences des Plantes de Paris-Saclay (IPS2).

Dans un premier temps, je poserai le contexte de l'étude dans laquelle j'ai réalisé mon stage en introduisant des généralités sur le génome chloroplastique, les évènements de maturation des transcrits, le séquençage nanopore et les fichiers BAM. Dans un second temps, j'exposerai la problématique sur laquelle s'est basé mon stage, c'est-à-dire la dépendance ou non entre les évènements de maturation chez A. thaliana, les données nanopores sur lesquelles j'ai travaillé, ainsi que les méthodes utilisées pour tenter de répondre à cette problématique, notamment sous RStudio. Enfin, j'exposerai quelques résultats obtenus concernant les évènements d'édition chez les transcrits d'A. thaliana.

I – CONTEXTE DE L'ETUDE

A) Généralités sur le génome chloroplastique d'*Arabidopsis thaliana*

L'une des particularités des plantes par rapport aux autres espèces peuplant la planète, est qu'elles possèdent trois génomes : un génome nucléaire, un génome mitochondrial, et un génome chloroplastique. Ce dernier, appelé également plastome ou génome plastidial, constitue donc le génome issu du chloroplaste. Il est d'une longueur bien inférieure aux deux autres types de génome. S'il a longtemps été décrit dans la littérature comme étant circulaire, des recherches plus récentes ont identifié le génome chloroplastique comme étant en réalité linéaire chez la majorité des espèces, in vivo (70%). Chez A. thaliana, ce génome est d'une longueur de 154 478 paires de bases, et code pour environ 132 gènes (Baudry K.,2019).

Les ARN du génome chloroplastique sont majoritairement polycistroniques. Sur les ARN polycistroniques, les gènes sont groupés en unités transcriptionnelles. Une fois transcrits, les ARN subissent des modifications post-transcriptionnelles : ils sont tout d'abord clivés en monocistrons (cleavage), puis maturés (Stern D. B., 2010). L'étape de maturation comprend trois types d'évènements : une maturation des extrémités des transcrits (processing), un épissage (splicing) et une édition de certaines bases nucléotidiques (editing). Ces étapes de maturation, systématiques, ont été identifiées comme primordiales pour la stabilité des transcrits et leur permettre d'être fonctionnels.

B) Maturation des transcrits

1) Edition des transcrits (editing)

L'édition est la modification ponctuelle d'une base de la séquence d'ARN. Il existe plusieurs éditions possibles, mais la plus répandue et la plus étudiée est la désamination de cytosines en uraciles (édition C vers U). L'édition d'une base peut avoir un impact sur la séquence protéique, en créant un codon start par exemple, ou peut être silencieuse (pas de changement dans la protéine codée, car codon synonyme ou alors la base éditée est dans une région intergénique). L'édition est catalysée par l'éditosome, lequel est composé de protéines nucléaires, dont les PPR (Small I. D, 2013).

Figure 1 : Schéma d'un brin d'ARNm (en bleu) s'alignant sur le génome de référence (en violet) et présentant une édition C vers U

2) Epissage des introns (splicing)

Les introns de type I et II sont excisés par processus auto-catalytique (repliements) ou enzymatique, avec l'implication de maturases et de facteurs d'épissage (splicing factors). Chez A. thaliana, nous dénombrons 20 introns excisés (14 dans les ARNm et 6 dans les ARNt). Une fois les introns excisés, les exons sont ensuite ligués entre eux. L'ARNm est donc plus court.

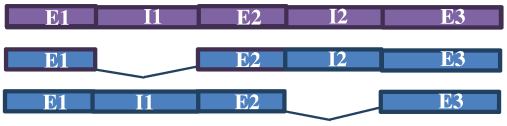


Figure 2 : Schéma de deux brins d'ARNm (en bleu) s'alignant sur le génome de référence (en violet) et présentant respectivement un épissage de l'intron 1 (I1) et un épissage de l'intron 2 (I2). Les lettres E correspondent aux exons.

3) Maturation des extrémités (formation des UTR) (5' – 3' processing)

Les extrémités sont digérées par deux types de ribonucléases d'origine nucléaire, les endonucléases et les exonucléases, afin de former des UTR matures. Les endonucléases reconnaissent les séquences cibles à l'intérieur d'un ARN pour les couper (RNAse E et RNAse J), tandis que les exonucléases digèrent l'ARN à partir de ses extrémités (PNPase et RNR1). La maturation des extrémités apporte davantage de stabilité à l'ARN.

Figure 3 : Schéma d'un brin d'ARNm (en bleu) s'alignant sur le génome de référence (en violet) et présentant des maturations aux extrémités du brin, représentées en vert

Une question se pose alors : celle de la dépendance entre différents évènements de maturation, comme par exemple entre une base éditée et un site épissé. Par exemple, nous pourrions imaginer que l'édition d'une base soit systématiquement associée à l'épissage d'un intron (voir Figure 4 ci-dessous).

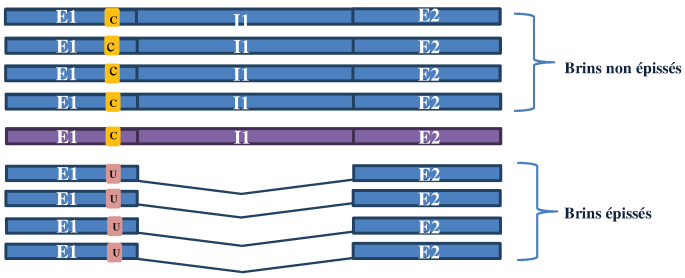


Figure 4 : Schéma de brins d'ARNm (en bleu) s'alignant sur le génome de référence (en violet), et présentant deux évènements de maturation : une édition C vers U et un épissage de l'intron 1 (I1). Dans cet exemple fictif, l'évènement d'édition C vers U observé dans l'exon 1 semble être lié à l'évènement d'épissage de l'intron 1. Cette dépendance doit alors être testée au moyen de tests statistiques.

C) Données nanopores (long-reads)

L'analyse des transcrits chloroplastiques et des évènements de maturation qu'ils subissent peut se faire par une analyse par RT-PCR, par Northern Blot, ou encore par RNA-Seq. Cette dernière technique consiste à séquencer l'ensemble des ARN (Next-Generation Sequencing). Les séquences produites se présentent généralement sous la forme de fragments courts d'ADN complémentaires d'environ 100 paires de bases, appelés « short-reads ». Des fragments plus longs, de plusieurs milliers de paires de bases, appelés « long-reads », peuvent également être générés suite à l'utilisation d'une technique de séquençage particulière : le séquençage nanopore (Oxford Nanopore Technologies). Cette technique, relativement récente et très prometteuse, consiste à faire passer les acides nucléiques d'intérêt à travers des nanopores, c'est-à-dire des trous dont la taille est de l'ordre du nanomètre, en utilisant un courant électrique (Grünberger F., 2020). Chaque base traversée est lue une par une, ce qui permet de séquencer le brin sur une grande longueur, voire dans son entièreté. L'un des intérêts des données nanopores est que de par leur longueur, elles permettent de voir sur un même read plusieurs évènements de maturation, par exemple édition et épissage, et donc d'étudier la dépendance entre ces évènements (Figure 2).

Figure 2 : Visualisation de deux évènements d'édition sur des données short-reads (à gauche) et des données long-reads (à droite). Le phénomène de co-évènement est ici uniquement identifiable sur le long-read.

Toutefois, un inconvénient du séquençage nanopore est son taux d'erreur de l'ordre de 4%, soit supérieur à celui des données short-reads (0,1 à 1% d'erreur). Il conviendra donc de prendre ce taux d'erreur en considération lors de l'interprétation finale des résultats, au mois de juillet, en particulier pour les éditions.

D) Fichiers BAM

A la suite du séquençage, les résultats des alignements des reads avec la séquence de référence sont stockés au format fastq sous forme brute, puis au format SAM, BAM ou BED. Le fichier BAM est la forme binaire et compressée du fichier SAM et contient les résultats de l'alignement du long read avec la séquence de référence. Ainsi, chaque ligne du fichier BAM correspond à un alignement entre un read et la séquence de référence, et contient pour chaque read son identifiant unique (qname), le nom de sa séquence de référence (rname), le brin sur

lequel se mappe le read (strand), la position de la première base du read par rapport à la séquence de référence (pos), la taille du read (width), sa séquence entière (seq) et également le cigar (cigar) qui donne des informations détaillées et précises sur l'alignement telles que les délétions ou les insertions. (Annexe 1)

Le contenu des fichiers BAM ou SAM peut être visualisé à l'aide de logiciels informatiques tels qu'IGV afin d'obtenir une représentation visuelle de notre alignement de séquences, ou Rstudio pour manipuler les données avec l'intermédiaire du package Rsamtools. D'autres packages sont également très utiles dans la manipulation de données génomiques sous R, tels que le package rtracklayer qui permet d'importer des fichiers Gff, et le package GenomicRanges qui permet notamment de générer des objets GRanges, lesquels sont très utiles pour effectuer des requêtes sur des reads ou des intervalles. L'utilisation de l'invite de commandes sur Unix permet également de traiter ce type de fichiers.

II – OBJECTIFS DU STAGE

A) Problématique de l'étude

Au sein de ce stage, je me suis intéressée aux évènements d'édition, d'épissage et de maturation des extrémités des transcrits chloroplastiques chez A. thaliana, dans le but de voir s'il existait un lien entre ces différents évènements de maturation, et donc s'il existait un phénomène de co-régulation. L'objectif était donc de recenser les différents évènements de maturation observés pour chaque read, afin de créer une table de contingence par paires d'évènements pour des régions génomiques données, puis réaliser des analyses statistiques pour tester l'existence ou non d'une dépendance significative.

Pour ce faire, nous avons décidé de procéder en deux étapes. La première étape consiste à identifier les reads chevauchant des évènements d'intérêt et les annoter. A ce stade du stage, je me suis uniquement intéressée aux évènements d'édition. La seconde étape consiste à construire les matrices de contingence, en récupérant les reads précédemment annotés. Cette approche en deux temps permettra de calculer plus simplement des matrices de contingence sur plus de deux évènements.

B) Méthodes

B.1. Bibliographie

En début de stage, j'ai effectué une recherche bibliographique sur le génome chloroplastique d'A. thaliana et les évènements de maturation existant, afin de mieux connaître la réalité biologique de notre sujet d'étude et de notre problématique, ainsi que pour faciliter les discussions avec nos collègues biologistes. J'ai également effectué une recherche bibliographique sur quelques outils bio-informatiques, notamment Rsamtools.

B.2. Outils informatiques utilisés

Par la suite, j'ai travaillé sur le logiciel RStudio afin de manipuler le contenu des fichiers BAM et Gff3. Pour cela, j'ai utilisé les packages nécessaires à la manipulation des données génomiques, à savoir Rsamtools, rtracklayer et GenomicRanges, eux-mêmes issus du site Bioconductor qui fournit des outils pour la manipulation de données génomiques sous R.

C) Données

Nous avions en notre possession trois fichiers BAM contenant chacun des milliers d'alignements de long-reads avec la séquence de référence, chaque ligne correspondant à un alignement et donc à un read. Les long-reads, d'environ 1000 paires de bases, sont considérés comme représentant chaque transcrit entier du génome chloroplastique d'A. thaliana.

En plus de ces trois fichiers, nous avions également un fichier de format Gff3 contenant les 43 sites d'édition C vers U connus chez les transcrits du génome chloroplastique d'A. Thaliana (editing_sites), ainsi qu'un fichier du même format contenant les sites connus d'épissage (splice_sites), et enfin un fichier contenant les maturations connues en 5' (different_5_ends). Un dernier fichier contenant les maturations en 3' devrait nous être prochainement fourni par l'équipe de l'IPS2.

A ce stade du stage, je n'ai fait des tests et calculs que sur deux des fichiers précédemment cités, afin d'étudier les évènements d'édition : l'un des trois fichiers BAM (Ct2.bam) et le fichier gff3 qui contenait les positions connues des sites d'édition chez le génome chloroplastique (editing_sites.gff3).

III – RESULTATS

A) Annotation des reads pour l'édition

A.1. Importation des données du fichier editing_sites.gff3 (sites d'édition)

Dans un premier temps, nous avons importé les données du fichier gff3 contenant les sites d'édition, que nous avons stockées dans un objet GRanges nommé *editing_sites*. Nous avons pu voir qu'il contenait 43 sites d'édition C vers U chez le chromosome chloroplastique, et avons nommé chacun de ces sites afin de les identifier de manière unique, en créant une nouvelle colonne de métadonnées. Chaque site d'édition possède une position sur le génome de référence.

A.2. Importation et sélection des reads chevauchant les sites d'édition

Le contenu du fichier BAM étant trop lourd pour être importé entièrement sur R, nous avons importé uniquement les alignements qui nous intéressaient, à savoir ceux dont les reads chevauchaient les sites d'édition connus. Nous avons donc utilisé la fonction readAlignment du package GenomicAlignments pour effectuer un premier tri. Nous avons ainsi importé 71 884 reads, correspondant théoriquement aux transcrits chevauchant un ou plusieurs de nos 43 sites d'édition. Cette première sélection de reads a été stockée dans une variable *all_reads* (Annexe 2). Toutefois, l'alignement généré par readAlignement ne prenant pas en compte le brin (sens ou antisens), certains des 71 884 reads ne s'alignaient pas en réalité sur le bon brin. Un autre tri plus spécifique était donc à faire, un peu plus tard dans le traitement de nos données.

A.3. Annotation des reads

Une fois les données importées et notre première sélection de reads d'intérêt effectuée, nous avons souhaité annoter ces reads en précisant les bases d'édition qu'ils chevauchent, et si ces bases étaient éditées ou non (oui si la base était un « T », et non si c'était un « C »). Pour ce faire, nous avons choisi de représenter ces annotations sous la forme d'une liste de listes (*annotated_reads*) : une première grande liste contenant chaque identifiant de read, puis pour chacun de ces reads, une liste contenant les bases chevauchées (entre 1 et 3 bases au maximum pour notre jeu d'observations) en précisant pour chaque base si elle est éditée ou non (Annexe 2). Cette liste était ensuite remplie à l'aide d'une boucle for, ainsi que deux foncions décrites ci-dessous.

Table d'association par paires

Afin de simplifier le remplissage de notre liste *annotated_reads*, nous avons représenté chaque chevauchement d'un site d'édition par un read sous la forme d'une table d'association de paires read / site d'édition chevauché, que nous avons stockée dans une variable *map_read_to_edition* (Annexe 2). Pour obtenir cette table, nous avons mappé chacun de nos reads sur les sites d'édition qu'ils chevauchaient, grâce à la fonction findOverlaps, issue du package GenomicAlignment.

Fonction is.edited()

Nous avons ensuite créé une fonction *is.edited()* prenant en entrée chacune des paires de notre table d'association, afin d'identifier si la base nucléotidique située au niveau du site d'édition est éditée (« T ») ou non (« C »). Cette fonction utilisait elle-même une autre fonction, *getBasePosition()*, créée pour l'occasion et décrite ci-dessous.

Fonction getBasePosition()

Cette dernière fonction, prenant en entrée le cigar de notre read, nous permettait de lire la base à une position donnée dans le génome, en l'occurrence notre site d'édition, en prenant en compte les informations de délétion, insertion et autres, indiqués dans le cigar. Cela nous permettait ainsi d'atteindre la position de la base correspondant au site d'édition dans la séquence de notre read d'intérêt.

B) Annotation des reads pour l'épissage et la maturation des extrémités

L'annotation des reads sur les évènements d'épissage et de maturation des extrémités se fera sur le même schéma que pour l'annotation des sites édités. En ce qui concerne l'épissage, nous sélectionnerons dans un premier temps les reads présentant une ou plusieurs absences de régions nucléotidiques, lesquelles correspondront aux régions d'épissage répertoriées dans le fichier splice_sites.gff3. Pour cela, nous regarderons les reads commençant avant un site d'épissage connu, et terminant après ce site, et ne possédant pas la région en question, signe qu'elle a été épissée.

Concernant les évènements de maturation aux extrémités des transcrits, nous nous intéresserons dans un premier temps aux maturations en 5'. Nous nous baserons ainsi sur les séquences répertoriées dans le fichier different_5_ends.gff3.

C) Tableau de contingence et tests statistiques

Lorsque tous les reads seront annotés, pour les évènements d'édition, d'épissage et de maturation, nous procèderons à une extraction des reads pour une région génomique donnée, avec une liste d'évènements donnés. Les occurrences des évènements identifiés pourront ensuite être rapportées dans un tableau de contingence. Il faudra par la suite effectuer des tests statistiques sur nos données, afin de déterminer si la dépendance observée entre certains évènements est significative ou non, avec notamment un test de Fisher exact pour un échantillon, et une extension du DESeq2 par la suite pour plusieurs échantillons. Il faudra également veiller à prendre en compte le taux d'erreurs généré par le séquençage Nanopore lors de l'interprétation de nos résultats.

Ces résultats devront ensuite être transmis à l'équipe de biologistes de l'IPS2 afin d'en discuter avec eux et tirer une interprétation biologique de nos résultats.

IV - CONCLUSION

Ce début de stage aura été très enrichissant sur bien des aspects. Tout d'abord il m'aura permis de découvrir le milieu de la recherche en sciences, qui m'était jusqu'alors inconnu. J'ai pu ainsi découvrir le travail effectué en laboratoire en tant qu'enseignant chercheur, ainsi que l'importance de travailler en coopération avec plusieurs corps de disciplines toutes complémentaires entre elles, comme la biologie et la bio-informatique. J'ai également pu mettre en pratique et surtout développé mes compétences en informatique et en bio-informatique dans un contexte professionnel, en me basant sur les connaissances et les compétences acquises durant ma double licence. Malgré les nombreuses difficultés rencontrées lors de la partie informatique sur RStudio, j'ai pu compter sur mes encadrants pour m'aider à trouver des solutions, comprendre mes erreurs, et ainsi progresser.

V – BIBLIOGRAPHIE

- Baudry, K. (2019). *L'éditosome du chloroplaste: questions, éléments de réponses et digressions* (Doctoral dissertation, Université Paris Saclay (COmUE)).
- Grünberger, F., Knüppel, R., Jüttner, M., Fenk, M., Borst, A., Reichelt, R., ... & Grohmann, D. (2020). Exploring prokaryotic transcription, operon structures, rRNA maturation and modifications using Nanopore-based native RNA sequencing. *bioRxiv*, 2019-12.
- Small, I. D., Rackham, O., & Filipovska, A. (2013). Organelle transcriptomes: products of a deconstructed genome. *Current opinion in microbiology*, *16*(5), 652-658.
- Stern, D. B., Goldschmidt-Clermont, M., & Hanson, M. R. (2010). Chloroplast RNA metabolism. *Annual review of plant biology*, *61*, 125-155.

ANNEXE 1 : Exemple de fichier BAM importé sur RStudio

```
[[1]]
[[1]]$qname
[1] "bd45cf15-71fd-4e27-9222-ae100fe65cd1" "4b319853-59c5-4dff-8a05-ce5c7262fa4b"
[3] "688b6111-bd47-4a37-a2c8-0f91783db5bb" "cac1f95b-1f8e-4e94-9caa-c130e7941a58"
[5] "f231f456-a91b-407e-8e10-78698a2980dd"

[[1]]$flag
[1] 16 16 16 16 16

[[1]]$rname
[1] 1 1 1 1 1
Levels: 1 2 3 4 5 Pt Mt

[[1]]$strand
[1] - - - - -
Levels: + - *

[[1]]$pos
[1] 6763 6807 6807 6819 6825

[[1]]$qwidth
[1] 1330 1206 1070 1054 1083

[[1]]$mapq
[1] 60 60 60 60 60

[[1]]$cigar
[1]
"268S15M1D13M2D82M1I86M1I8M1D32M1I67M87N43M3D15M3D3M1D8M151N27M1D3M1D1M3D31M113N1I2M2I62M
```

```
[[1]]$mrnm
[1] <NA> <NA> <NA> <NA> <NA>
Levels: 1 2 3 4 5 Pt Mt

[[1]]$mpos
[1] NA NA NA NA NA

[[1]]$isize
[1] 0 0 0 0 0

[[1]]$seq
DNAStringSet object of length 5:
    width seq
[1]  1330 TTACTCTTGTTCGAGGGGTTCGGTAACCTTACTTGCCTG...TGTTTATACCACTGCTTAGGTTAAACACCCAAGCAGAC
[2]  1206 GACCTTGTTGAGGGTTTTTTTGTGTAACCCTTTCTGTTG...TATACCACTGCTTAGGTTAAACACCCAAGCAGTACGCC
[3]  1070 GGCGTCTGCTCGGTGTTTAACCTAAGCAGTGGTATCAAC...AAACACTCGGCCCAAAAACGAACGAAGGAGATCCATAA
[4]  1054 GGCGTTTGTGTTTGGGTGTTTAACCTAACCAGTGGTATA...GGCAATGCAAGGTTACACAAAACCCTGGACAAGGTCAC
[5]  1083 GGTACTGTCGAGGTAATTTTTTTGTGTAACCTTACTTGC...TGATACCACTGCTTAGGTTAAACACCCAAGCAGACGCC

[[1]]$qual
PhredQuality object of length 5:
    width seq
[1]  1330 %$4*/')),4$'$20.,,&/#%%.0/6645>>556:0'2...,<G112@DFC@I@??@@CEDDEDAIIHB@,46.,-*$$
[2]  1206 +2274676#%-,-,)*,/-)'(&*,5378*;D<*%:</<...(038311.,5853;;:20?CDCA?5/966A?<'=7=?)
[3]  1070 .5,,1&&(.%"3/*AH)&5?<=+@B=AC<?A??BCCAG=...''#&&$$#)-#%&%'-20('21+&0&2$-$//(&'$*)
[4]  1054 $+-$%,&#&($10-96?<729-0)'944276;@=CD@.+...#$"%(%$")63499/6,0')?=0<@?A<@B;+98;97'
[5]  1083 *(,1,'5;+/'''#%%&),21&&,,52><;8@BAA=CC=...&$(.;;-/-((*('1-4$$%3&44.4'0,%(..((&).
```

ANNEXE 2 : Contenu des différentes variables pour l'annotation des reads pour les évènements d'édition

• Contenu de la variable *all_reads* (reads chevauchant les sites d'édition) :

```
```{r}
# Création du paramètre qui nous intéresse, à savoir les sites d'édition
scan_param <- ScanBamParam(which=editing_sites, what=c("seq","qname","cigar","pos"))

# Le paramètre "what" permet de garder les séquences, l'id, le cigar et la position de début du read

#Sélection des reads chevauchant nos sites d'édition, puis affichage des premiers reads
all_reads <- as(GenomicAlignments::readGAlignments(file=bam_file, param=scan_param), "GRanges")

# Affichage des premiers reads sélectionnés
head(all_reads)
length(all_reads)
```
```

```
GRanges object with 6 ranges and 4 metadata columns:
      seqnames     ranges strand |                     seq                qname                 cigar       pos
         <Rle>  <IRanges>  <Rle> |          <DNAStringSet>            <character>           <character> <integer>
  [1]       Pt   4-153623      + | GGCCTTGTCC...AGCAGACGCC b8955688-6f68-4b6b-8.. 322S9M1D12M2D18M1D5M..         4
  [2]       Pt  205-11705      - | GACCTTGTTC...CCAAGCAGAT d8a54c4d-fc94-43f0-b.. 210S7M9477N20M1I11M2..       205
  [3]       Pt  279-29874      - | GTGTTGCTTG...AGCAGACGCC 955e737d-19fa-40b1-8.. 54S6M2D3M2D134M1D54M..       279
  [4]       Pt  279-22277      - | GACCTTGTCC...AGCAGACGCC feb7e360-c50e-4b8a-9.. 120S9M3D333M1I8M2D19..       279
  [5]       Pt 285-134784      - | GGCGTTGCTT...AGCAGATGCT 24383b2e-584d-4e59-a.. 72S61M1D13M1I7M1D31M..       285
  [6]       Pt  286-64448      - | GACCTTGTCC...AGCAGACGCC 9a52ca73-db74-425e-9.. 107S29M1D3M1D5M2D6M1..       286
  -------
  seqinfo: 7 sequences from an unspecified genome
[1] 71884
```

• Initialisation de la liste *annotated_reads* :

```
```{r}

annotated_reads <- list()

# la liste est initialisée avec les identifiants de tous les reads
for (id in mcols(all_reads)$qname){
  annotated_reads[[id]] <- list()
}

head(annotated_reads) ; length(annotated_reads)

```
```

```
$`b8955688-6f68-4b6b-8d3f-0cb8b197ddda`
list()

$`d8a54c4d-fc94-43f0-b4f2-2872e5f0bde0`
list()

$`955e737d-19fa-40b1-8cdd-44326c5338b0`
list()

$`feb7e360-c50e-4b8a-9df5-4cf63a7989cd`
list()

$`24383b2e-584d-4e59-a60f-2024f772f2de`
list()

$`9a52ca73-db74-425e-971f-8b5f0088ae8e`
list()

[1] 35042
```

• Contenu de la variable *map_read_to_editions* (100 premiers reads) :

```
```{r}

# 100 premiers reads
reads_test <- all_reads[1:100]

# Mapping des reads sur chaque site d'édition (table d'association)
map_read_to_edition <- GenomicAlignments::findOverlaps(reads_test, editing_sites)

head(map_read_to_edition)
```
```

```
Hits object with 6 hits and 0 metadata columns:
      queryHits subjectHits
      <integer>   <integer>
  [1]         1           8
  [2]         1          14
  [3]         1          15
  [4]         1          18
  [5]         1          19
  [6]         2           1
  -------
  queryLength: 100 / subjectLength: 43
```

D.3 comaturationTrackeR (2^{nd} version)

Here below is the master's thesis of Benjamin Vacus, which I co-supervised alongside Benoît Castandet and Guillem Rigaill over a six-month period at the Institut des Sciences des Plantes de Paris-Saclay. This thesis describes the second iteration of ComaturationTracker, presented as an R package available temporarily at the following link : https://github.com/SimiliSerpent/comaturationTracker.

Analysis of nanopore data to study co-maturations of the chloroplast transcriptome

Master Thesis

Benjamin Vacus

28/02/22 - 27/07/22

Ecole Polytechnique - Université Paris Saclay M2 AMI2B - IPS2/PMIN/OGE[1]

[1]Master 2 Biologie Computationnelle : Analyse, Modélisation et Ingénierie de l'Information Biologique et Médicale - Institut for Plants Science Paris-Saclay - Plants Micro-organisms Interactions Department - Organellar Gene Expression Team

Contents

1 Introduction

This internship was conducted at the Institute for Plant Science Paris-Saclay (IPS2), located in Gif-sur-Yvette, France. The institute is under the supervision of *Université Paris Saclay – Faculté des Sciences*, CNRS[2], INRAE[3], *Université d'Evry Val d'Essone* and *Université Paris Cité*. As the name suggests, it is specialized in plant biology.

Within the Institute, I was welcomed in the Organellar Gene Expression (OGE) team. The team is directed by Etienne Delannoy and is composed of three other permanent members – Wojciech Majeran, Dario Monachello and my supervisor Benoît Castandet. They focus on the gene expression of organellar genomes in plant cells.

The internship project involved a collaboration with Etienne Delannoy, with biostatistician Guillem Rigaill (*Laboratoire de Mathématiques et Modélisation d'Evry* - LAMME; Genomic Networks team at IPS2 - GNet) and PhD student Arnaud Liehrmann (OGE/GNet/LAMME).

1.1 Chloroplast RNA maturations

Plant cells harbor three different genomes contained in three distinct organelles: the nucleus, the mitochondrion, and the chloroplast. Chloroplasts are the result of a complex evolutionary history. They originate from an endosymbiosis between a heterotrophic eukaryote and a cyanobacterium-like ancestor [1]. Before it became the current chloroplast, the symbiont has progressively been dispossessed of many of its genes that were either lost or transferred to the nuclear genome of the host. This complex history resulted in a complex RNA metabolism in the plastid [2, 3, 4] summed up in Figure 1.

Different transcription start sites

Transcription in the chloroplast is performed by three different RNA polymerases. The first one is homologous to the eubacterial RNA polymerase and its main subunits are encoded in the chloroplast genome. It is called the Plastid Encoded RNA Polymerase (PEP). The other two are homologous to T7-type bacteriophage polymerase, are nuclear encoded and localized in the chloroplast. They are called Nuclear Encoded Plastid RNA Polymerases (NEPs). Many genes can be transcribed by both PEP and NEP, which bind to different promoters, resulting in a set of transcripts with multiple start positions. Looking for an exhaustive list of transcript termini in *A. thaliana*, Castandet & *al.* [5] found 215 transcription start sites for a genome containing less than 130 identified genes.

Trimming

Coupled with a relatively inefficient transcription termination [6], this relaxed transcription often gives birth to polycistronic transcripts with multifarious start and end positions. Before translation occurs, this heterogeneous set of primary transcripts is trimmed and reshaped by a set of specialized enzymes [7]. Polycistrons are cleaved by endoribonucleases, mainly RNase E and RNase J. Transcripts extremities are processed by exoribonucleases: PNPase and RNase II degrade 3' ends, while 5' ends are degraded by RNase J which also possesses an exoribonuclease activity. All those enzymes interact with secondary structures (stem loops) or RNA-Binding Proteins (RBPs) that can stop them in their path and therefore participate in stabilizing specific

[2]Centre National de la Recherche Scientifique
[3]Institut National de Recherche pour l'Agriculture, l'Alimentation et l'Environnement

mature extremities positions. A prominent RBP family is the pentatricopeptide repeats (PPR) proteins that specifically bind to RNA.

Splicing

The plastome entails over 25 introns that need to be removed before translation [8]. Most organellar introns are group II introns and nuclear-encoded proteins play a major role in their splicing [9].

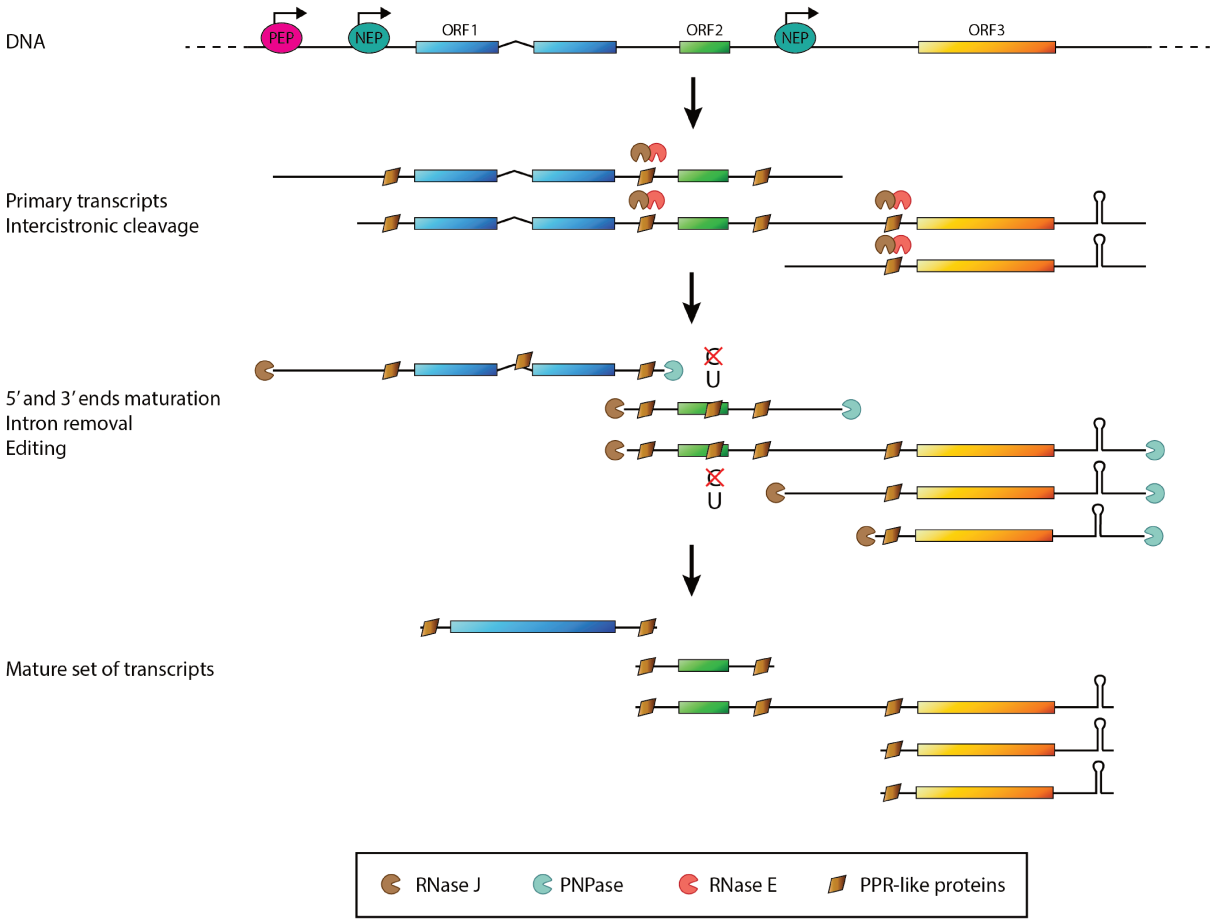


Figure 1: **Chloroplast RNAs maturation** - Different kind of polymerases (Plastid Encoded Polymerase - PEP or Nuclear Encoded Polymerases - NEPs) transcribe a DNA region into polycistronic RNAs. Some polycistrons are endonucleolytically cleaved into smaller fragments by RNase J and RNase E guided by Pentatricopeptide Repeat (PPR) proteins that specifically bind to the RNAs. Extremities are processed by RNase J (5' ends) and PNPase (3' ends) that can be blocked in their path by PPR proteins or stem-loops. Introns are spliced and some cytosines are edited into uraciles.

RNA editing

Before they get translated, many RNAs undergo another maturation called *editing* which consists in the modification of one nucleotide on a specific position into another one. In *A. thaliana*, it is restrained to C-to-U modifications. Over 40 editing sites have been numbered throughout the plastome [10]. As for splicing, PPR proteins also play a major role in editing sites recognition and cytidine deamination [11].

1.2 Studying co-maturations of the transcripts

A majority of the actors involved in those maturation steps are located in the nucleoids, compact structures containing several plastome copies as well as proteins and RNAs [12]. This spatial proximity suggests possible interactions between them. Even though the chloroplast community has gone a long way in understanding those maturations and in identifying the molecular actors, we often lack an accurate comprehension of how things articulate around each other, and we certainly ignore most of the interplays happening during RNA metabolism.

Starting with this idea that effectors of different kinds of RNA maturations interact together, it has been conjectured that some enzymes need other enzymes (or other enzymes' work) to start acting. As pointed out by the team in Guilcher & *al.* [8], this was verified multiple times. In 2001, Schmitz Linneweber & *al.* [13] evidenced a splicing-dependent editing event on spinach *ndhA* gene fragments in a tobacco plastid. The other way around, in 2015, Yap & *al.* [14] showed that the splicing of the *atpF* transcript was diminished in absence of the editing of that very same transcript, thus exhibiting an editing-dependent splicing event in the chloroplast of *A. thaliana*. Eventually, Malbert & *al.* [15] reported an editing default on a mitochondrial site in absence of an enzyme responsible for the editing of a neighboring site, proving the existence of editing-dependent editing events in the organelles.

From now on, we will refer to such dependencies using the term **co-maturations** introduced by the team in Guilcher & *al.* The previously mentioned studies prove that co-maturations are a thing. However, these studies targeted a very limited number of sites and might represent anecdotal exceptions only. Today, Next Generation Sequencing (NGS) techniques grant access to a more global view on all the RNA maturations in plant organelles. Namely, the growing use of Illumina sequencing devices fostered the systematic recognition of maturation sites at the transcriptome level [5]. Yet using those devices do not allow to jointly monitor multiple distant maturation events on the same read. This technique indeed produces reads that are at most 350 nucleotides long when chloroplast transcripts can be substantially longer, and maturation events can be distant from as much as several thousand bases. To address this issue, the team took advantage of the Oxford Nanopore Technology (ONT) that is able to produce long reads [16]. This sequencing method was developed on polyadenylated RNAs. However, in chloroplasts, polyadenylation of the RNAs acts as a degradation signal. The team succeeded in adapting the protocol for non-polyadenylated RNAs, even though they were unable to capture reads longer than about 2500 nucleotides.

Based on a list of 14 introns (after removing tRNAs because of their size) and on the list of 43 editing sites identified in Ruwe & *al.* [10], they surveyed the maturation states on every site across all reads. Then, they built a contingency table for each pair of maturation events indicating the number of reads in each maturation state for the two sites: it contained the number of reads maturated on both / the first / the second / none of the sites. Eventually, they ran a Fisher's exact test on those tables to test for independence between the two maturation events. P-values were adjusted with Benjamini-Hochberg correction [17]. Out of the 138 pairs of maturation sites that were simultaneously covered, they found a total of 42 co-maturations.

1.3 Limitations of the existing approach

This approach successfully yielded a first plastome-wide map of the dependencies between pairs of maturation events. Despite constituting a breakthrough in the field of chloroplast RNA maturations, it deserves to be examined and improved. The most questionable part cer-

tainly is the statistical determination of true co-maturations: after using the Fisher's exact test and applying the correction, they called co-maturation a pair of events with a corrected p-value < 0.1 on at least two replicates, and $< 5 \times 10^{-3}$ on the merge of the three replicates. This procedure bears several unsatisfying aspects:

- The way the different replicates are assembled is debatable to say the least. Even though arbitrary thresholds are commonplace in the field of statistics, the way the different replicates are combined leads to a conservative testing (requesting a corrected p-value < 0.1 on 2 out of 3 independent experiments is demanding), and no effort is made to consider their specificities (mean, internal variance).
- The method does not consider the specifics of the RNA-seq count data. It is common knowledge today that those data are characterized by non-Poisson and singular dependence of the variance on the mean. This does not correspond to the hypergeometric distribution hypotheses of the Fisher's exact test used in this approach.

Yet, modeling RNA-seq count data is an issue that has already been tackled in the recent literature, as it quickly became necessary to handle data generated by modern NGS tools, mainly in the study of differential expression of genes [18, 19]. Some widely used tools, like *edgeR* [20, 21] or *DESeq* [22] use a Negative-Binomial distribution (sometimes also called Gamma-Poisson distribution) to model the RNA-seq count data. Under this assumption, the variance V and the mean μ are related as follow: $V = \mu + \alpha\mu^2$ where α is the *dispersion* parameter. This dependence on the mean better fits to the RNA-seq data than in the classical Poisson distribution (often used to model discrete counts) where $V = \mu$. The estimation of the dispersion parameter differs according to the method, and the mean is estimated with a Generalized Linear Model (GLM) having a logarithmic link function.

In the standard case of differential expression analysis, raw counts are first filtered to remove genes with close-to-zero expression. This filtering step yields huge impact on the outcome of the analysis [23]. Then comes a no less important normalization step whose goal is to remedy the bias inherent in each bio-technical experiment: different coverage for each replicate, differences in the nature of the genes (GC content, length) or environmental variability (growth conditions). The usefulness of this normalization step is extensively accepted in the RNA-seq community; nonetheless, the modalities found in the literature are diverse and no consensus has emerged so far [24]. Differential expression is eventually tested, *e.g.* using a Wald test in the case of *DESeq*, and an FDR control is next performed (with Benjamini-Hochberg correction).

The goal of my M2 internship was to overcome the shortcomings of the strategy previously used by the team, by proposing a new statistical method based on contrast to track co-maturations of chloroplast RNAs. I take advantage of the *DESeq2* R package [25, 26] in an ingenious way to accurately model the count data, and I compare the results with those obtained with the Fisher former method. I present here a complete analysis pipeline that takes the aligned reads *.bam* files along with the annotations of known maturation sites as input and outputs a list of co-maturations. I also deliver the brand-new associated R package *comaturationTracker* [27].

2 Results

2.1 A new method based on statistical contrast

This section first details the theoretical methodology used to find RNAs co-maturations and how we model the co-maturation data with Deseq2, before moving on to the practical pipeline.

2.1.1 Using contrast

Notations

For a given pair of maturation events, we write A (resp. B) when the first event site (resp. the second) is matured, and we use $\overline{A}$ (resp. $\overline{B}$) when it is not. With these notations, we further denote by μ_{AB} the observed number of reads covering both sites *and* with both sites matured. Likewise, we write $\mu_{A\overline{B}}$ (resp. $\mu_{\overline{A}B}$, resp. $\mu_{\overline{AB}}$) the number of reads covering both sites where only the first one is matured (resp. only the second site, resp. none of the sites). When this number of reads corresponds to the situation in the i^{th} replicate, we sometimes write μ_{ABi}.

In our new method, read distribution is modelled by a Negative-Binomiale distribution. As explained in introduction, this distribution is characterized by its mean μ and variance $V = \mu + \alpha\mu^2$ where α is the dispersion parameter. The mean is estimated using a GLM with a logarithmic link function. In this model, we use a baseline μ_0. We add a parameter A when the first site is matured and B when the second site is matured. Because we are looking for a dependency between the two maturation events, we add a suitable interaction term AB when both sites are matured. Eventually, we can add a replicate term Ri when modeling counts in the i^{th} replicate. The model can be written as follows:

$$\begin{cases} log(\mu_{ABi}) = \mu_0 + A + B + AB \\ log(\mu_{A\overline{B}i}) = \mu_0 + A \\ log(\mu_{\overline{A}Bi}) = \mu_0 + B \\ log(\mu_{\overline{AB}i}) = \mu_0 \end{cases}$$

Statistical contrast

The question of the dependency between two maturation events can be asked as follows: does the maturation of one site impact the maturation of the other site? In other words, does the maturation state at one site influence the maturation rate at the other site? This can be answered by testing whether the proportion of reads matured at one site differs depending on the maturation state of the other site. Mathematically, this is the same as testing the equality:

$$\begin{aligned} \frac{\mu_{AB}}{\mu_{\overline{A}B}} = \frac{\mu_{A\overline{B}}}{\mu_{\overline{AB}}} &\Leftrightarrow log(\mu_{AB}) - log(\mu_{\overline{A}B}) = log(\mu_{A\overline{B}}) - log(\mu_{\overline{AB}}) \\ &\Leftrightarrow \big(log(\mu_{AB}) - log(\mu_{\overline{A}B})\big) - \big(log(\mu_{A\overline{B}}) - log(\mu_{\overline{AB}})\big) = 0 \end{aligned}$$

We write $C = \big(log(\mu_{AB}) - log(\mu_{\overline{A}B})\big) - \big(log(\mu_{A\overline{B}}) - log(\mu_{\overline{AB}})\big)$ and we call *statistical contrast* this last quantity. Testing whether the maturations at both sites are dependent or not is the same as testing whether the contrast significantly differs from 0 or not. When we replace the

logarithms with their estimations in our model, we eventually find $C = AB$. Note that this remains true when adding the replicate effects Ri (see APPENDIX A.1).

Estimating dispersion

The dispersion is estimated by the R package *DESeq2* [26] that is designed for classical gene expression analysis. It is estimated under the assumption that counts with similar average have similar dispersion. Thus, a smooth curve is fitted on empirical pair-wise dispersion that is then shrinked towards the curve.

2.1.2 R pipeline

My statistical analysis pipeline has been encapsulated in the R *comaturationTracker* package. Therefore, I describe the different steps of the pipeline by going through the corresponding functions of the package. Both the method described in Guilcher & *al.* and this new approach are summarized in FIGURE 2 along with the R functions.

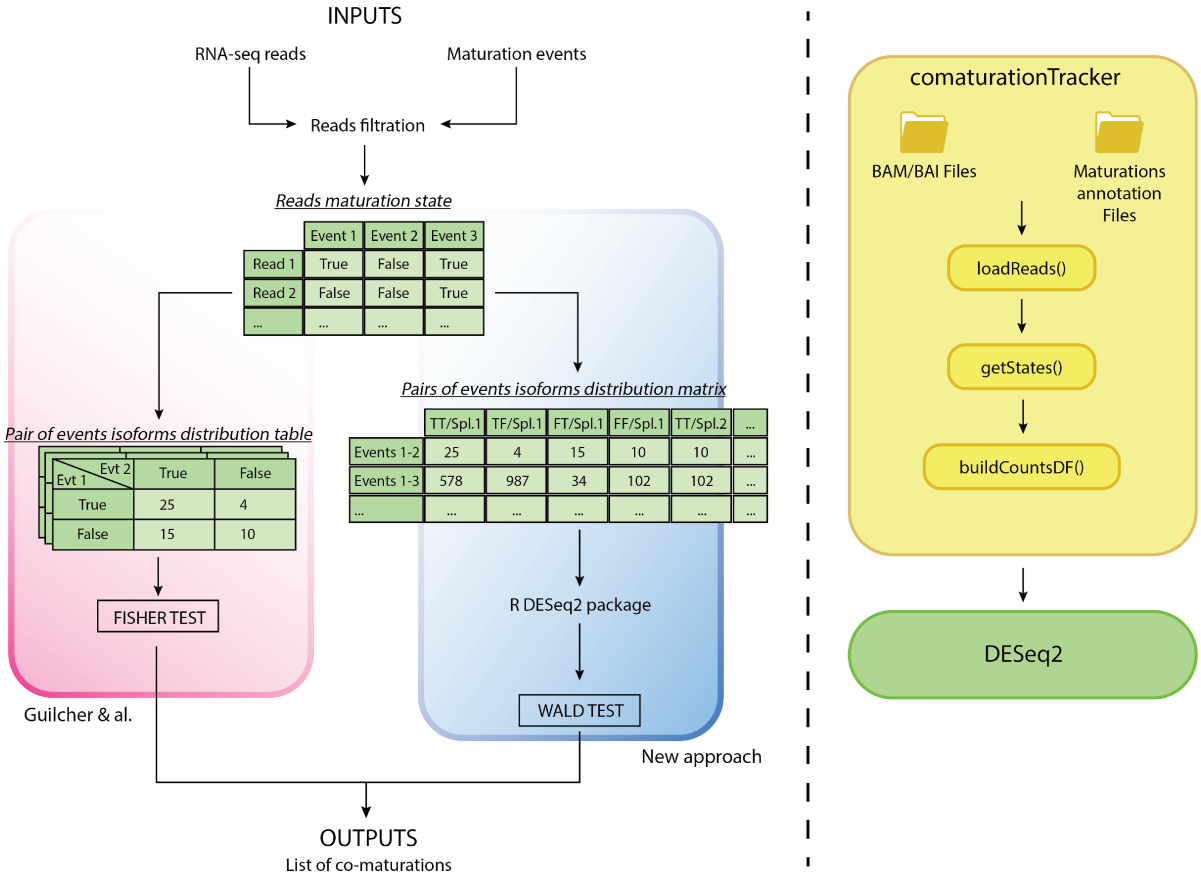


Figure 2: **Different methods to track RNA co-maturations** - *Left panel*: In the Guilcher & *al.* method, input reads are filtered and their maturation state at each maturation site are surveyed. Contingency tables are built for each pair of events, containing the number of reads matured (True) or not (False) at each site. Fisher's exact test are run on the tables. In our new method, counts of reads in each maturation states are stored in a matrix with one line for one pair of events. This table is used as input for a DESeq2 analysis with an appropriate design. *Right panel*: To each step of our new method corresponds a R function available in the *comaturationTracker* R package [27].

`loadReads()`

Aligned reads are first filtered as follows. Reads mapping outside of the genome range are removed. The same is done for reads mapping none of the specified maturation sites, as well as for reads with abnormal lengths or for reads mapping different locations on the genome. This can be done in R using the *comaturationTracker* `loadReads()` function, with reads and maturation sites of interest loaded from respectively user specified *.bam* and *.gff3* files.

`getStates()`

For each read, the maturation state of any maturation site it covers is assessed. This is straightforward for known editing sites where a C indicates an absence of maturation whereas a U reflects a matured position. It is not as easy for splicing because introns are never completely (or not at all) spliced. In this case we retained the heuristic used in Guilcher & *al.*: if less than 10% of the intron is found, it is considered spliced, *i.e.* matured. These maturation states are stored in a matrix with one row for one read and one column for one maturation event site. This assession step is accomplished by the *comaturationTracker* `getStates()` function.

`buildCountsDF()`

The GLM and statistical testing part is handled by the R *DESeq2* package. *DESeq2* functions take as input a matrix with one row for each gene and one column for each experiment, since it was built for a different purpose. To make it able to handle our data, I build a matrix where one row corresponds to one pair of events, and one column describes a combination of replicate number and maturation state at each site (see FIGURE 2). I also build a matrix equivalent to the *DESeq2* "conditions" table that links the experiments with the parameters of the GLM. Both matrices are computed by the *comaturationTracker* `buildCountsDF()` function.

I use the *DESeq2* package to estimate the dispersion and the parameters of our GLM. Eventually, I use it again to test whether the contrast term AB statistically differs from 0 or not. This is done internally using a Wald test followed by a Benjamini-Hochberg adjustment of p-values. Pairs of events with an adjusted p-value below an arbitrary 5×10^{-3} threshold are considered dependent.

2.2 Application

In this section, I present the results obtained with this new method on a few application cases. I start with comparing it with the Guilcher & *al.* approach before moving on to more complex cases.

2.2.1 One condition, pairs of events

The pipeline was run on the same set of reads as in Guilcher & *al.*. My package took 24 minutes 44 seconds to complete the analysis - see APPENDIX B.2 for details. Out of the $6,2 \times 10^6$ reads across the three biological replicates, 386885 reads have passed the filters. Analysis results are shown in FIGURE 3.

From a set of 43 editing sites and 25 introns, one can consider 2278 pairs of maturation event sites. However, for 2137 of them, no read covers both sites simultaneously. This issue is similar to the one encountered in classical gene expression analysis, where many genes are lowly or not

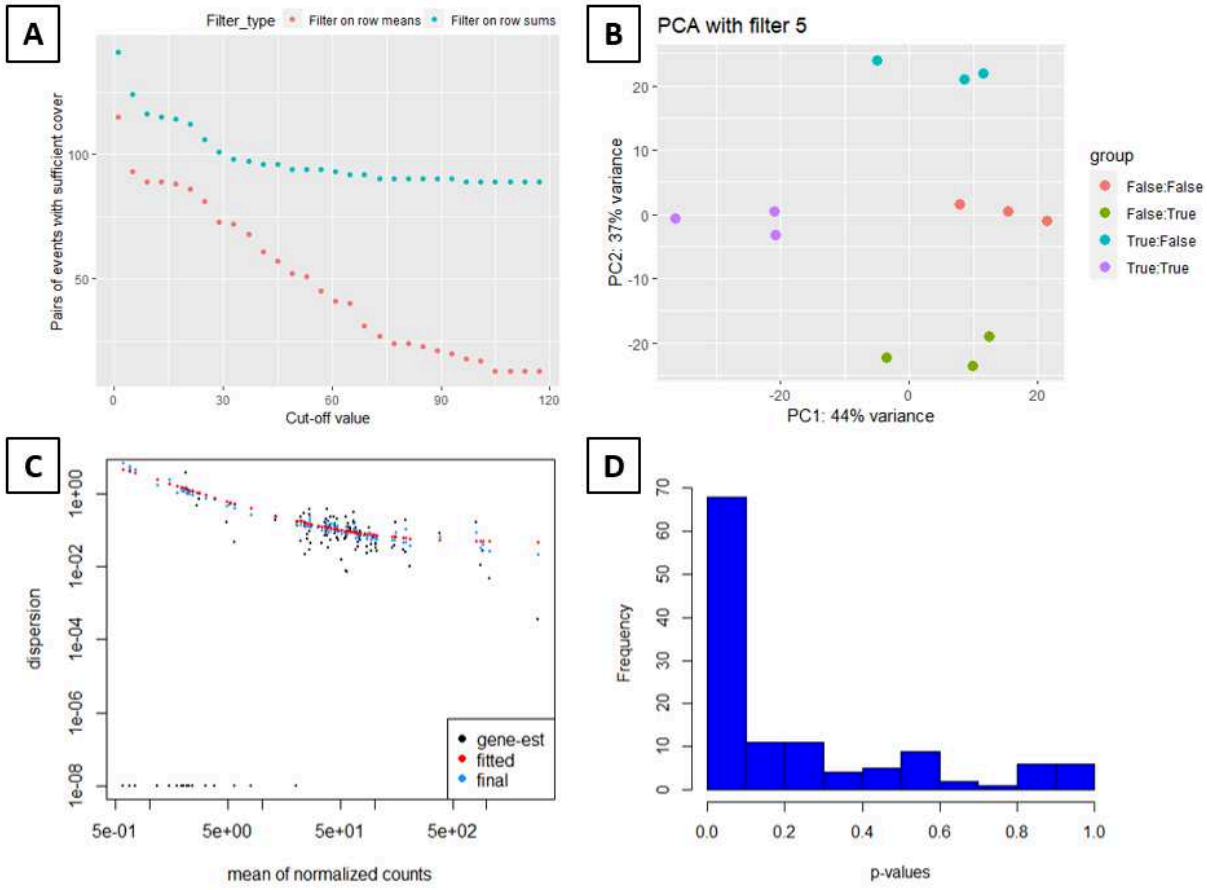


Figure 3: **Analysis of co-maturations in col-0 *A. thaliana*** - **(A)** Number of pairs of events passing through a cut-off on reads coverage. Cut-off can be applied on the total sum of reads covering both sites across all replicates (blue dots), or on the mean number of reads across replicates and maturation states (red dots). **(B)** Principal Component Analysis (PCA) on the counts matrix after filtration with a size 5 cut-off on total number of reads covering each pair. The twelve points accounts for the three replicates and four maturation isoforms and are coloured according to maturation state at each site. **(C)** Dispersion plot as output by *DESeq2*. Pair-wise empirical dispersions are represented by black dots. Estimated dispersion (blue dots) is a shrinkage of empirical dispersion towards a fit (red dots). **(D)** Histogram of p-values when testing the interaction term (or contrast term) AB.

at all expressed. To filter "low expression pairs", I used representations such as in FIGURE 3-**A** where I show the number of pairs left after filtering on the total sum (blue dots) or the average (red dots) of reads simultaneously covering both sites of the pair (note that the average is the mean across all replicates and all possible isoforms). I chose a naive size 5 cut-off on the total number or reads covering both sites of the pair. This resulted in 123 sufficiently covered pairs.

PCA and dispersion plot are displayed in FIGURE 3-**B,C**. As explained in 2.1, we tested the dependence with a Wald test on the contrast term AB estimated in the GLM. Raw p-values distribution is shown in FIGURE 3-**D**. After *DESeq2*-integrated Benjamini-Hochberg FDR control, I applied a 5×10^{-3} arbitrary threshold on adjusted p-values. The independence hypothesis $H0$ was rejected for 43 pairs of events. The raw list of co-maturation found with *comaturationTracker* is available in APPENDIX B.3.

A brief comparison with the outcome of the Guilcher & *al.* method is shown in FIGURE 4. 40 pairs of maturation events are found to be co-maturations using both pipelines. For 3 other pairs, $H0$ is rejected with our method but not using the former one. Conversely, for 2 pairs of events, $H0$ is rejected in Guilcher & *al.* but not with our new approach. FIGURE 4-**B** displays

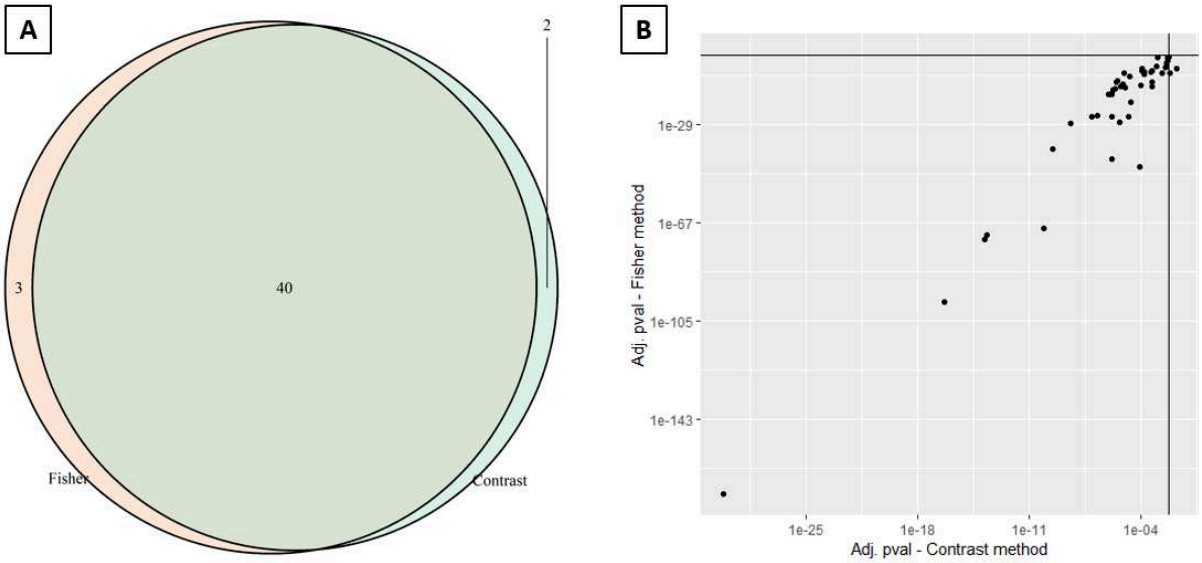


Figure 4: **Comparison of the two methods** - **(A)** Venn diagram comparing co-maturations found with both approaches. **(B)** Comparison between Guilcher & *al.* (Y axis) and our new method (X axis) of the adjusted p-values for all pairs of events where $H0$ is rejected in at least one method. Black lines materialize the 5×10^{-3} thresholds.

the adjusted p-value for all of those 45 pairs. It reveals that ranking of co-maturations is similar regardless of the method, with discriminant pairs lying close to the arbitrary thresholds.

2.2.2 Second condition, pairs of events

Plant culture and Nanopore sequencing was not only conducted on *A. thaliana* col-0 ecotypes but also on mutants for the PNPase enzyme, now referred to as PNP-mutant. PNPase is an exoribonuclease responsible for 3'-ends degradation in the chloroplasts (see 1). Naturally, the team wondered what impact the aforementionned mutation could have on co-maturations. Again I ran the pipeline on those data to obtain the results shown in FIGURE 5.

As before, the filtering power of two kinds of cut-off (on the total number of counts or on the mean of the counts across the different isoforms/replicates/conditions) are presented in panel **(A)**. With the same naive size 5 cut-off, I selected a set of 136 sufficiently covered pairs of events. PCA, dispersion plot and raw p-values distribution are displayed in panel **(A)**, **(B)** and **(C)**, respectively.

A Venn diagram between the set of co-maturations found in the PNP-mutant and the set found in the *WT* is shown in panel **(E)**. 43 co-maturations are found in both conditions, while 12 pairs of maturation events are found to be dependent only in the PNP-mutant, and 1 only in the *WT*. Adjusted p-values for these 56 pairs are displayed for comparison in panel **(F)**. Details of the co-maturations found in both ecotypes are available in APPENDIX B.3.

2.2.3 Two conditions, pairs of events

The previous comparison between the different biological conditions does not take into account the variability between the *WT* and the PNP-mutant experiments. An asset of our new method is that it can be easily modified to study the evolution of co-maturations without

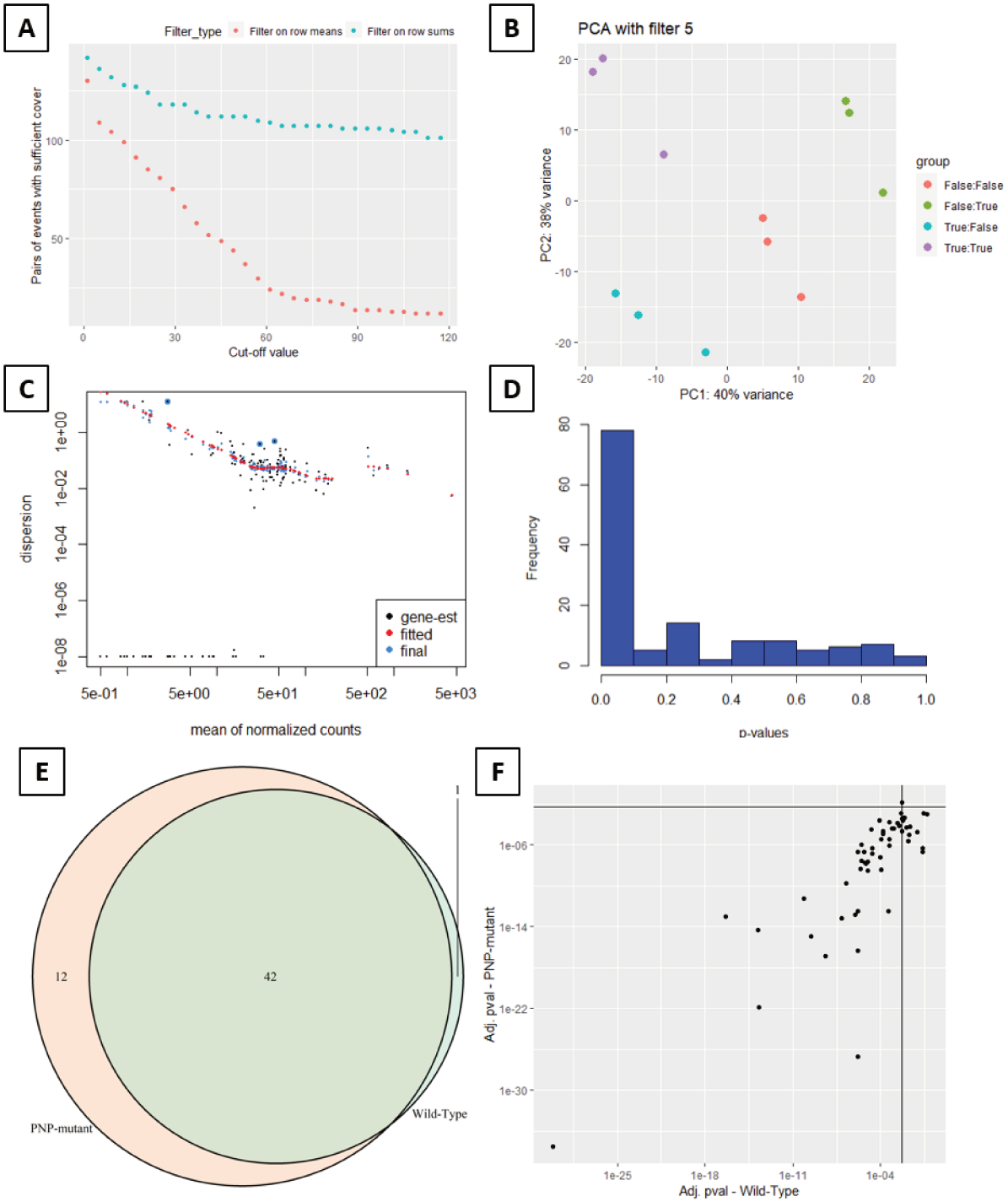


Figure 5: **Analysis of co-maturations in PNP-mutant *A. thaliana*** - **(A)** Number of pairs of events passing through a cut-off on reads coverage (cf *supra*). **(B)** PCA on the counts matrix after filtration with a size 5 cut-off on total number of reads covering each pair. The twelve points accounts for the three replicates and four maturation isoforms and are coloured according to maturation state at each site. **(C)** Dispersion plot as output by *DESeq2*. Pair-wise empirical dispersions are represented by black dots. Estimated dispersion (blue dots) is a shrinkage of empirical dispersion towards a fit (red dots). **(D)** Histogram of p-values when testing the interaction term (or contrast term) AB. **(E)** Venn diagram comparing co-maturations found in both conditions. **(F)** Comparison between WT (Y axis) and PNP-mutant (X axis) of the adjusted p-values for all pairs of events where $H0$ is rejected in at least one condition. Black lines materialize the 5×10^{-3} thresholds.

neglecting this aspect. To adress this question, we denote by μ_{*X} the means of counts in the *wild-type* (WT) condition, and by μ_{*Y} the means of counts in the *PNP-mutant* condition. We take this second biological condition into account in our GLM by adding a *mutant* parameter Y. Because we are interested in finding interactions between the condition (WT or *mutant*) and the co-maturations, we also add new interaction terms in the model. We obtain the following, where the first four lines model counts in the mutant condition:

$$\begin{cases} log(\mu_{ABYi}) = \mu_0 + A + B + AB + Y + AY + BY + ABY \\ log(\mu_{A\overline{B}Yi}) = \mu_0 + A + Y + AY \\ log(\mu_{\overline{A}BYi}) = \mu_0 + B + Y + BY \\ log(\mu_{\overline{AB}Yi}) = \mu_0 + Y \\ log(\mu_{ABXi}) = \mu_0 + A + B + AB \\ log(\mu_{A\overline{B}Xi}) = \mu_0 + A \\ log(\mu_{\overline{A}BXi}) = \mu_0 + B \\ log(\mu_{\overline{AB}Xi}) = \mu_0 \end{cases}$$

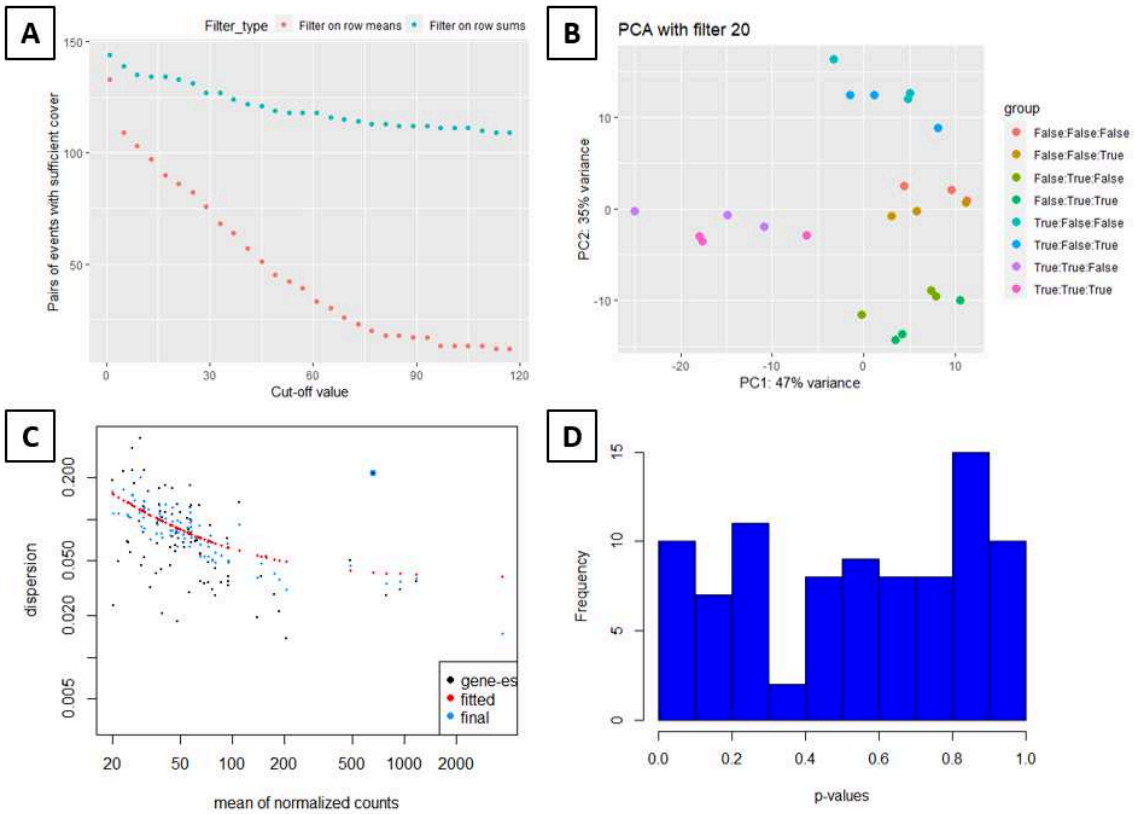


Figure 6: **Analysis of the impact of mutation (*pnp* knock-out) on co-maturations - (A)** Number of pairs of events passing through a cut-off on reads coverage (cf *supra*). **(B)** PCA on the counts matrix after filtration with a size 5 cut-off on total number of reads covering each pair. The 24 points accounts for the three replicates and four maturation isoforms across both conditions. They are coloured as follows: matured on 1^{st} site / matured on 2^{nd} site / mutant. **(C)** Dispersion plot as output by *DESeq2*. **(D)** Histogram of p-values when testing on the contrast term ABY.

The above-mentioned question translates as follows: how does the dependency between the two events differ from one condition to another? The dependence in one condition can be tested as we did with the previous model. Thus, the question mathematically comes down to knowing

if:

$$\frac{\mu_{ABY}}{\mu_{\overline{A}BY}} / \frac{\mu_{A\overline{B}Y}}{\mu_{\overline{A}\overline{B}Y}} = \frac{\mu_{ABX}}{\mu_{\overline{A}BX}} / \frac{\mu_{A\overline{B}X}}{\mu_{\overline{A}\overline{B}X}}$$

This implies:

$$\begin{aligned}\big(log(\mu_{ABY}) - log(\mu_{\overline{A}BY})\big) - \big(log(\mu_{A\overline{B}Y}) - log(\mu_{\overline{A}\overline{B}Y})\big) \\ = \big(log(\mu_{ABX}) - log(\mu_{\overline{A}BX})\big) - \big(log(\mu_{A\overline{B}X}) - log(\mu_{\overline{A}\overline{B}X})\big)\end{aligned}$$

Which, by replacing with the model estimators, and after simplification, gives:

$$ABY = 0$$

We thus adapted the method shown before and re-used the R *DESeq2* package with a new input and a new design. As in the previous section, I present the analysis in FIGURE 6. Like above, panel **(A)** shows the filtering power of two kinds of cut-off. Out of the 2278 imaginable pairs, 144 were covered by at least one read. I chose a less naive size 20 cut-off on the means. This left only 88 pairs that were sufficiently covered to be retained in the analysis. Impact of the mutation was tested as before, using a Wald test on the ABY term. Distribution of raw p-values is shown in panel **(D)**. No pairs showed significant dependency evolution between WT and PNP-mutant after FDR control.

2.2.4 One condition, trios of events

Since many maturation events sometimes take place on a unique transcript, it is likely that groups of more than two events show co-maturations. Our method can be further extended to look for such dependencies between more than two events. In this section we modify the model to examine trios of events.

Just like in the previous section, we start with the basic model (one condition, pairs of events). We add an event labeled C along with a new corresponding parameter C in the model. We also try adding interaction terms AC, BC and ABC. The new model can be written like so:

$$\begin{cases} log(\mu_{ABCi}) = \mu_0 + A + B + C + AB + BC + AC + ABC \\ log(\mu_{A\overline{B}Ci}) = \mu_0 + A + C + AC \\ log(\mu_{\overline{A}BCi}) = \mu_0 + B + C + BC \\ log(\mu_{\overline{A}\overline{B}Ci}) = \mu_0 + C \\ log(\mu_{AB\overline{C}i}) = \mu_0 + A + B + AB \\ log(\mu_{A\overline{B}\overline{C}i}) = \mu_0 + A \\ log(\mu_{\overline{A}B\overline{C}i}) = \mu_0 + B \\ log(\mu_{\overline{A}\overline{B}\overline{C}i}) = \mu_0 \end{cases}$$

Again we look at the following quantity:

$$\begin{aligned}\Big(\big(log(\mu_{ABC}) - log(\mu_{\overline{A}BC}) - \big(log(\mu_{A\overline{B}C}) - log(\mu_{\overline{A}\overline{B}C})\big)\Big) \\ - \Big(\big(log(\mu_{AB\overline{C}}) - log(\mu_{\overline{A}B\overline{C}}) - \big(log(\mu_{A\overline{B}\overline{C}}) - log(\mu_{\overline{A}\overline{B}\overline{C}})\big)\Big)\end{aligned}$$

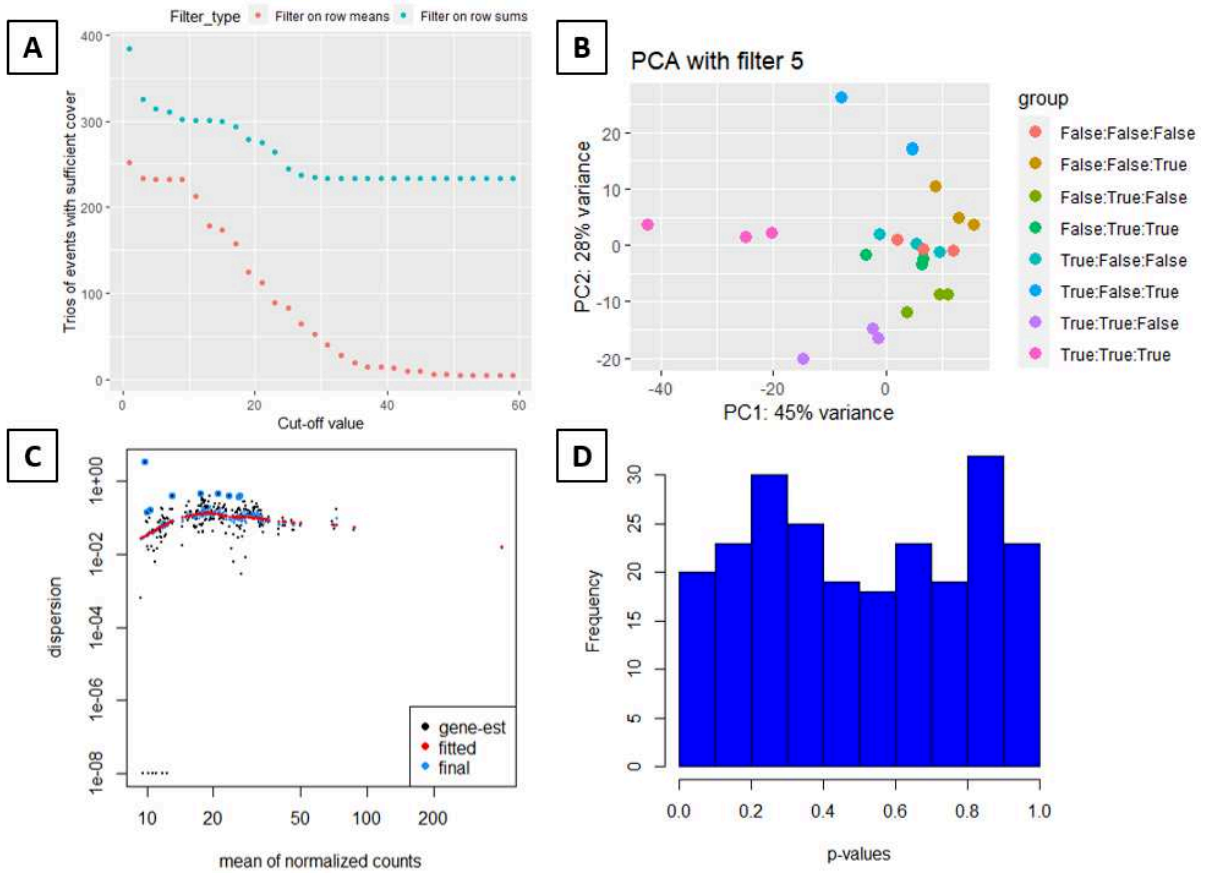


Figure 7: **Analysis of co-maturations in trios of events** - **(A)** Number of trios of events passing through a cut-off on reads coverage. Cut-off can be applied on the total sum of reads covering all three sites across all replicates, or on the mean number of reads across replicates and maturation states. **(B)** PCA on the counts matrix after filtration with a size 5 cut-off on the average number of reads covering each pair (averaged across isoforms and replicates). The 24 points accounts for the three replicates and eight maturation isoforms. They are coloured according to the following: matured on first site / second site / third site. **(C)** Dispersion plot as output by *DESeq2*. **(D)** Histogram of p-values when testing on the interaction term (or contrast term) ABC.

This expression results in just ABC after simplification. Therefore, the question becomes: is the estimated parameter ABC significantly different from 0? Again, I used the DESeq2 approach on the wild-type data. Results are shown in FIGURE 7. No trios showed significant dependency after FDR control.

2.3 Analysis of reads extremities processing state

Besides editing and splicing, the processing of the RNAs extremities represent a decisive maturation step that has not been considered so far. Starting with the raw reads, one way to visualize how they are processed is to look at the distribution of their extremities along the genome. This is represented in FIGURE 8-**A**.

Because we were seeking for dependencies with other maturation events, I also analyzed the distribution of the extremities of reads mapping editing or splicing sites. We divided the profile in two plots based on their maturation state for this site. This allows for visual comparison of the processing profiles. An example is given in FIGURE 8-**B**.

Eventually, FIGURE 8-**C** displays the comparison between matured-vs-primary profiles in the wild-type case and in the PNP-mutant case.

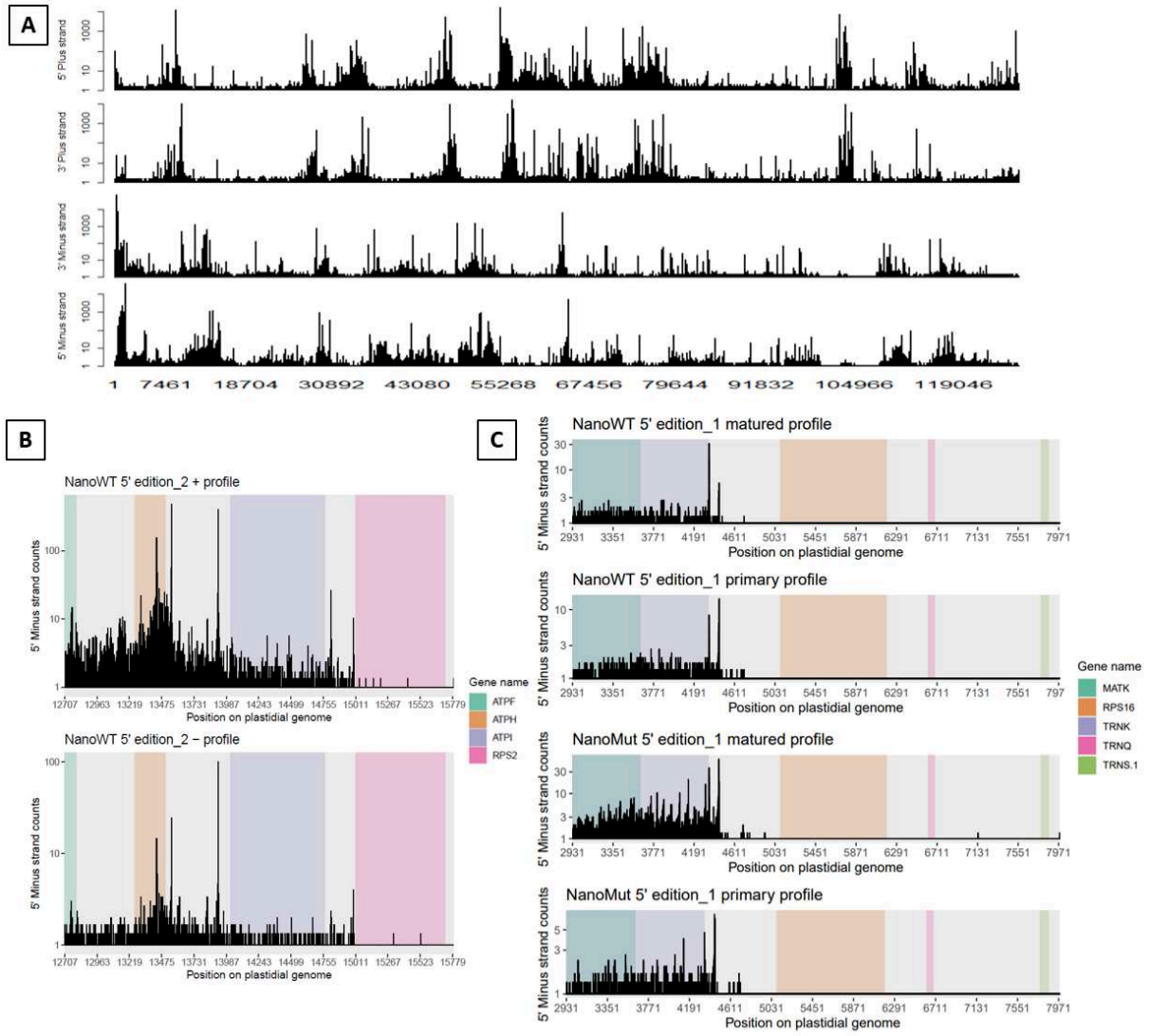


Figure 8: **Overview of the distribution of reads extremities** - Reads extremities positions have been surveyed and their distribution is displayed using bar-plots. **(A)** Distribution of extremities of all reads along the genome - inverted-repeated regions excluded. Extremities counts are averaged over the three replicates and projected on a logarithmic scale. Represented distributions, from top to bottom: plus strand 5' ends ; plus strand 3' ends ; minus strand 3' ends ; minus strand 5' ends. **(B)** This bar-plot is zoomed in on a specific region of the minus strand. An editing site is located on the leftmost position (position 12707). Only the 5' extremities of reads mapping this site are shown. The top graphic displays the distribution of the positions of 5' ends of reads where the site is matured (edited), whereas the bottom graphic shows the distribution of 5' extremities of reads where the site has not been edited. Genes ranges are represented with background colors. **(C)** Same as in (B) but with a different editing site located at position 2931. Besides the two matured-vs-primary-site top graphics, the same profiles for the PNP-mutant are displayed below.

3 Methods

3.1 RNA-seq data

The *WT* RNAseq data I used during this internship are the same data as those used in Guilcher & *al.* [8]. RNAs were extracted from the leaves of 5-weeks old col-0 *A. thaliana*. Please refer to the *Material and methods* section of the aforementioned paper if you want to reproduce the data. Protocols used for Nanopore sequencing can be found online at `https://forgemia.inra.fr/guillem.rigaill/nanopore_chloro`. The output reads are publicly available from the NCBI SRA database under the accession number PRJNA748959.

The PNP-mutant RNAseq data were produced in the exact same way, with the i^{th} replicate of each condition being cultivated in the same batch. However, the data have not been published yet.

3.2 Statistical pipeline and package development

Finding co-maturations with R

All the work was conducted using the R statistical programming language (version 4.0.3 - 2022-10-10) under the windows RStudio IDE (version 2022.02.3) [25, 28]. I used *Gitlab* as a version control tool.

All packages used are publicly available from the CRAN or from the Bioconductor project. Maturation sites annotations were imported with the `rtracklayer` package [29]. RNA-seq reads were imported using the `readGAlignments()` function from the `GenomicAlignments` package [30]. Reads filtration was done with `IRanges` and `Rsamtools` [31]. GLM creating and statistical testing were performed with the `DESeq2` package [26].

The *comaturationTracker* package

The `comaturationTracker` package was developed based on the principles exposed in the online *R Packages (2e)* book, available at `https://r-pkgs.org/`. The package itself is available at `https://github.com/SimiliSerpent/comaturationTracker`. To install the package, see Appendix B.1.

To make the package, I mainly used the `devtools` package [32], but I also used `usethis`, `knitr`, `testthat` and `roxygen2` [33, 34, 35, 36].

4 Discussion

4.1 A new competitive statistical method

We have developed a new method to assess the co-maturations of the chloroplast RNAs. It was meant to overcome some of the weaknesses of the Fisher method described in introduction [8]. The fundamental remains: in both case we wonder whether the maturation state at one site influences the maturation rate at the other site. We answer this question by testing whether the proportion of reads matured at one site differs depending on the maturation state at the other site - and this is precisely what the Fisher's exact test does on the contingency tables. In comparison our method shows several strengths.

By using the *DESeq2* R package, we ensure a good modeling of the RNA-seq count data using a tried-and-true negative-binomial distribution. Using a GLM we are able to efficiently incorporate the different replicates with a *replicate effect* parameter Ri that takes into account their specificities.

Like for the use of this replicate effect, the choice not to use the *DESeq2* normalization of the count data was not trivial. As pointed out by Dillies *et al.* [24], the DESeq normalization method stands out, and we did not replace it with any other kind of normalization. Nevertheless, we suffer from several biases which may for example be related to library size differences. Of course this latter bias is corrected with the *replicate* parameter, but using this parameter in the model prevents us from modeling batch effects, as is usually done in RNA-seq data analysis. The decision not to normalize the data and to add a replicate effect was taken after trying all kind of modalities on the data (results of the tests can be found in Appendix B.4). It comes out that using replicate effects without any normalization is the only approach that gives results coherent with those of the Fisher method. These results are discussed below.

Our method is also flexible and can be adapted to meet other constraints. For example, another complication that I have neglected so far is the high error rate (or low precision) of the Nanopore Technology devices. Even though this drawback is progressively corrected on the latest versions of the Nanopore sequencers, it remains relatively imprecise compared to its Illumina counterpart. The error rate on our data had previously been estimated to be about 5%. This means any nucleotide in any sequence has 5% chance to be misread as another nucleotide. Our method allows to account for this error rate by modifying the counts in the input matrix - under the strong hypothesis that the error rate is constant along every read (see details in Appendix A.2).

4.2 Promising results

The results obtained on the *one condition* case are derived from simple design choices. As mentioned above, a short investigation has motivated us to remove all kind of normalization and to add replicate effects in the model. Given the curves shown in Figure 3.A, I used a small filter to arbitrarily remove any pair of maturation sites that were almost never simultaneously covered. This naive filter was retained because it yielded satisfying results, but it can probably be optimized. Because the number of reads covering a pair of events is limited overall (< 100 for most pairs), power to detect any dependence is likely to be very small. Yet the results or our new method on the WT data are deemed good because of several signals:

- the different maturation isoforms are well clustered on the output of the Principal Component Analysis (see Figure 3.B)

- the dispersion estimates are well approached by the *DESeq2* fit and the raw p-values histogram is close to a uniform distribution between 0 and 1 suggesting the Negative-Binomial is a relevant choice to model the data (see FIGURE 3.C&D)
- the obtained co-maturations are close to what was found with the Fisher method, illustrating the coherence between the two methods (see FIGURE 4.A)

The list of co-maturations obtained here resembles the list found with the Fisher method but with changes nonetheless. Before trying to analyze those differences, it is important to notice that pairs of events considered to be a co-maturation in one case and not in the other *always* lie close to the rejection threshold in the latter case. This can be seen in the top-right corner of FIGURE 4.B. Another striking element in this figure is that the points are approximately aligned along the $y = x$ axis. This means the ranking of the co-maturations is well preserved between the two methods.

That being said, it is virtually impossible to verify whether those gray cases should be held as co-maturations or not. It would require many other replicates (and maybe also a higher transcripts coverage) and given the cost of such experiments and the arbitrariness of the rejection thresholds, the game is definitely not worth the gamble. However, it is plausible that the different enhancements brought with the new method explain those changes, and I therefore advise working with the final co-maturations list obtained.

Dense maturation sites clusters drive the analysis

It is critical to report here the influence of dense maturation sites clusters. During our exploration of the data, I noticed that a small number of maturation sites that are close-enough to be simultaneously covered by a few transcripts generate an exponential number of pairs of events. In our case, the 15 sites of the *ndhb* gene region account for 22% of the sites. However, almost every possible pair between those events is sufficiently covered to pass the cut-off. In the end, they account for 92 pairs of events in the analysis - 75% of the 123 analyzed pairs. Consequently, this region yields a huge impact on the *DESeq2* estimates as well as on the shape of the raw p-values histogram and, thus, on our appreciation of the method. To circumvent this issue, I suggest weighting the pairs in the p-values histogram in order to give a lower strength to highly represented sites.

4.3 The PNP-mutant

The co-maturations found in the PNP-mutant ecotype substantially differ from those found in the *WT*. However, this comparison ignores the differences in library size and all other possible bias and environmental factors between the two sets of experiments. For that reason, I do not discuss these differences any further.

The *two conditions* model study limitations

Conversely, the *two conditions* model takes those biases into accounts by adding the Y parameter to the second set of experiments. Hence the results are expected to be different.

However, the first time I ran the analysis, we were quite dissatisfied with the results, especially with the p-values histogram that was far from being uniform between 0 and 1 and whose density was clustered towards 1 (data not shown). This indicates that our data are poorly modeled by the chosen Binomial-Negative distribution. Using the new model, it is also possible to test

the impact of the condition on the maturation of site A (or B). To do so, I test the contrast $2AY + ABY$ - or $2BY + ABY$ (see Appendix A.3). The profiles of the p-values histograms generated for sites A and sites B were very different. This came as a surprise because A is nothing more than *the first site in the pair* and thus, we expect a symmetrical behavior between the two profiles.

We improved the results by rethinking the filtration step. Indeed, having two biological conditions means we have approximately twice more reads and thus the naive size 5 cut-off ends up being less selective than before. We chose to filter the data with a size 20 cut-off on the mean number of reads across conditions, replicates and maturation states (red dots on Figure 6.A). This led to the results presented in 2.2.

The dispersion plot in Figure 6.C shows no oddity. Looking at the result of the Principal Component Analysis in Figure 6.B, it seems that the points are all slightly shifted in the same direction between the two conditions, but they cannot be clearly separated. This suggests there is no big difference between the PNP-mutant and the *WT*, which is confirmed by the p-values histogram (Figure 6.D). It is relatively close to a uniform distirbution between 0 and 1, but lacks the close-to-zero peak indicating significant differences in the counts. Logically enough, no co-maturation is found after FDR control.

These results are no real surprise because we are only looking at *editing* and *splicing* events, but the mutated gene in the PNP-mutant encodes a ribonuclease and therefore we would expect mainly changes at the extremities of the reads. However, I went from testing a dual interaction - AB - to testing a triple interaction - ABY. Thus, it might require more statistical power to perceive small differences (more replicates, more coverage) which is beyond our reach given the current cost of those experiments. Unfortunately, I did not have data from a mutant with an editing or splicing defect on hand.

This may also be the reason why we observe inconclusive results in the *trios of events* case. Dispersion plot (Figure 7.C) and PCA (Figure 7.C) are more messy, and no co-maturation of 3 events is found after p-values adjusting. This goes against the results found by Guilcher & *al.* [8] who exhibit dependencies between no less than 9 events in the *ndhb* genomic region.

4.4 Future perspective

Analyzing the *read ends processing* event

What would be very interesting (given the available data from a PNP-mutant ecotype) is the addition of the *processing of read ends* event in our list of maturations.

We can see in Figure 8 that the distribution of extremities is noisy - however, some peaks can easily be distinguished. Those peaks correspond to preferred end positions. On Figure 8.B (with the distribution of extremities for all reads mapping the 12707 editing site), we can see that reads admit several precise preferred terminations. Even more interesting is the profile difference between the top graphic (reads having site 12707 edited) and the bottom graphic (where the site has not been edited). Some peaks can only be found in the former one and hence this is an example of flagrant dependency between an editing event and the processing of reads extremities: it could be that transcripts processed to a given length are *always* and *quickly* edited.

Another example is given in Figure 8.C with the editing site localized at position 2931, where

we can see again in the top two graphs an inversion of the preferred terminations. Below those two profiles (*edited state versus primary state* in the *WT*), distribution of the extremities in the PNP-mutant are displayed for comparison, and the previously seen inversion is missing. It suggests that co-maturations are not the same in the *WT* and in a PNP-mutant with a read ends processing defect.

But this third kind of maturation event is not as easy to model as the other two. Where it is straightforward to assess the editing state of a read, and relatively easy to decide whether an intron is spliced or not, the distribution of reads extremities is noisy and we do not possess a pre-computed list of preferential termination positions. To integrate the processing of extremities in the analysis, for the sake of generalizability, we do not want to build a method based on a list of known preferential terminations. I propose to test, for every editing/splicing event and any position, the ratio between the number of reads (covering the event) ending on this position and the number of reads (covering the event) ending elsewhere, before we compare the *matured* and the *primary* states. Due to a lack of time, I have not implemented this test yet.

Pending developments

In its current shape, our method (along with the *comaturationTracker* R package) only takes as input a set of reads and a list of maturation event sites. Thus, it can easily be transposed to other data to test for co-maturations in different species or biological environment (like in the mitochondrion for example).

However, there is still a lot of space for improvement and further development. No biological investigations have been conducted on the obtained co-maturations so far, and many questions are yet to be considered. What are the molecular mechanisms at work behind the newly obtained co-maturations? What does it tell us about the role of transcripts maturation in the cell? The statistical study of the processing of extremities still needs to be realized and tested. In Guilcher & *al.*, the authors proposed an ordering of the maturation events based on the counts of reads maturated at both sites: can we do the same in the PNP-mutant? Would we find the same ordering? How does the taking into account of the error rate modify this ordering?

On the package side, and beside adding the study of the extremities, some work is still required. It lacks a control of the functions inputs for example, as well as some unit tests to facilitate future developments. Before it becomes a handier tool, one might want to upgrade the outputs, *e.g.* with graphics to display the several clusters of interdependent maturation events. Maybe it would be better to encapsulate the *DESeq2* part of the analysis in homemade functions to make it even more plug-and-play. It would also be great to add some extremities visualization tools since those are quite unprecedented observations. No doubt the package is going to evolve in the foreseeable future.

5 Career Assessment

From a professional point of view, I deem this internship a success because it accurately met my expectations with, nevertheless a fair amount of challenge and difficulties.

Technically speaking, it was the first opportunity for me to extensively use the R language. Despite being familiar with the software already - thanks to its use at school, I found out that I was still very inexperienced with it. It was a long way before figuring out the spirit of R programming and I am only beginning to see my limitations when it comes to producing time and memory efficient R code. Although there is plenty of room for optimization, I was proud to eventually assemble my own package.

This internship allowed me to further familiarize with the statistical analysis of transcriptomics data. I had time to dive into the brass tracks of Generalized Linear Models and exponential families. I discovered the statistical contrast which is quite a basic tool, but which is not limited to bio-informatics and that sometimes proves to be surprisingly powerful. Using the *DESeq* method, I have also refined my knowledge of read counts modeling. Even though I spent most of my time behind a computer screen, I was largely immersed in the world of biology, and especially in the context of chloroplasts in plant cells.

Methodologically speaking, I once again was given the chance to explore a scientific field and to perform an ongoing bibliographical work. I was well directed and helped in the pursue of my research and I discovered the circle *making assumptions*, *testing*, *interpreting*, *investigating the snags to understand better*. It was hard yet instructive to spend weeks in the fog of misunderstanding when faced with confusing results. I found out the researcher way that the outcome is almost never as expected and that answering one question raises many others.

As mentioned above, I had a great time working with three accomplished researcher and a PhD student. I had to learn to translate scientific questions in each of their respective languages, from the most statistical-ish to the most biological-ish. It was very stimulating to be part of an office with researchers, research engineers, post-doctoral fellows, PhD students and other interns. I also had the unique opportunity to attend the JOBIM[4] conference in Rennes and to present my work with a poster[5].

When I applied for this internship I was looking forward to work with Nanopore Technology data. This is exactly what it was about and I believe most of what I learned during those five months will be useful when I start working for the Defense. It seems the world of bio-informatics is a tiny world: before I join the DGA[6] in 2023, I applied for an internship at CEA Evry that was connected to the IPS2 - and my future team leader found out about my work at the JOBIM conference.

[4] *Journées Ouvertes pour la Biologie, l'Informatique et les Mathématiques*
[5] Poster is available at `https://github.com/SimiliSerpent/JOBIM-2022/blob/main/poster_vacus.pdf`
[6] *Direction Générale de l'Armement*

References

[1] Beverley R Green. "Chloroplast genomes of photosynthetic eukaryotes". In: *The plant journal* 66.1 (2011), pp. 34–44.

[2] David B Stern, Michel Goldschmidt-Clermont, and Maureen R Hanson. "Chloroplast RNA metabolism". In: *Annual review of plant biology* 61 (2010), pp. 125–155.

[3] Uwe G Maier et al. "Complex chloroplast RNA metabolism: just debugging the genetic programme?" In: *BMC biology* 6.1 (2008), pp. 1–9.

[4] Alice Barkan. "Expression of plastid genes: organelle-specific elaborations on a prokaryotic scaffold". In: *Plant physiology* 155.4 (2011), pp. 1520–1532.

[5] Benoıt Castandet et al. "Systematic sequencing of chloroplast transcript termini from Arabidopsis thaliana reveals¿ 200 transcription initiation sites and the extensive imprints of RNA-binding proteins and secondary structures". In: *Nucleic acids research* 47.22 (2019), pp. 11889–11905.

[6] David B Stern and Wilhelm Gruissem. "Control of plastid gene expression: 3 inverted repeats act as mRNA processing and stabilizing elements, but do not terminate transcription". In: *Cell* 51.6 (1987), pp. 1145–1157.

[7] Arnaud Germain et al. "RNA processing and decay in plastids". In: *Wiley Interdisciplinary Reviews: RNA* 4.3 (2013), pp. 295–316.

[8] Marine Guilcher et al. "Full length transcriptome highlights the coordination of plastid transcript processing". In: *International journal of molecular sciences* 22.20 (2021), p. 11297.

[9] Andeol Falcon De Longevialle, Ian D Small, and Claire Lurin. "Nuclearly encoded splicing factors implicated in RNA splicing in higher plant organelles". In: *Molecular Plant* 3.4 (2010), pp. 691–705.

[10] Hannes Ruwe et al. "Arabidopsis chloroplast quantitative editotype". In: *FEBS letters* 587.9 (2013), pp. 1429–1433.

[11] Mizuho Ichinose et al. "Two DYW subclass PPR proteins are involved in RNA editing of ccmFc and atp9 transcripts in the moss Physcomitrella patens: first complete set of PPR editing factors in plant mitochondria". In: *Plant and Cell Physiology* 54.11 (2013), pp. 1907–1916.

[12] Wojciech Majeran et al. "Nucleoid-enriched proteomes in developing plastids and chloroplasts from maize leaves: a new conceptual framework for nucleoid functions". In: *Plant physiology* 158.1 (2012), pp. 156–189.

[13] Christian Schmitz-Linneweber et al. "Heterologous, splicing-dependent RNA editing in chloroplasts: allotetraploidy provides trans-factors". In: *The EMBO Journal* 20.17 (2001), pp. 4874–4883.

[14] Aaron Yap et al. "AEF 1/MPR 25 is implicated in RNA editing of plastid atpF and mitochondrial nad5, and also promotes atpF splicing in Arabidopsis and rice". In: *The Plant Journal* 81.5 (2015), pp. 661–669.

[15] Bastien Malbert et al. "The analysis of the editing defects in the dyw2 mutant provides new clues for the prediction of RNA targets of Arabidopsis E+-class PPR proteins". In: *Plants* 9.2 (2020), p. 280.

[16] Yunhao Wang et al. "Nanopore sequencing technology, bioinformatics and applications". In: *Nature biotechnology* 39.11 (2021), pp. 1348–1365.

[17] Yoav Benjamini and Yosef Hochberg. "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 (1995), pp. 289–300.

[18] Koen Van den Berge et al. "RNA sequencing data: Hitchhiker's guide to expression analysis". In: *Annual Review of Biomedical Data Science* 2.1 (2019), pp. 139–173.

[19] Rory Stark, Marta Grzelak, and James Hadfield. "RNA sequencing: the teenage years". In: *Nature Reviews Genetics* 20.11 (2019), pp. 631–656.

[20] Mark D Robinson and Gordon K Smyth. "Moderated statistical tests for assessing differences in tag abundance". In: *Bioinformatics* 23.21 (2007), pp. 2881–2887.

[21] Davis J McCarthy, Yunshun Chen, and Gordon K Smyth. "Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation". In: *Nucleic acids research* 40.10 (2012), pp. 4288–4297.

[22] Simon Anders and Wolfgang Huber. "Differential expression analysis for sequence count data". In: *Nature Precedings* (2010), pp. 1–1.

[23] Richard Bourgon, Robert Gentleman, and Wolfgang Huber. "Independent filtering increases detection power for high-throughput experiments". In: *Proceedings of the National Academy of Sciences* 107.21 (2010), pp. 9546–9551.

[24] Marie-Agnès Dillies et al. "A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis". In: *Briefings in bioinformatics* 14.6 (2013), pp. 671–683.

[25] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2021. URL: `https://www.R-project.org/`.

[26] Michael I Love, Wolfgang Huber, and Simon Anders. "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2". In: *Genome biology* 15.12 (2014), pp. 1–21.

[27] Benjamin Vacus. *comaturationTracker: Using Contrast to Study RNA Transcripts CoMaturations*. 2022. URL: `https://github.com/SimiliSerpent/comaturationTracker`.

[28] RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, PBC. Boston, MA, 2022. URL: `http://www.rstudio.com/`.

[29] Michael Lawrence, Robert Gentleman, and Vincent Carey. "rtracklayer: an R package for interfacing with genome browsers". In: *Bioinformatics* 25 (2009), pp. 1841–1842. DOI: `10.1093/bioinformatics/btp328`. URL: `http://bioinformatics.oxfordjournals.org/content/25/14/1841.abstract`.

[30] Michael Lawrence et al. "Software for Computing and Annotating Genomic Ranges". In: *PLoS Computational Biology* 9 (8 2013). DOI: `10.1371/journal.pcbi.1003118`. URL: `http://www.ploscompbiol.org/article/info%3Adoi%2F10.1371%2Fjournal.pcbi.1003118`.

[31] Martin Morgan et al. *Rsamtools: Binary alignment (BAM), FASTA, variant call (BCF), and tabix file import*. R package version 2.6.0. 2020. URL: `https://bioconductor.org/packages/Rsamtools`.

[32] Hadley Wickham et al. *devtools: Tools to Make Developing R Packages Easier*. R package version 2.4.3. 2021. URL: `https://CRAN.R-project.org/package=devtools`.

[33] Hadley Wickham, Jennifer Bryan, and Malcolm Barrett. *usethis: Automate Package and Project Setup*. R package version 2.1.6. 2022. URL: `https://CRAN.R-project.org/package=usethis`.

[34] Yihui Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.39. 2022. URL: https://yihui.org/knitr/.

[35] Hadley Wickham. "testthat: Get Started with Testing". In: *The R Journal* 3 (2011), pp. 5–10. URL: https://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf.

[36] Hadley Wickham et al. *roxygen2: In-Line Documentation for R*. R package version 7.2.0. 2022. URL: https://CRAN.R-project.org/package=roxygen2.

Appendices

A Methodological comments

A.1 The contrast is independent from the replicate influence

If we write the whole model with three replicates, we have:

$$\begin{cases} log(\mu_{AB1}) = \mu_0 + A + B + AB + R1 \\ log(\mu_{A\overline{B}1}) = \mu_0 + A + R1 \\ log(\mu_{\overline{A}B1}) = \mu_0 + B + R1 \\ log(\mu_{\overline{AB}1}) = \mu_0 + R1 \\ log(\mu_{AB2}) = \mu_0 + A + B + AB + R2 \\ log(\mu_{A\overline{B}2}) = \mu_0 + A + R2 \\ log(\mu_{\overline{A}B2}) = \mu_0 + B + R2 \\ log(\mu_{\overline{AB}2}) = \mu_0 + R2 \\ log(\mu_{AB3}) = \mu_0 + A + B + AB + R3 \\ log(\mu_{A\overline{B}3}) = \mu_0 + A + R3 \\ log(\mu_{\overline{A}B3}) = \mu_0 + B + R3 \\ log(\mu_{\overline{AB}3}) = \mu_0 + R3 \end{cases}$$

By taking the mean of contrast on every replicate, we get:

$$\begin{aligned} C &= \frac{1}{3}\sum_{i=1}^{3}\big(log(\mu_{ABi}) - log(\mu_{\overline{A}Bi})\big) - \big(log(\mu_{A\overline{B}i}) - log(\mu_{\overline{AB}i})\big) \\ &= \frac{1}{3}\sum_{i=1}^{3}\big(\mu_0 + A + B + AB + Ri - (\mu_0 + B + Ri)\big) - \big(\mu_0 + A + Ri - (\mu_0 + Ri)\big) \\ &= \frac{1}{3}\sum_{i=1}^{3}(A + AB) - (A) \\ &= \frac{1}{3}\sum_{i=1}^{3} AB \\ &= AB \end{aligned}$$

This demonstrates that the contrast does not change when adding different replicate effects into the model.

A.2 Contrast modified with respect to sequencing error rate

We propose to take the error rate of the Nanopore sequencer into account in our model under strong hypotheses, namely:

- the operation of sequencing one base of any RNA transcript suffers from an error rate τ which is independent from the base and the location on the genome.
- this error holds for the maturation event "edition of a base", but obviously not for the event "splicing of an intron" because many bases are involved in the latter - an error rate of 0 is supposed for this second type of maturation.

As described in introduction, during the transcript maturation, a nucleotide C is sometimes edited and turned into a U. The observed data is the result of the sequencing step in which a nucleotide N is read as N with probability $(1-\tau)$ and is misread as another nucleotide with probability τ. Thus, one C would for instance be misread as a U with probability $\frac{\tau}{3}$ since there are three other possible nucleotides. Using the total probability formula, we can write:

$$\begin{aligned}\mathbb{P}(\tilde{U}) &= \mathbb{P}(\tilde{U}|U) \times \mathbb{P}(U) + \mathbb{P}(\tilde{U}|\bar{U}) \times \mathbb{P}(\bar{U}) \\ &= (1-\tau) \times \mathbb{P}(U) + \tau \times \mathbb{P}(\bar{U})\end{aligned} \tag{1}$$

where $\tilde{U}$ is the event "a U is read", U is the event "the base really is a U" and $\bar{U}$ the event "the base actually is not a U (it can be A, G or C)".

From now on, the notations introduced in 2.1 designate the "true" *in vivo* events, with observed events being denoted with a $\sim$: $\tilde{\mu}_{AB}$, $\tilde{\mu}_{A\overline{B}}$, $\tilde{\mu}_{\overline{A}B}$ and $\tilde{\mu}_{\overline{AB}}$. Then with numerous counts, (1) allows to write the observed counts in terms of the "true" ones:

$$\begin{cases} \tilde{\mu}_{AB} = (1-\tau)^2\mu_{AB} + \frac{\tau}{3}(1-\tau)(\mu_{A\overline{B}} + \mu_{\overline{A}B}) + \frac{\tau^2}{9}\mu_{\overline{AB}} \\ \tilde{\mu}_{A\overline{B}} = (1-\tau)^2\mu_{A\overline{B}} + \frac{\tau}{3}(1-\tau)(\mu_{AB} + \mu_{\overline{AB}}) + \frac{\tau^2}{9}\mu_{\overline{A}B} \\ \tilde{\mu}_{\overline{A}B} = (1-\tau)^2\mu_{\overline{A}B} + \frac{\tau}{3}(1-\tau)(\mu_{AB} + \mu_{\overline{AB}}) + \frac{\tau^2}{9}\mu_{A\overline{B}} \\ \tilde{\mu}_{\overline{AB}} = (1-\tau)^2\mu_{\overline{AB}} + \frac{\tau}{3}(1-\tau)(\mu_{A\overline{B}} + \mu_{\overline{A}B}) + \frac{\tau^2}{9}\mu_{AB} \end{cases}$$

Which one can rewrite:

$$\begin{pmatrix} \tilde{\mu}_{AB} \\ \tilde{\mu}_{A\overline{B}} \\ \tilde{\mu}_{\overline{A}B} \\ \tilde{\mu}_{\overline{AB}} \end{pmatrix} = \begin{pmatrix} (1-\tau)^2 & \frac{\tau}{3}(1-\tau) & \frac{\tau}{3}(1-\tau) & \frac{\tau^2}{9} \\ \frac{\tau}{3}(1-\tau) & (1-\tau)^2 & \frac{\tau^2}{9} & \frac{\tau}{3}(1-\tau) \\ \frac{\tau}{3}(1-\tau) & \frac{\tau^2}{9} & (1-\tau)^2 & \frac{\tau}{3}(1-\tau) \\ \frac{\tau^2}{9} & \frac{\tau}{3}(1-\tau) & \frac{\tau}{3}(1-\tau) & (1-\tau)^2 \end{pmatrix} \begin{pmatrix} \mu_{AB} \\ \mu_{A\overline{B}} \\ \mu_{\overline{A}B} \\ \mu_{\overline{AB}} \end{pmatrix}$$

Or again:

$$\begin{pmatrix} \tilde{\mu}_{AB} \\ \tilde{\mu}_{A\overline{B}} \\ \tilde{\mu}_{\overline{A}B} \\ \tilde{\mu}_{\overline{AB}} \end{pmatrix} = \frac{\tau}{3}(1-\tau) \begin{pmatrix} \frac{3}{\tau}(1-\tau) & 1 & 1 & (\frac{3}{\tau}(1-\tau))^{-1} \\ 1 & \frac{3}{\tau}(1-\tau) & (\frac{3}{\tau}(1-\tau))^{-1} & 1 \\ 1 & (\frac{3}{\tau}(1-\tau))^{-1} & \frac{3}{\tau}(1-\tau) & 1 \\ (\frac{3}{\tau}(1-\tau))^{-1} & 1 & 1 & \frac{3}{\tau}(1-\tau) \end{pmatrix} \begin{pmatrix} \mu_{AB} \\ \mu_{A\overline{B}} \\ \mu_{\overline{A}B} \\ \mu_{\overline{AB}} \end{pmatrix}$$

The matrix in the middle is of the following form:

$$\begin{pmatrix} a & 1 & 1 & \frac{1}{a} \\ 1 & a & \frac{1}{a} & 1 \\ 1 & \frac{1}{a} & a & 1 \\ \frac{1}{a} & 1 & 1 & a \end{pmatrix} \text{ with } a = \frac{3}{\tau}(1-\tau), \text{ of inverse matrix: } \frac{a}{(a^2-1)^2} \begin{pmatrix} a^2 & -a & -a & 1 \\ -a & a^2 & 1 & -a \\ -a & 1 & a^2 & -a \\ 1 & -a & -a & a^2 \end{pmatrix}$$

Thus, after matrix inversion:

$$\begin{pmatrix} \mu_{AB} \\ \mu_{A\overline{B}} \\ \mu_{\overline{A}B} \\ \mu_{\overline{AB}} \end{pmatrix} = (\frac{\tau}{3}(1-\tau))^{-1} \frac{\frac{3}{\tau}(1-\tau)}{((\frac{3}{\tau}(1-\tau))^2-1)^2} \begin{pmatrix} (\frac{3}{\tau}(1-\tau))^2 & -\frac{3}{\tau}(1-\tau) & -\frac{3}{\tau}(1-\tau) & 1 \\ -\frac{3}{\tau}(1-\tau) & (\frac{3}{\tau}(1-\tau))^2 & 1 & -3\frac{3}{\tau}(1-\tau) \\ -\frac{3}{\tau}(1-\tau) & 1 & (\frac{3}{\tau}(1-\tau))^2 & -\frac{3}{\tau}(1-\tau) \\ 1 & -\frac{3}{\tau}(1-\tau) & -\frac{3}{\tau}(1-\tau) & (\frac{3}{\tau}(1-\tau))^2 \end{pmatrix} \begin{pmatrix} \tilde{\mu}_{AB} \\ \tilde{\mu}_{A\overline{B}} \\ \tilde{\mu}_{\overline{A}B} \\ \tilde{\mu}_{\overline{AB}} \end{pmatrix}$$

Or:

$$\begin{pmatrix} \mu_{AB} \\ \mu_{A\overline{B}} \\ \mu_{\overline{A}B} \\ \mu_{\overline{AB}} \end{pmatrix} = \frac{9}{(8\tau^2-18\tau+9)^2} \begin{pmatrix} 9(1-\tau)^2 & -3\tau(1-\tau) & -3\tau(1-\tau) & \tau^2 \\ -3\tau(1-\tau) & 9(1-\tau)^2 & \tau^2 & -3\tau(1-\tau) \\ -3\tau(1-\tau) & \tau^2 & 9(1-\tau)^2 & -3\tau(1-\tau) \\ \tau^2 & -3\tau(1-\tau) & -3\tau(1-\tau) & 9(1-\tau)^2 \end{pmatrix} \begin{pmatrix} \tilde{\mu}_{AB} \\ \tilde{\mu}_{A\overline{B}} \\ \tilde{\mu}_{\overline{A}B} \\ \tilde{\mu}_{\overline{AB}} \end{pmatrix}$$

This leads to the reformulation of the initial system:

$$\begin{cases} \mu_{AB} = \frac{9}{(8\tau^2-18\tau+9)^2}\left[9(1-\tau)^2\tilde{\mu}_{AB} - 3\tau(1-\tau)(\tilde{\mu}_{A\overline{B}} + \tilde{\mu}_{\overline{A}B}) + \tau^2\tilde{\mu}_{\overline{AB}}\right] \\ \mu_{A\overline{B}} = \frac{9}{(8\tau^2-18\tau+9)^2}\left[9(1-\tau)^2\tilde{\mu}_{A\overline{B}} - 3\tau(1-\tau)(\tilde{\mu}_{AB} + \tilde{\mu}_{\overline{AB}}) + \tau^2\tilde{\mu}_{\overline{A}B}\right] \\ \mu_{\overline{A}B} = \frac{9}{(8\tau^2-18\tau+9)^2}\left[9(1-\tau)^2\tilde{\mu}_{\overline{A}B} - 3\tau(1-\tau)(\tilde{\mu}_{AB} + \tilde{\mu}_{\overline{AB}}) + \tau^2\tilde{\mu}_{A\overline{B}}\right] \\ \mu_{\overline{AB}} = \frac{9}{(8\tau^2-18\tau+9)^2}\left[9(1-\tau)^2\tilde{\mu}_{\overline{AB}} - 3\tau(1-\tau)(\tilde{\mu}_{A\overline{B}} + \tilde{\mu}_{\overline{A}B}) + \tau^2\tilde{\mu}_{AB}\right] \end{cases} \quad (2)$$

Based on the same hypotheses, the estimated log-means can be re-examined to take this error rate into account. Contrast can then be used to test if some of them are significantly different from 0 or not. For instance, if $\mu_{A\overline{B}}$ is not significantly different from 0, then it means that A is probably never edited before B is - unless $\mu_{\overline{A}B}$ also does not significantly differ from 0, then whether sites A and B are simultaneously edited, or they are both never edited.

A.3 Impact of the second condition on the maturation at one site

Recall that, with two conditions, the model can be written as follows:

$$\begin{cases} log(\mu_{ABYi}) = \mu_0 + A + B + AB + Y + AY + BY + ABY \\ log(\mu_{A\overline{B}Yi}) = \mu_0 + A + Y + AY \\ log(\mu_{\overline{A}BYi}) = \mu_0 + B + Y + BY \\ log(\mu_{\overline{AB}Yi}) = \mu_0 + Y \\ log(\mu_{ABXi}) = \mu_0 + A + B + AB \\ log(\mu_{A\overline{B}Xi}) = \mu_0 + A \\ log(\mu_{\overline{A}BXi}) = \mu_0 + B \\ log(\mu_{\overline{AB}Xi}) = \mu_0 \end{cases}$$

Thus, one can derive the effect of the condition Y compared to condition X on the maturation of site A (or, symmetrically, B) by looking at the ratio (or log-difference) between the count of reads having site A matured and the count of reads with site A non-matured, and then looking at the difference between the two conditions:

$$\begin{aligned} C_{Y\to A} &= \left[(log(\mu_{ABYi}) + log(\mu_{A\overline{B}Yi})) - (log(\mu_{\overline{A}BYi}) + log(\mu_{\overline{AB}Yi}))\right] \\ &\quad - \left[(log(\mu_{ABXi}) + log(\mu_{A\overline{B}Xi})) - (log(\mu_{\overline{A}BXi}) + log(\mu_{\overline{AB}Xi}))\right] \\ &= \left[(\mu_0 + A + B + AB + Y + AY + BY + ABY + \mu_0 + A + Y + AY)\right. \\ &\quad \left. - (\mu_0 + B + Y + BY + \mu_0 + Y)\right] - \left[(\mu_0 + A + B + AB + \mu_0 + A) - (\mu_0 + B + \mu_0)\right] \\ &= (2A + AB + 2AY + ABY) - (2A + AB) \\ &= 2AY + ABY \end{aligned}$$

B comaturationTracker package

B.1 Installation

To install the *comaturationTracker* R package, run the following code:

```
install.packages("devtools")
library(devtools)
devtools::install_github("SimiliSerpent/comaturationTracker")
library(comaturationTracker)
```

B.2 Runtime

I present in TABLE 1 the runtime of the different functions of the package. Wild-type RNAs are used as input, along with the list of 43 editing and 25 splicing maturation sites.

Function	Runtime	Input size
`loadReads()`	51.90 sec	6199156 reads
`getStates()`	22.66 min	386885 reads
`buildCountsDF()`	1.07 min	386885x68 matrix

Table 1: Runtime of different functions in the *comaturationTracker* package.

B.3 List of co-maturations found

The list of co-maturations found with our new method in the *WT* and in the PNP-mutant are shown in Table 2 and 3, respectively.

Site A		Site B		p-value	Adjusted p-value
Type	Position/Range	Type	Position/Range		
editing	116281	editing	116290	5.601000×10^{-33}	6.889230×10^{-31}
editing	12707	splicing	11939-12653	7.437862×10^{-19}	4.574285×10^{-17}
editing	116281	editing	116494	3.775895×10^{-16}	1.548117×10^{-14}
editing	116290	editing	116494	6.808637×10^{-16}	2.093656×10^{-14}
editing	37092	editing	37161	3.334053×10^{-12}	8.201771×10^{-11}
editing	95608	editing	95644	1.380346×10^{-11}	2.829709×10^{-10}
editing	95608	editing	95650	2.145083×10^{-10}	3.769217×10^{-09}
editing	116494	editing	116785	5.260672×10^{-09}	8.088283×10^{-08}
editing	95608	splicing	95703-96387	1.339157×10^{-08}	1.830181×10^{-07}
editing	96579	editing	96698	7.217350×10^{-08}	8.877341×10^{-07}
editing	63985	editing	64109	1.471742×10^{-07}	1.543392×10^{-06}
editing	94999	editing	95644	1.631227×10^{-07}	1.543392×10^{-06}
editing	95644	editing	95650	1.527332×10^{-07}	1.543392×10^{-06}
editing	94999	editing	95650	1.853527×10^{-07}	1.628456×10^{-06}
editing	116290	editing	116785	3.014314×10^{-07}	2.471737×10^{-06}
editing	95608	editing	96579	3.815855×10^{-07}	2.933439×10^{-06}
editing	95650	editing	96579	4.247158×10^{-07}	3.072944×10^{-06}
editing	96419	splicing	95703-96387	6.534599×10^{-07}	4.465309×10^{-06}
editing	96698	splicing	95703-96387	8.514756×10^{-07}	5.512184×10^{-06}
editing	96579	splicing	95703-96387	1.178918×10^{-06}	7.250346×10^{-06}
editing	96579	editing	97016	1.552259×10^{-06}	9.091803×10^{-06}
editing	94999	editing	95608	1.698825×10^{-06}	9.497977×10^{-06}
editing	116290	editing	117166	3.225854×10^{-06}	1.725131×10^{-05}
editing	95608	editing	96698	3.958034×10^{-06}	2.028492×10^{-05}
editing	95644	splicing	95703-96387	4.225503×10^{-06}	2.078948×10^{-05}
editing	69942	splicing	70138-70652	1.630433×10^{-05}	7.713202×10^{-05}
editing	95650	splicing	95703-96387	1.937094×10^{-05}	8.824539×10^{-05}
editing	96419	editing	96579	2.556382×10^{-05}	1.122982×10^{-04}
editing	94622	editing	97016	2.754275×10^{-05}	1.168192×10^{-04}
editing	95644	editing	96698	3.992642×10^{-05}	1.612055×10^{-04}
editing	95650	editing	96698	4.062903×10^{-05}	1.612055×10^{-04}
editing	116281	editing	116785	1.070179×10^{-04}	4.113501×10^{-04}
editing	95644	editing	96579	1.243285×10^{-04}	4.580159×10^{-04}
editing	116494	editing	117166	1.266060×10^{-04}	4.580159×10^{-04}
editing	116281	editing	117166	1.334544×10^{-04}	4.689970×10^{-04}
splicing	74847-75650	splicing	76489-77197	2.537521×10^{-04}	8.669864×10^{-04}
editing	95650	editing	97016	3.186044×10^{-04}	1.059144×10^{-03}
editing	95608	editing	96419	6.677647×10^{-04}	2.161449×10^{-03}
editing	95650	editing	96419	9.864158×10^{-04}	3.111004×10^{-03}
editing	96419	editing	96698	1.286260×10^{-03}	3.955248×10^{-03}
editing	94999	editing	96698	1.613005×10^{-03}	4.839016×10^{-03}
editing	97016	splicing	95703-96387	1.658236×10^{-03}	4.856262×10^{-03}
editing	96419	editing	97016	1.738506×10^{-03}	4.972936×10^{-03}

Table 2: List of co-maturations found in the *WT* using the *comaturationTracker* R package.

Site A		Site B		p-value	Adjusted p-value
Type	Position/Range	Type	Position/Range		
editing	116281	editing	116290	2.055134×10^{-38}	2.794982×10^{-36}
editing	63985	editing	64109	2.506005×10^{-29}	1.704083×10^{-27}
editing	116290	editing	116494	3.046326×10^{-24}	1.381001×10^{-22}
editing	95608	editing	95650	3.608106×10^{-19}	1.226756×10^{-17}
editing	95644	editing	95650	1.578165×10^{-18}	4.292610×10^{-17}
editing	95608	editing	95644	5.105095×10^{-17}	1.157155×10^{-15}
editing	116281	editing	116494	2.213633×10^{-16}	4.300772×10^{-15}
editing	116494	editing	116785	3.804978×10^{-15}	6.468462×10^{-14}
editing	12707	splicing	11939-12653	6.030501×10^{-15}	9.112757×10^{-14}
editing	96579	editing	96698	1.033457×10^{-14}	1.405502×10^{-13}
editing	94999	editing	95650	2.658270×10^{-14}	3.286589×10^{-13}
editing	116281	editing	116785	2.926917×10^{-14}	3.317173×10^{-13}
editing	37092	editing	37161	5.542289×10^{-13}	5.798086×10^{-12}
editing	95608	splicing	95703-96387	1.607115×10^{-11}	1.561197×10^{-10}
editing	94999	editing	95608	3.292122×10^{-10}	2.984857×10^{-09}
editing	96698	editing	97016	4.085608×10^{-10}	3.472767×10^{-09}
editing	116290	editing	116785	5.761951×10^{-10}	4.609561×10^{-09}
editing	96579	splicing	95703-96387	1.876296×10^{-09}	1.417646×10^{-08}
editing	96698	splicing	95703-96387	2.304918×10^{-09}	1.649836×10^{-08}
editing	96579	editing	97016	3.519788×10^{-09}	2.393456×10^{-08}
editing	95608	editing	96579	4.350625×10^{-09}	2.817547×10^{-08}
editing	95650	splicing	95703-96387	9.499729×10^{-09}	5.872560×10^{-08}
editing	95644	splicing	95703-96387	2.494577×10^{-08}	1.475054×10^{-07}
editing	94999	editing	95225	3.584929×10^{-08}	1.875194×10^{-07}
editing	94999	editing	95644	3.324331×10^{-08}	1.875194×10^{-07}
editing	96419	splicing	95703-96387	3.527716×10^{-08}	$1.8751942 \times 10^{-07}$
editing	95225	editing	95608	9.751609×10^{-08}	4.736496×10^{-07}
editing	95608	editing	96698	9.715398×10^{-08}	4.736496×10^{-07}
editing	95644	editing	96579	1.654067×10^{-07}	7.757005×10^{-07}
editing	95650	editing	96579	2.224292×10^{-07}	1.008346×10^{-06}
editing	94999	splicing	95703-96387	5.667863×10^{-07}	2.486546×10^{-06}
editing	96419	editing	96579	7.520987×10^{-07}	3.177987×10^{-06}
editing	116494	editing	117166	7.711292×10^{-07}	3.177987×10^{-06}
editing	95608	editing	97016	2.182530×10^{-06}	8.730119×10^{-06}
editing	95644	editing	96698	3.105333×10^{-06}	1.206644×10^{-05}
editing	94999	editing	96419	4.637217×10^{-06}	1.751838×10^{-05}
editing	97016	splicing	95703-96387	5.612643×10^{-06}	2.063026×10^{-05}
editing	95650	editing	96698	6.304477×10^{-06}	2.256339×10^{-05}
editing	116290	editing	117166	9.502202×10^{-06}	3.313588×10^{-05}
splicing	74847-75650	splicing	76489-77197	1.141289×10^{-05}	3.880384×10^{-05}
editing	95650	editing	97016	1.285807×10^{-05}	4.265116×10^{-05}
editing	95225	editing	95650	1.568623×10^{-05}	5.079351×10^{-05}
editing	25779	editing	25992	1.913574×10^{-05}	6.052233×10^{-05}
editing	95650	editing	96419	2.262315×10^{-05}	6.992609×10^{-05}
editing	95608	editing	96419	4.553058×10^{-05}	1.376035×10^{-04}
editing	116281	editing	117166	5.445590×10^{-05}	1.610001×10^{-04}
editing	94999	editing	96579	7.981105×10^{-05}	2.309426×10^{-04}
editing	69942	splicing	70138-70652	8.767131×10^{-05}	2.484021×10^{-04}
editing	96419	editing	97016	1.447213×10^{-04}	4.016755×10^{-04}
editing	95644	editing	96419	1.844770×10^{-04}	4.934014×10^{-04}
editing	95644	editing	97016	1.850255×10^{-04}	4.934014×10^{-04}
editing	95225	editing	95644	3.545197×10^{-04}	9.272054×10^{-04}
editing	117166	editing	118858	5.009070×10^{-04}	1.285346×10^{-03}
editing	96419	editing	96698	5.143193×10^{-04}	1.295323×10^{-03}

Table 3: List of co-maturations found in the PNP-mutant.

B.4 Effect of normalization and replicate parameter

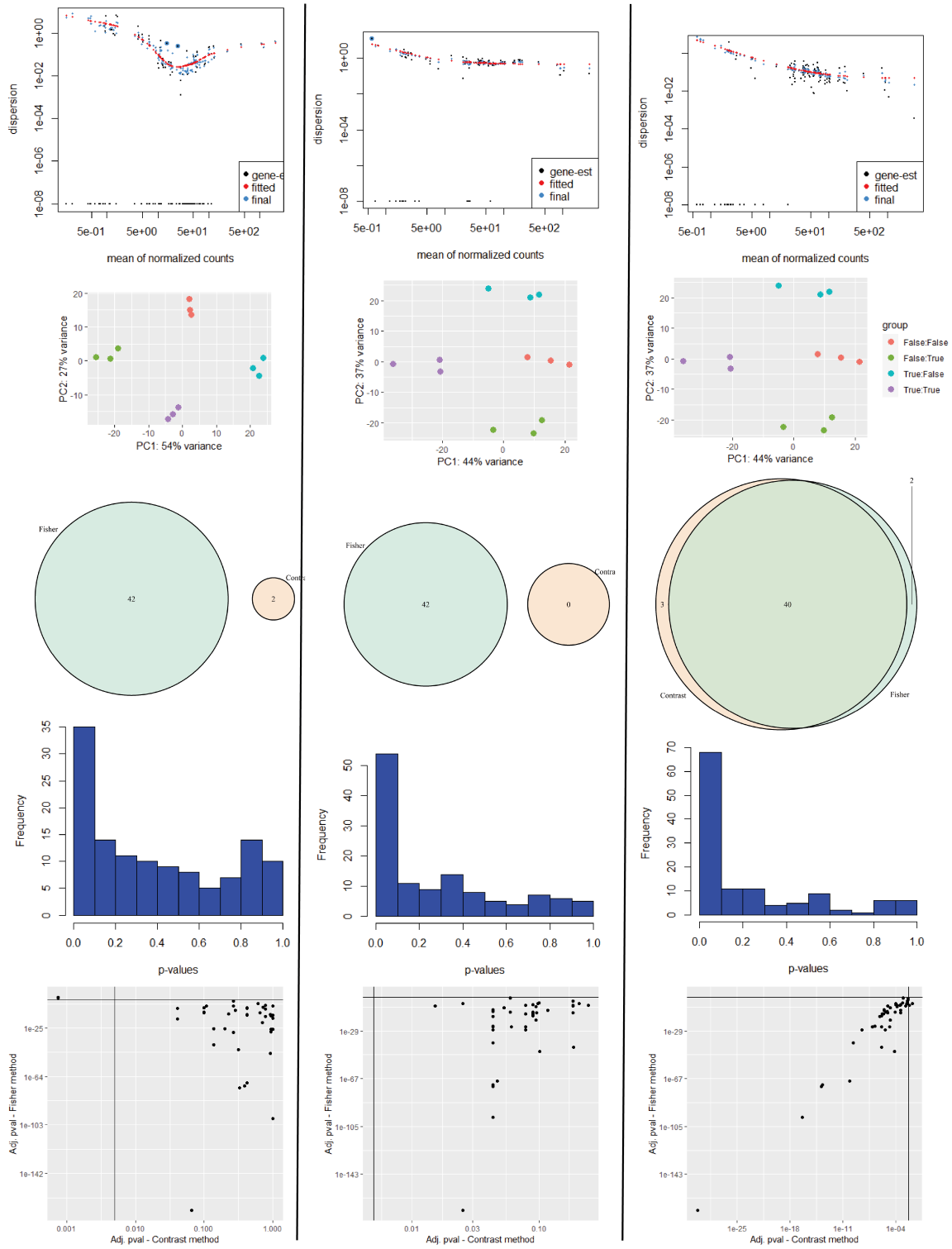


Figure 9: **Comparison of methodological designs - From Left to Right:** With *DESeq2* normalization, without replicate effect - Without *DESeq2* normalization, without replicate effect - Without *DESeq2* normalization, with replicate effect. **From Top to Bottom:** *DESeq2* dispersion plot - Principal Component Analysis - Venn diagram comparing the co-maturations found here and with the Fisher method - Raw p-values histogram - Adjusted p-values of all co-maturations found either with Fisher (p-values on Y axis) or Contrast method (p-values on X axis) (black lines materialize the 5×10^{-3} thresholds).

Chapter E

Résumé détaillé

E.1 Sur quoi et avec qui ?

Pendant ma recherche doctorale, à la croisée de la biologie, des statistiques, de la bioinformatique et de l'informatique, j'ai travaillé sur le développement et l'application de modèles statistiques, d'algorithmes et de méthodes pour l'analyse et l'interprétation des données biologiques à haut débit (séquençage). J'ai soumis ou publié trois articles de recherche en tant que premier auteur, ainsi qu'un autre article en tant que second auteur :

1. Liehrmann et al. [2021] est un article de recherche de modélisation où, en collaboration avec Guillem Rigaill et Toby Hocking (Université du Nord de l'Arizona), j'ai comparé différents modèles de détection de ruptures multiples et des heuristiques bioinformatiques spécialisées dans le contexte de la détection de marques épigénétiques ;

2. Liehrmann et al. [2023] est un article de recherche méthodologique et appliqué où, en collaboration avec Étienne Delannoy, Guillem Rigaill et Benoît Castandet, j'ai introduit DiffSegR, une méthode conçue pour identifier les différences d'expression à l'échelle du transcriptome entre deux conditions biologiques dans les données de séquençage d'ARN ;

3. Liehrmann and Rigaill [2023] est un article de recherche algorithmique où, en collaboration avec Guillem Rigaill, j'ai introduit Ms.FPOP, un algorithme de détection de ruptures multiples rapide et exact incorporant une pénalité à multi-échelles [Verzelen et al., 2020] ;

4. Guilcher et al. [2021] est un article de recherche appliqué où nous avons étudié la coordination des événements de maturation de l'ARN des chloroplastes à l'échelle du transcriptome en utilisant des données de séquençage d'ARN basées sur Nanopore.

Ce dernier article a été rendu possible grâce au développement d'une méthode appelée comaturationTrackeR. Ce projet collaboratif a débuté avec Chloé Seyman, une étudiante en licence, et s'est poursuivi avec Benjamin Vacus, un étudiant en master. J'ai eu l'opportunité de co-superviser Chloé et Benjamin pendant les deux premières années de ma recherche doctorale.

Dans les chapitres suivants de ce manuscrit, je propose différentes perspectives sur un ou plusieurs de ces articles de recherche, qui peuvent être trouvés en annexe. Je recommande au lecteur de lire d'abord les chapitres introductifs 3 et 4 dans leur intégralité, puis de se référer à la Figure E.1 pour une investigation plus approfondie d'un problème particulier. Comme le montre la Figure E.1, les lecteurs peuvent choisir de lire les chapitres 5 et 6 dans l'ordre qui leur convient le mieux. Néanmoins, il peut être bénéfique de se familiariser d'abord avec le modèle standard de ruptures multiples présenté dans le Chapitre 5, car il est au cœur de la méthode DiffSegR introduite dans le Chapitre 6.

E.2 Chapitre 3 : Introduction

Ma thèse explore principalement le transcriptome, qui désigne l'ensemble des molécules d'ARN générées dans une cellule, un tissu ou un organisme spécifique à un stade de développement ou physiologique particulier. Deux de mes articles de recherche, Liehrmann et al. [2023] et Guilcher et al. [2021], portent directement sur son analyse. Pour contextualiser ces articles, dans le Chapitre 3, qui fait également office d'introduction générale, j'illustre une perspective multi-échelle de l'analyse du transcriptome (Section 3.2.1), allant du gène, événement, paire d'événements, jusqu'aux isoformes. Je souligne une série de défis comprenant des facteurs techniques, statistiques et biologiques rencontrés à chaque échelle (Section 3.2.2). Ces défis sont particuliè-

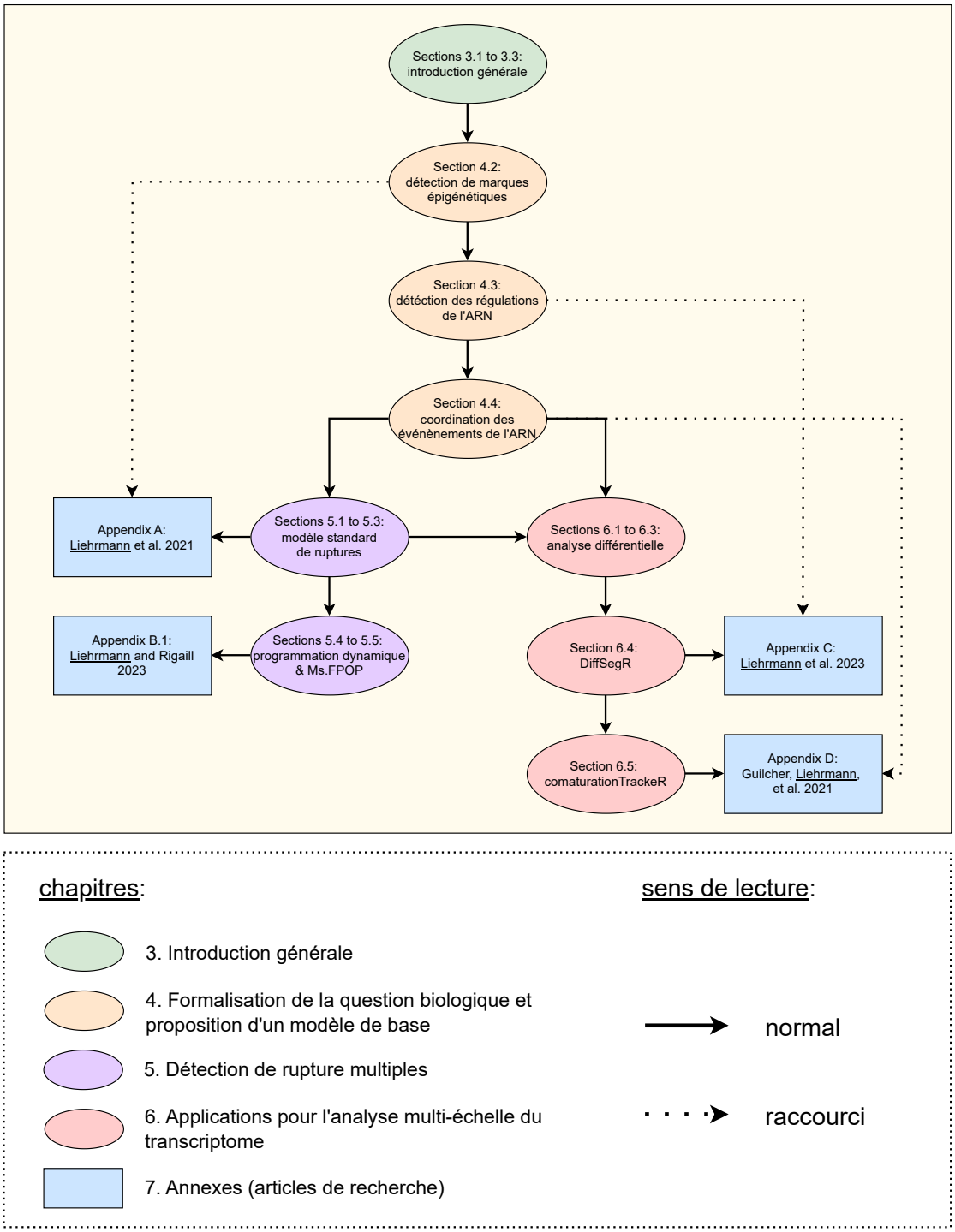


FIGURE E.1 – Dépendances entre les sections de ce manuscrit.

rement aigus au niveau des isoformes. En conclusion, je suggère deux stratégies, Stratégie 1 et Stratégie 2, pour améliorer l'analyse du transcriptome (Section 3.2.3). Avec mes co-auteurs, nous avons employé la Stratégie 1 et la Stratégie 2 dans Liehrmann et al. [2023]. Nous avons également appliqué la Stratégie 2 dans Guilcher et al. [2021].

E.2.1 Une feuille de route pour améliorer l'analyse du transcriptome

Dans la Section 3.2.2, je présente comment l'étude d'un nombre croissant d'événements le long des molécules d'ARN peut rendre l'analyse du transcriptome plus complexe d'un point de vue statistique, technique et biologique. Pour contourner la complexité exponentielle de l'étude de chaque isoforme individuelle tout en permettant aux chercheurs d'avoir un aperçu plus détaillé de la régulation génique par rapport à une analyse agrégée au niveau du gène, une approche prometteuse consiste à :

Strategy 1

☞ développer des méthodes qui examinent simultanément un nombre petit d'événements.

Cela peut être réalisé soit en étudiant chaque événement indépendamment, soit en analysant conjointement quelques événements (par exemple, par paires). Dans ce contexte, l'utilisation de technologies de lectures dites longues peut être bénéfique pour surveiller conjointement des événements d'ARN qui peuvent être séparés par des centaines, voire des milliers de nucléotides.

Une autre considération est que la précision des résultats de l'analyse à chaque échelle dépend fortement de la qualité des annotations, qui sont connues pour être incomplètes pour les gènes, les événements et par conséquent, les isoformes. Ainsi, pour améliorer l'analyse du transcriptome, une autre approche prometteuse consiste à :

Strategy 2

☞ développer des méthodes qui analysent le transcriptome sans se fier aux annotations préexistantes.

Ces techniques sont communément reconnues comme des approches data-driven.

E.3 Chapitre 4 : Formalisation de la question biologique et proposition d'un modèle de base

Dans ce chapitre, je présente les questions biologiques que j'ai étudiées, ainsi que les problèmes statistiques correspondants, et les modèles statistiques que j'ai proposés pour aborder ces problèmes spécifiques. Mon objectif était de proposer des modèles simples qui faciliteraient l'interprétation des données pour les biologistes, et par conséquent, amélioreraient la communication interdisciplinaire. De plus, j'ai cherché à tirer parti des méthodologies existantes chaque fois que cela était possible. Plus précisément, j'ai appris grâce à une première expérience sur la détection de marques épigénétiques (Section 4.2), puis confirmé par une autre expérience sur la détection des régulations de l'ARN (Section 4.3), que des modèles plus simples, bien que parfois mathématiquement insatisfaisants, peuvent être simultanément (1) plus faciles à comprendre pour les non-spécialistes, (2) plus faciles à mettre en œuvre et à calibrer, et (3) étonnamment efficaces, voire supérieurs pour répondre à la question biologique. Par conséquent, je pense que de tels modèles devraient être privilégiés. De plus, reconnaissant que dans le pire des cas, ces modèles peuvent être moins efficaces, ils jouent néanmoins un rôle crucial en justifiant la nécessité de développer et d'implémenter des modèles plus sophistiqués. Ce principe de parcimonie, auquel je souscris pleinement, m'a guidé tout au long de ma recherche doctorale, en particulier lorsque j'ai travaillé sur la détection des régulations de l'ARN (Section 4.3) et des co-maturations (Section 4.4).

E.3.1 Résumé du chapitre en un coup d'œil

1. Dans la Section 4.2, je discute du problème de la détection des marques épigénétiques, en partant de l'objectif biologique (Question biologique 1) et en procédant à la formulation

du problème statistique respectif (Problème statistique 1). Un état de l'art des méthodes de pointe, récemment conçues pour aborder le problème statistique, est ensuite exposé. Finalement, je présente un modèle de base (Modèle de base 1), qui repose purement sur des principes conventionnels de transformation et de segmentation du signal, développés respectivement dans les années 1940 et 1980. L'efficacité de cette solution de base est observée comme étant aussi précise, sinon supérieure, aux avancées récentes. Cette comparaison est élucidée dans Liehrmann et al. [2021].

2. Dans la Section 4.3, reprenant la structure de la Section 4.2, je m'intéresse au problème de la détection des régulations de l'ARN. Finalement, je rappelle le modèle standard de ruptures multiples précédemment utilisé dans la détection des marques épigénétiques. Une fois de plus, ce modèle est démontré supérieur aux méthodes de pointe dans la détection des régulations de l'ARN, comme détaillé dans Liehrmann et al. [2023].

3. Dans la Section 4.4, je présente brièvement le problème de l'étude de la coordination des événements d'ARN, un problème sur lequel j'ai co-supervisé deux stagiaires durant la première et la deuxième année de ma thèse.

E.4 Chapitre 5 : Détection de ruptures multiples

Avant de commencer cette thèse, ma vision succincte d'un projet interdisciplinaire réussi impliquait le développement d'un nouveau modèle statistique ou d'un nouvel algorithme pour chaque nouveau projet biologique (la question et les données). Cependant, cette vision a rapidement évolué. Comme je l'ai démontré dans le Chapitre 4, il peut être judicieux d'économiser sur le développement en proposant ou en adaptant un modèle ou algorithme existant et éprouvé. Néanmoins, faire confiance aux méthodologies existantes implique également de continuer à développer des modèles et algorithmes intéressants. Tout au long de ma recherche doctorale, j'ai mis en pratique cette vision révisée de la recherche interdisciplinaire. Dans ce chapitre, je com-

mence par introduire un modèle standard de ruptures multiples que j'ai employé dans la détection de marques épigénétiques [Liehrmann et al., 2021] et la détection de régulations de l'ARN [Liehrmann et al., 2023], avec des résultats prometteurs. Dans la seconde partie de ce chapitre, j'introduis un nouvel algorithme de détection de ruptures multiples, Ms.FPOP, qui intègre une pénalité multi-échelle avec de meilleures propriétés statistiques que les pénalités précédemment introduites dans la littérature [Liehrmann and Rigaill, 2023].

E.4.1 Détection des ruptures dans la moyenne

La détection de ruptures multiples, un problème de régression, est un domaine de recherche actif depuis les années 1950 [Page, 1954, 1957, Girshick and Rubin, 1952]. Initialement motivé par un besoin de contrôle de qualité dans les opérations de fabrication, il est désormais désormais considéré comme l'un des « grands défis de l'inférence » dans l'analyse de données massives, selon le Conseil National de Recherche des États-Unis [Council et al., 2013]. La détection de ruptures multiples est importante dans un large éventail de disciplines, y compris la génomique [Muggeo and Adelfio, 2010], les neurosciences [Koepcke et al., 2016], l'économétrie [Bai, 1997], la sécurité des réseaux informatiques [Tartakovsky, 2014], et la recherche climatique [Reeves et al., 2007].

Le problème de détection de ruptures multiples le plus typique et le plus répandu est l'identification des changements abrupts dans la moyenne d'un signal univarié ordonné, comme ceux manifestés dans le temps ou le long du génome. Ces décalages soudains, connus sous le nom de ruptures, délimitent des segments caractérisés par un signal homogène. Dans le contexte de ma recherche, ces ruptures peuvent signifier soit le début/la fin d'un pic dans les données ChIP-Seq, soit le début/la fin d'une région différentiellement exprimée dans les données RNA-Seq issues de deux conditions biologiques distinctes. Dans les deux scénarios, ces ruptures révèlent des événements biologiques, tels que des régions génomiques enrichies en marqueurs épigénétiques H3K4me3 ou des disparités dans les processus de maturation de l'ARN.

La Figure E.2.A illustre un exemple de profil de transcription différentielle issu d'une ex-

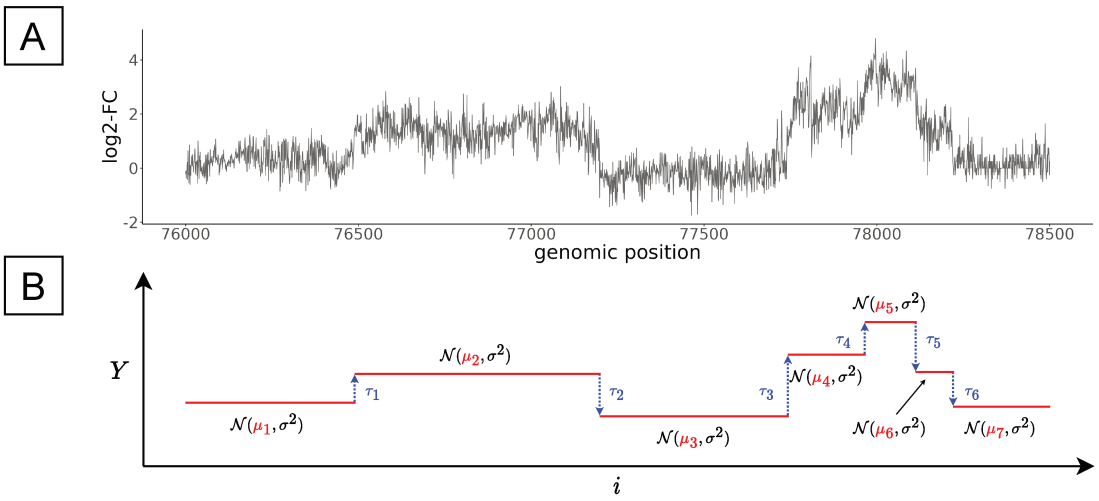


FIGURE E.2 – La moyenne du $\log_2$-FC par base est affectée par plusieurs ruptures notables. (A) Le $\log_2$-FC par base a été calculé à partir de données RNA-Seq comparant deux conditions biologiques, spécifiquement pour les positions 76000 à 78500 du génome du chloroplaste. Plusieurs ruptures peuvent être visuellement identifiés, notamment autour des positions 76500 et 77250. Ces points de changement marquent un événement biologique spécifique : l'accumulation d'un intron dans l'une des deux conditions, comme détaillé dans le Tableau 1 de Liehrmann et al. [2023]. (B) Je présente ci-dessous un schéma du modèle standard de ruptures multiples appliqué au même profil de transcription différentielle. Chaque segment j est délimité par deux ruptures, τ_{j-1} et τ_j. Dans ce segment, les observations Y_i sont indépendantes et suivent une distribution gaussienne avec une moyenne μ_j et une variance σ^2.

périence de séquençage d'ARN impliquant deux conditions distinctes. Le signal présente des variations significatives dans sa moyenne. Bien que la détection de ces changements puisse initialement sembler facile à repérer à l'œil, c'est en réalité un problème difficile. Une manière de comprendre le défi est de considérer le nombre de segmentations potentielles d'un profil avec n observations. Chaque point, à l'exception du dernier, peut servir de rupture, résultant en un total de $n-1$ ruptures possibles. Par conséquent, le nombre de segmentations possibles atteint 2^{n-1}. Par exemple, pour $n = 100$, le nombre total de segmentations dépasse 6×10^{29}. Cela soulève de nombreux problèmes statistiques et algorithmiques.

Dans ce contexte, lors de ma thèse, j'ai développé [Liehrmann and Rigaill, 2023] et appliqué à des données génomiques [Liehrmann et al., 2021, 2023] plusieurs algorithmes de détection de ruptures multiples qui maximisent une vraisemblance pénalisée. Cette approche, profondément ancrée dans les statistiques traditionnelles, offre des garanties statistiques à la fois asymptotiques [Yao and Au, 1989, Boysen et al., 2009] et non asymptotiques [Lebarbier, 2005, Garreau and Ar-

lot, 2018, Arlot et al., 2019] pour l'estimation du signal et la détection de ruptures multiples. Son efficacité computationnelle est particulièrement adaptée aux exigences intensives de l'analyse des données génomiques, où il est courant de traiter des profils avec des millions d'observations [Rigaill, 2015, Maidstone et al., 2016]. Enfin, des preuves empiriques issues à la fois de simulations [Fearnhead and Rigaill, 2020] et d'applications réelles offrent fréquemment des résultats satisfaisants, démontrant son efficacité. Notamment, elle est déjà considérée comme l'état de l'art dans de nombreuses applications génomiques [Lai et al., 2005, Hocking et al., 2013a, Cleynen et al., 2014b, Hocking et al., 2016].

E.4.2 Résumé du chapitre en un coup d'œil

1. Dans la Section 5.3, je présente un modèle standard pour la détection de ruptures multiples, ainsi que le problème de vraisemblance pénalisée associé. J'ai mis en œuvre ce modèle sur des données de ChIP-Seq dans Liehrmann et al. [2021], et sur des données de RNA-Seq dans Liehrmann et al. [2023], respectivement comme solutions pratiques pour la détection de pics et de régions différentiellement exprimées candidates. Divers algorithmes de programmation dynamique visant à maximiser la vraisemblance pénalisée ont été proposés au fil des ans. Je présente quelques-uns de ces algorithmes dans la seconde partie de cette première section.

2. Dans la Section 5.5, je présente une nouvelle pénalité multi-échelle, introduite par Verzelen et al. [2020], qui possède des propriétés statistiques supérieures en termes de détection et de localisation par rapport aux autres pénalités documentées dans la littérature. Par la suite, j'introduis un nouvel algorithme de segmentation, Ms.FPOP, qui utilise des techniques d'élagage fonctionnel pour minimiser efficacement un critère des moindres carrés avec cette pénalité multi-échelle comme présenté dans Liehrmann and Rigaill [2023].

E.5 Chapitre 6 : Application à l'analyse multi-échelle du transcriptome

Ce chapitre met en lumière l'aspect ingénierie de ma thèse. Je commence par formuler ma stratégie pour une analyse précise et rigoureuse des différences d'expression et des co-maturations. Cette stratégie s'appuie sur le modèle DESeq2 et inclut le contrôle des différences évaluées, par exemple, en utilisant une procédure post-hoc. Par la suite, je détaille comment j'ai intégré cette stratégie dans deux packages R—DiffSegR et comaturationTrackeR. Ces outils illustrent l'intégration réussie de méthodologies analytiques complexes dans des solutions logicielles pratiques et conviviales.

E.5.1 Analyse différentielle

Un aspect important de la détection transcriptomique des différences d'expression et des co-maturations est la quantification des changements systématiques entre deux groupes, également connue sous le nom d'analyse différentielle. Dans le premier cas, le changement concerne le niveau d'expression d'un site en fonction de la condition biologique ; dans le second, il se rapporte au niveau de maturation d'un site, en fonction de l'état de maturation d'un second site. Quantifier ces changements est un défi car les niveaux d'expression et de maturation d'un site peuvent varier entre les échantillons. Pour tenir compte de cette variabilité, à la fois technique et biologique, il est crucial de modéliser efficacement les comptages par événement ou par paire d'événements. Le modèle linéaire généralisé avec une distribution négative binomiale pour les données RNA-Seq, tel qu'implémenté dans le package R DESeq2 Love et al. [2014], réalise cette tâche de manière satisfaisante.

E.5.2 Résumé du chapitre en un coup d'œil

1. La Section 6.3 présente les éléments clés du modèle statistique des comptages par gènes implémenté dans DESeq2.

2. Dans la Section 6.4, je dévoile comment utiliser le modèle statistique de DESeq2 pour évaluer les régions différentiellement exprimées candidates identifiées à l'aide de FPOP. Ceci est suivi par une courte présentation de DiffSegR, un package R qui intègre le Modèle de Base 2 et DESeq2 comme montré dans Liehrmann et al. [2023].

3. Dans la Section 6.5, je fournis une brève introduction de comaturationTrackeR, une méthode qui existe sous deux formes : un pipeline R publié [Guilcher et al., 2021] et un package R (toujours en développement). La seconde version utilise également le modèle statistique de DESeq2 pour évaluer les co-maturations.

E.6 Chapitre 7 : Discussion

Dans le Chapitre 7, je présente quelques perspectives relatives aux études menées au cours de cette thèse.