Quelques développements statistiques et algorithmiques pour l'analyse de données génomiques

Guillem Rigaill

Université Paris-Saclay

École doctorale de mathématiques Hadamard (ED 574)

Laboratoire de Mathématiques et Modélisation d'Évry
&
L'Institut des Sciences des Plantes - Paris-Saclay

Mémoire présenté pour l'obtention du

Diplôme d'habilitation à diriger les recherches

Discipline : Mathématiques

par

Guillem RIGAILL

Quelques développements statistiques et algorithmiques pour l'analyse de données génomiques

Rapporteurs :
Anne-Laure BOULESTEIX
David CAUSEUR
Sophie SCHBATH

Date de soutenance : 18 Septembre 2020

Composition du jury :

Anne-Laure BOULESTEIX	(Rapportrice)
David CAUSEUR	(Rapporteur)
Sophie SCHBATH	(Rapportrice)
Sylvain ARLOT	(Examinateur)
Avner BAR-HEN	(Examinateur)
Jean-Philippe VERT	(Examinateur)

Remerciements

> - Roi Loc, répondit Nur, je pourrais savoir beaucoup de choses et n'être qu'un imbécile. Mais je connais le moyen d'apprendre quelques-unes des innombrables choses que j'ignore, et c'est pourquoi je suis justement renommé comme un savant
>
> Abeille - Anatole France

Résumé

Qui remercie tous les étudiants, les scientifiques et les savants avec qui j'ai eu la chance de travailler, collaborer et discuter. Qui remercie aussi mes amis et ma famille.

Pour commencer, je souhaite remercier Anne-Laure Boulesteix, David Causeur et Sophie Schbath d'avoir accepté de rapporter mon mémoire. Je remercie également Sylvain Arlot, Avner Bar-Hen et Jean-Philippe Vert d'avoir accepté de faire partie de mon jury. Soutenir mon habilitation devant vous, à distance ou en présentiel, fut un honneur et un plaisir.

Je souhaite également remercier les nombreuses personnes avec qui j'ai travaillé et collaboré ces dernières années. Je ne me hasarderai pas à en faire une liste exhaustive, ma mémoire me jouerait des tours. Je remercie plus particulièrement Marie-Laure Martin-Magniette, Julien Chiquet, Toby Hocking, Paul Fearnhead et Vincent Runge.

Je tiens aussi à remercier tous ceux que j'ai eu l'occasion de côtoyer professionnellement depuis une dizaine d'années notamment à l'IPS2, au LaMME, dans l'équipe GNet, dans l'équipe Stats et Génome et lors de séminaires et conférences. J'ai beaucoup appris et évolué grâce à vous.

Je remercie tous ceux qui soutiennent et supportent la recherche, il me semble que sans vous « le monde de la recherche » serait bien peu de chose.

Je remercie mes infatigables relecteurs : Marie-Laure, Cécile, mon père, Julien, Vincent, Pierre.

Je remercie enfin mes amis et ma famille.

Je remercie Erika et Alessio, pour leur amour, leur patience et leur soutien constants. Sans vous je ne serais pas qui je suis.

Je supplie les personnes que j'ai oubliées ou que je n'ai pas remerciées autant que nécessaire de me pardonner.

Table des matières

Chapitre 1

Production scientifique

> Le laissant à ses réflexions, j'ouvris un livre que je lus avec intérêt, car c'était un catalogue de manuscrits. Je ne sais pas de lecture plus facile, plus attrayante, plus douce que celle d'un catalogue.
>
> Le Crime de Sylvestre Bonnard - Anatole France

Résumé

Je remercie mes co-auteurs sans qui je n'aurais pu réaliser ces travaux.

1.1 Publications méthodologiques

Dans des journaux

[1]T. D. Hocking, Guillem Rigaill, P. Fearnhead et G. Bourque, « Generalized Functional Pruning Optimal Partitioning (GFPOP) for Constrained Changepoint Detection in Genomic Data », **Journal of Statistical Software (accepté)** (2020).

[2]T. D. Hocking, Guillem Rigaill, P. Fearnhead et G. Bourque, « A log-linear time algorithm for constrained changepoint detection », **JMLR** (2020).

[3]A. Hulot, J. Chiquet, F. Jaffrézic et Guillem Rigaill, « Fast tree aggregation for consensus hierarchical clustering », **BMC bioinformatics 21**, 1-12 (2020).

[4]C. Ambroise, A. Dehman, P. Neuvial, Guillem Rigaill et N. Vialaneix, « Adjacency-constrained hierarchical clustering of a band similarity matrix with application to Genomics. », **Algorithms for Molecular Biology** (2019).

[5]J. Chiquet, G. Rigaill et M. Sundqvist, « A multiattribute Gaussian graphical model for inferring multiscale regulatory networks : an application in breast cancer », in **Gene Regulatory Networks** (Springer, 2019), p. 143-160.

[6]A. Celisse, G. Marot, M. Pierre-Jean et Guillem Rigaill, « New efficient algorithms for multiple change-point detection with reproducing kernels », **Computational Statistics & Data Analysis 128**, 200-220 (2018).

[7]P. Fearnhead et Guillem Rigaill, « Changepoint detection in the presence of outliers », **Journal of the American Statistical Association**, 1-15 (2018).

[8]J. Chiquet, Y. Grandvalet et Guillem Rigaill, « On coding effects in regularized categorical regression », **Statistical Modelling 16**, 228-237 (2016).

[9]R. Maidstone, T. Hocking, Guillem Rigaill et P. Fearnhead, « On optimal multiple changepoint algorithms for large data », **Statistics and Computing**, 1-15 (2016).

[10]J. Chiquet, P. Gutierrez et Guillem Rigaill, « Fast tree inference with weighted fusion penalties », **Journal of Computationnal and Graphical Statistics** (2015).

[11]Guillem 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**, 150-175 (2015).

[12]A. Cleynen, T. M. Luong, Guillem Rigaill et G. Nuel, « Fast estimation of the Integrated Completed Likelihood criterion for change-point detection problems with applications to Next-Generation Sequencing data », **Signal Processing 98**, 233-242 (2014).

[13]A. Cleynen, M. Koskas, E. Lebarbier, Guillem Rigaill et S. Robin, « Segmentor3IsBack : an R package for the fast and exact segmentation of Seq-data. », **Algorithms for Molecular Biology 9**, 6 (2014).

[14]Guillem Rigaill, S. Balzergue, V. Brunaud, E. Blondet, A. Rau, O. Rogier, J. Caius, C. Maugis-Rabusseau, L. Soubigou-Taconnat, S. Aubourg, L. Claire, M.-M. Marie-Laure et D. Etienne, « Synthetic data sets for the identification of key ingredients for RNA-seq differential analysis », **Briefings in bioinformatics** (2016).

[15]M. Pierre-Jean, Guillem Rigaill et P. Neuvial, « Performance evaluation of DNA copy number segmentation methods », **Briefings in bioinformatics**, bbu026 (2014).

[16]Guillem Rigaill, S. Cadot, R. J. Kluin, Z. Xue, R. Bernards, I. J. Majewski et L. F. Wessels, « A regression model for estimating DNA copy number applied to capture sequencing data », **Bioinformatics 28**, 2357-2365 (2012).

[17]Guillem Rigaill, É. Lebarbier et S. Robin, « Exact posterior distributions and model selection criteria for multiple change-point detection problems », **Statistics and Computing 22**, 917-929 (2012).

[18]F. Picard, E. Lebarbier, M. Hoebeke, Guillem Rigaill, B. Thiam et S. Robin, « Joint segmentation, calling, and normalization of multiple CGH profiles », **Biostatistics** (2011).

[19]Guillem Rigaill, P. Hupé, A. Almeida, P. La Rosa, J.-P. Meyniel, C. Decraene et E. Barillot, « ITALICS : an algorithm for normalization and DNA copy number calling for Affymetrix SNP arrays », **Bioinformatics 24**, 768-774 (2008).

Dans des actes de conférences

[20]T. D. Hocking, Guillem Rigaill et G. Bourque, « PeakSeg : constrained optimal segmentation and supervised penalty learning for peak detection in count data », in Proceedings of The 32nd International Conference on Machine Learning (2015), p. 324-332.

[21]Guillem Rigaill, T. Hocking, J.-P. Vert et F. Bach, « Learning sparse penalties for change-point detection using max margin interval regression », in Proceedings of The 30th International Conference on Machine Learning (2013), p. 172-180.

[22]Guillem Rigaill, E. Lebarbier et S. Robin, « Exact posterior distributions over the segmentation space and model selection for multiple change-point detection problems », in **Proceedings of COMPSTAT'2010** (Springer, 2010), p. 557-564.

1.2 Publications dans un domaine d'application

En bioinformatique

[23]B. Malbert, Guillem Rigaill, V. Brunaud, C. Lurin et E. Delannoy, « Bioinformatic Analysis of Chloroplast Gene Expression and RNA Posttranscriptional Maturations Using RNA Sequencing », in **Plastids** (Springer, 2018), p. 279-294.

[24]R. Zaag, J. P. Tamby, C. Guichard, Z. Tariq, Guillem Rigaill, E. Delannoy, J.-P. Renou, S. Balzergue, T. Mary-Huard, S. Aubourg, M. Martin-Magniette et V. Brunaud, « GEM2Net : from gene expression modeling to-omics networks, a new CATdb module to investigate Arabidopsis thaliana genes involved in stress response », **Nucleic acids research**, gku1155 (2014).

[25]T. D. Hocking, V. Boeva, Guillem Rigaill, G. Schleiermacher, I. Janoueix-Lerosey, O. Delattre, W. Richer, F. Bourdeaut, M. Suguro, M. Seto et F. Bach, « SegAnnDB : interactive Web-based genomic segmentation », **Bioinformatics 30**, 1539-1546 (2014).

[26]J. J. de Ronde, Guillem Rigaill, S. Rottenberg, S. Rodenhuis et L. F. Wessels, « Identifying subgroup markers in heterogeneous populations », **Nucleic acids research**, gkt845 (2013).

[27]T. Popova, E. Manié, D. Stoppa-Lyonnet, Guillem Rigaill, E. Barillot et M.-H. Stern, « Genome Alteration Print (GAP) : a tool to visualize and mine complex cancer genomic profiles obtained by SNP arrays », **Genome Biology 10**, R128-R128 (2009).

En biologie

[28]V. S. G. de la Torre, C. Majorel-Loulergue, Guillem Rigaill, D. A. Gonzalez, L. Soubigou-Taconnat, Y. Pillon, L. Barreau, B. Fogliani, V. Burtet-Sarramegna, S. Merlot et al., « Wide cross-species RNA-Seq comparison reveals a highly conserved role for Ferroportins in nickel hyperaccumulation in plants », **New Phytologist (accepté)** (2020).

[29]M.-A. Lemay, D. Torkamaneh, Guillem Rigaill, B. Boyle, A. O. Stec, R. M. Stupar et F. Belzile, « Screening populations for copy number variation using genotyping-by-sequencing : a proof of concept using soybean fast neutron mutants », **BMC genomics 20**, 1-16 (2019).

[30]S. Maubant, B. Tahtouh Tania, Amélie, V. Maire, F. Némati, B. Tesson, M. Ye, Guillem Rigaill, M. Noizet, A. Dumont, D. Gentien, M.-P. Bérengère, de Koning Leanne, S. F. Mahmood, D. Decaudin, F. Cruzalegui, T. G. C, S. Roman-Roman et T. Dubois, « LRP5 regulates the expression of STK40, a new potential target in triple-negative breast cancers », **Oncotarget** (2018).

[31]V. Maire, F. Mahmood, Guillem Rigaill, M. Ye, A. Brisson, F. Némati, D. Gentien, G. C. Tucker, S. Roman-Roman et T. Dubois, « LRP8 is overexpressed in estrogen-negative breast cancers and a potential target for these tumors », **Cancer medicine** (2018).

[32]E. Albert, R. Duboscq, M. Latreille, S. Santoni, M. Beukers, J.-P. Bouchet, F. Bitton, J. Gricourt, C. Poncet, V. Gautier, J. M. Jiménez-Gómez, Guillem Rigaill et M. Causse, « Allele specific expression and genetic determinants of transcriptomic variations in response to mild water deficit in tomato », **The Plant Journal** (2018).

[33]A. Lloyd, A. Blary, D. Charif, C. Charpentier, J. Tran, S. Balzergue, E. Delannoy, Guillem Rigaill et E. Jenczewski, « Homoeologous exchanges cause extensive dosage-dependent gene expression changes in an allopolyploid crop », **New Phytologist** (2018).

[34]J.-T. Brandenburg, T. Mary-Huard, Guillem Rigaill, S. J. Hearne, H. Corti, J. Joets, C. Vitte, A. Charcosset, S. D. Nicolas et M. I. Tenaillon, « Independent introductions and admixtures have contributed to adaptation of European maize and its American counterparts », **PLoS genetics 13**, e1006666 (2017).

[35]D. Guillaumot, M. Lopez-Obando, K. Baudry, A. Avon, Guillem Rigaill, A. F. de Longevialle, B. Broche, M. Takenaka, R. Berthomé, G. De Jaeger et al., « Two interacting PPR proteins are major Arabidopsis editing factors in plastid and mitochondria », **Proceedings of the National Academy of Sciences 114**, 8877-8882 (2017).

[36]A.-S. Dumas, L. Taconnat, E. Barbas, Guillem Rigaill, O. Catrice, D. Bernard, A. Benamar, D. Macherel, A. El Amrani et R. Berthomé, « Unraveling the early molecular and physiological mechanisms involved in response to phenanthrene exposure », **BMC genomics 17**, 818 (2016).

[37]C. Baldeyron, A. Brisson, B. Tesson, F. Némati, S. Koundrioukoff, E. Saliba, L. De Koning, E. Martel, M. Ye, Guillem Rigaill, D. Meseure, A. Nicolas, D. Gentien, D. Decaudin, M. Debatisse, S. Depil, F. Cruzalegui, A. Pierré, S. Roman-Roman, G. Tucker et T. Dubois, « TIPIN depletion leads to apoptosis in breast cancer cells », **Molecular oncology** (2015).

[38]S. Maubant, B. Tesson, V. Maire, M. Ye, Guillem Rigaill, D. Gentien, F. Cruzalegui, G. C. Tucker, S. Roman-Roman et T. Dubois, « Transcriptome analysis of Wnt3a-treated triple-negative breast cancer cells », **PloS one 10**, e0122333 (2015).

[39]V. Maire, F. Némati, M. Richardson, A. Vincent-Salomon, B. Tesson, Guillem Rigaill, E. Gravier, B. Marty-Prouvost, L. De Koning, G. Lang, D. Gentien, A. Dumont, E. Barillot, E. Marangoni, D. Decaudin, S. Roman-Roman, A. Pierré, F. Cruzalegui, S. Depil, G. Tucker et T. Dubois, « Polo-like kinase 1 : a potential therapeutic option in combination with conventional chemotherapy for the management of patients with triple-negative breast cancer », **Cancer research 73**, 813-823 (2013).

[40]A. Vincent-Salomon, V. Benhamo, E. Gravier, Guillem Rigaill, N. Gruel, S. Robin, Y. de Rycke, O. Mariani, G. Pierron, D. Gentien, F. Reyal, P. Cottu, A. Fourquet, R. Rouzier, X. Sastre-Garau et O. Delattre, « Genomic instability : a stronger prognostic marker than proliferation for early stage luminal breast carcinomas », **PloS one 8** (2013).

[41]V. Maire, C. Baldeyron, M. Richardson, B. Tesson, A. Vincent-Salomon, E. Gravier, B. Marty-Prouvost, L. De Koning, Guillem Rigaill, A. Dumont, D. Gentien, E. Barillot, S. Roman-Roman, S. Depil, F. Cruzalegui, A. Pierré, G. Tucker et T. Dubois, « TTK/hMPS1 is an attractive therapeutic target for triple-negative breast cancer », **Plos One 8** (2013).

[42]A. Toullec, D. Gerald, G. Despouy, B. Bourachot, M. Cardon, S. Lefort, M. Richardson, Guillem Rigaill, M.-C. Parrini, C. Lucchesi, D. Bellanger, M. Stern, T. Dubois, X. Sastre-Garau, O. Delattre, A. Vincent-Salomon et F. Mechta-Grigoriou, « Oxidative stress promotes myofibroblast differentiation and tumour spreading », **EMBO molecular medicine 2**, 211-230 (2010).

[43]F. Lizárraga, R. Poincloux, M. Romao, G. Montagnac, G. Le Dez, I. Bonne, Guillem Rigaill, G. Raposo et P. Chavrier, « Diaphanous-related formins are required for invadopodia formation and invasion of breast tumor cells », **Cancer research 69**, 2792-2800 (2009).

[44]B. Marty, V. Maire, E. Gravier, Guillem Rigaill, A. Vincent-Salomon, M. Kappler, I. Lebigot, F. Djelti, A. Tourdès, P. Gestraud, P. Hupé, E. Barillot, F. Cruzalegui, G. Tucker, M. Stern, J. Thiery, J. Hickman et T. Dubois, « Frequent PTEN genomic alterations and activated phosphatidylinositol 3-kinase pathway in basal-like breast cancer cells », **Breast Cancer Res 10**, R101 (2008).

[45]M. A. Bollet, N. Servant, P. Neuvial, C. Decraene, I. Lebigot, J.-P. Meyniel, Y. De Rycke, A. Savignoni, Guillem Rigaill, P. Hupé, A. Fourquet, B. Sigal-Zafrani, E. Barillot et J. Thiery, « High-resolution mapping of DNA breakpoints to define true recurrences among ipsilateral breast cancers », **Journal of the National Cancer Institute 100**, 48-58 (2008).

1.3 Pré-publications et articles en préparation

Pré-publications

[46]Guillem Rigaill, « Pruned dynamic programming for optimal multiple change-point detection », **arXiv preprint arXiv :1004.088** (2010).

En préparation

[47]V. Runge, T. D. Hocking, G. Romano, F. Afghah, P. Fearnhead et Guillem Rigaill, « gfpop : an R Package for Univariate Graph-Constrained Change-point Detection », **arXiv preprint arXiv :2002.03646** (2020).

[48]G. Romano, Guillem Rigaill, V. Runge et P. Fearnhead, « Detecting Abrupt Changes in the Presence of Local Fluctuations and Autocorrelated Noise », **arXiv preprint arXiv :2005.01379** (2020).

1.4 Logiciels

Packages R

[50]G. Rigaill, T. Hocking, R. Maidstone et P. Fearnhead, *Fpop*, `https://cran.r-project.org/web/packages/fpop/index.html`, R package for segmentation using optimal partionning and function pruning., 2018.

[51] P. Fearnhead et G. Rigaill, *robseg*, `https://github.com/guillemr/robust-fpop`, R package for change-point detection in the presence of outliers, 2017.

[52] J. Chiquet, V. Dervieux et G. Rigaill, *AriCode*, `https://CRAN.R-project.org/package=aricode`, aricode : a package for efficient computations ofstandard clustering comparison measures., 2018.

[53] A. Hulot, J. Chiquet et G. Rigaill, *MergeTree*, `https://CRAN.R-project.org/package=mergeTress`, mergeTrees : Aggregating Trees., 2018.

[54] P. Gutierrez, G. Rigaill et J. Chiquet, *Fused-Anova*, `https://r-forge.r-project.org/projects/fusedanova/`, This package adjusts a penalized ANOVA model with Fused-LASSO penality., 2013.

[55] T. Hocking et G. Rigaill, *PeakSegDP*, `https://github.com/tdhock/PeakSegDP`, Peak detection via Constrained Optimal Segmentation (Dynamic Programming algorithm)., 2014.

[56] A. Cleynen, E. Lebarbier, M. Koskas, G. Rigaill et S. Robin, *Segmentor3IsBack*, `https://CRAN.R-project.org/package=Segmentor3IsBack`, R package for the segmentation of RNAseq profiles, 2014.

[57] M. Pierre-Jean, P. Neuvial et G. Rigaill, *jointSeg*, `https://CRAN.R-project.org/package=jointSeg`, jointseg : Joint Segmentation of Multivariate (Copy Number) Signals, 2015.

[58] P. Picard, M. Hoebeke, E. Lebarbier, V. Miele, G. Rigaill et S. Robin, *cghseg*, `http://cran.r-project.org/web/packages/cghseg/index.html`, cghseg is an R package dedicated to the analysis of CGH profiles using segmentation models., 2011.

1.5 Présentations

Invitées

[59] *Changepoint detection in the presence of outliers and constraints*, GDR Stat et Santé, Conservatoire national des Arts et Métiers (CNAM), Paris, 2019.

[60] *A statistical method to detect abrupt changes in trees*, Workshop Change Point Detection : Limit Theorems, Algorithms, and Applications in Life Sciences, Alfried Krupp Wissenschaftskolleg, Greifswald, 2019.

[61] *Changepoint detection in the presence of outliers*, Statistical Scalability Workshop 2018, Windemere, UK, 2018.

[62] *On exact multiple changepoint algorithms. Optimal partitioning and functional pruning*, Time Dynamic Change Point Models and its Applications, Institut für Mathematische Stochastik der Universität Göttingen, 2014.

[63] *On exact multiple changepoint algorithms. Optimal partitioning and functional pruning*, Inference for Change-Point and Related Processes, Isaac Newton Institute. Cambridge, 2014.

[64] *Partitionnement optimal et élagage fonctionnel pour la détection de ruptures multiples.* 46èmes Journées de Statistique. Session "New challenges for new data", Rennes, 2014.

Acceptées

[65] *A fast homotopy algorithm for a large class of weighted classification problems*, Proceedings of the MLCB NIPS'14 workshop, Montreal, 2014.

[66] *A fast homotopy algorithm for a large class of weighted classification problems*, SMPGD : Statistical Methods for Post-Genomic Data workshop, Paris, 2014.

[67] *Segmentor3IsBack : an R package for the fast and exact segmentation of Seq-data*, The R User Conference, useR !, Albacete, 2013.

[68] *Exact posterior distributions and model selection criteria for multiple change-point detection problems*, Recent Advances in Changepoint Analysis, University of Warwick, 2012.

[69] *A statistical approach to estimate copy number from capture sequencing data*, STATSEQ, Verona, 2012.

[70] *An exact algorithm for estimating the read depth of NGS profiles using irregular histograms*, STATSEQ, Toulouse, 2011.

[71] *An Exact Algorithm for the Segmentation of NGS Profiles using Compression*, JOBIM : Journées Ouvertes en Biologie, Informatique et Mathématiques, Paris, 2011.

[72] *Exact and fast segmentation of large SNP/CGH profiles*, SMPGD : Statistical Methods for Post-Genomic Data workshop, Marseille, 2010.

[73] *DNA Copy number analysis. Defining a probability for the segmentation space*, SMPGD : Statistical Methods for Post-Genomic Data workshop, Paris, 2009.

[74] *GINCOS : a method to normalize Affymetrix GeneChip Human Mapping 50K Set*, JOBIM : Journées Ouvertes en Biologie, Informatique et Mathématiques, Marseille, 2007.

Chapitre 2

Un tour d'horizon

Qui traite de ce que l'on voit du donjon des Clarides.

Abeille - Anatole France

Résumé

Dans ce chapitre, je décris mes quatre principaux thèmes de recherche. Je détaille ensuite où, quand et avec qui j'ai conduit mes recherches. Je conclus par une liste des principaux projets scientifiques auxquels j'ai participé.

Avant-propos

Ce mémoire présente, de manière succincte, l'essentiel des travaux que j'ai réalisés durant ma thèse, mon post-doctorat, en tant que maître de conférence puis chargé de recherche. Je ne les reprends pas tous avec le même degré de détail. J'en détaille certains qui sont postérieurs à ma thèse.

2.1 Quatre thèmes de recherche

Mes recherches portent sur le développement et l'application de modèles et méthodes statistiques pour l'analyse de données omiques. Mes travaux se divisent en quatre axes ou thèmes que voici :

(1) la détection de ruptures multiples,

(2) l'analyse de données omiques,

(3) la classification régularisée,

(4) l'évaluation de méthodologies pour l'analyse de données omiques.

Les deux premiers thèmes sont plus importants. Ils font l'objet de près des trois quarts de mes publications. Je les présente de manière détaillée dans les chapitres 3 et 4. Je synthétise mes contributions aux deux autres thèmes, moins importants, dans les chapitres 5 et 6. Je présente succinctement ces quatre axes de recherche ci-dessous.

2.1.1 La détection de ruptures multiples

Il s'agit de détecter des changements abrupts dans des signaux ordonnés le long du temps, du génome... Ces changements ou ruptures délimitent des portions ou segments homogènes du signal. Une application emblématique est la détection d'altérations du nombre de copies d'ADN. Ces altérations sont impliquées dans le développement des cancers. La figure 2.1 donne un exemple de profil de nombre de copies d'ADN. On y observe visuellement des changements abrupts dans la moyenne. La détection de ces changements semble simple à notre œil humain. Mais il ne faut pas s'y tromper, c'est un problème difficile !

Pour le concevoir on peut compter le nombre de segmentations d'un profil avec n points. On peut placer une rupture après chaque point sauf le dernier. On a donc $n-1$ ruptures et donc 2^{n-1} segmentations envisageables. C'est beaucoup. Pour $n = 100$ cela donne déjà plus de 10^{29} segmentations. C'est la source de nombreux problèmes statistiques et algorithmiques et la difficulté croît si l'on modélise des dépendances. Dans bien des applications, au vu de la complexité des données, l'indépendance n'est pas une hypothèse satisfaisante ou acceptable. Elle conduit souvent à une mauvaise interprétation des données.

Dans le chapitre 3 je présente, de manière unifiée, un certain nombre d'algorithmes récents permettant de retrouver la segmentation de vraisemblance maximale parmi les 2^{n-1} possibles. Certains de ces algorithmes permettent de traiter des profils avec plus de 10^5 points en quelques minutes même pour des modèles intégrant des structures de dépendance riches.

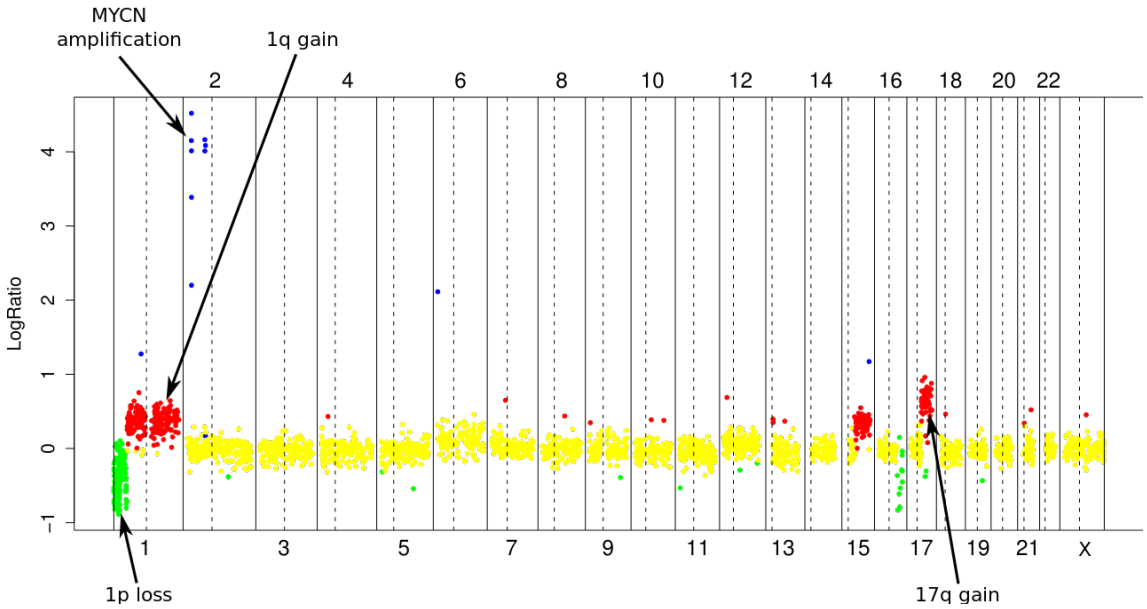

FIGURE 2.1 – Profil CGH (Comparative Genomic Hybridization) de la lignée cellulaire IMR32 [102]. Sans entrer dans trop de détails, chaque point ou sonde représente le log-ratio du nombre de copies d'ADN entre la lignée IMR2 et une lignée normale. Les sondes sont ordonnées en fonction de leurs coordonnées génomiques. On détecte visuellement des segments du génome où le log-ratio est anormalement grand (en rouge) ou petit (en vert). Ces segments correspondent à des amplifications ou délétions d'ADN chez IMR32. Certains, identifiés par des flèches, sont impliqués dans le développement du cancer.

2.1.2 L'analyse de données omiques

L'analyse d'un jeu de données omiques est souvent présentée comme une chose assez simple. De nombreux outils et packages sont déjà disponibles à cet effet. Il faut les exécuter dans un certain ordre pour obtenir les résultats. J'ai moi-même contribué à promouvoir cette vision du problème dans (i) des articles de biologie ou de bioinformatique et (ii) des articles méthodologiques.

(i) Dans mes articles de biologie ou de bioinformatique, la description de l'analyse est, presque toujours, succincte. Ne sont présentées que les principales étapes qu'il faut suivre pour reproduire les résultats.

(ii) Dans mes articles méthodologiques, les performances des méthodes sont souvent illustrées sur quelques applications omiques. Ces applications semblent immédiates. Il ne s'agit en fait que d'étapes très précises de l'analyse.

Ainsi, la justification des différentes étapes de l'analyse et de leur enchaînement est souvent omise. Or il faudrait argumenter le choix d'une approche statistique pour répondre à une question biologique

précise. Souvent les approches statistiques, les questions biologiques et les données sont complexes. Un dialogue entre deux ou trois disciplines s'impose. Je parlerai des analyses que j'ai réalisées et des difficultés que j'ai dû surmonter dans le chapitre 4.

2.1.3 Classification régularisée

Je me suis intéressé aux techniques de régression régularisée pour inférer des réseaux de gènes. Ces approches supposent souvent que les individus sont des réplicats indépendants et identiquement distribués. Sur le plan applicatif, il est souhaitable de prendre en compte ou d'identifier des structures de groupes lors de l'inférence du réseau. Cela m'a conduit à étudier des méthodes de régularisation pour l'analyse différentielle et la classification non-supervisée. Dans le chapitre 5, je présenterai succinctement mes contributions méthodologiques sur le sujet.

2.1.4 Évaluation de méthodologies omiques

Pour chaque étape d'une analyse omique des dizaines de méthodologies sont envisageables. Pour choisir judicieusement, il faut évaluer les performances pratiques de ces méthodologies. Ce n'est pas simple car :

— les articles décrivant ces méthodologies sont souvent optimistes,
— leurs validations expérimentales sont souvent de petite ampleur pour des raisons de coûts,
— leurs jeux de données simulées semblent peu crédibles biologiquement.

Je décrirai quelques travaux visant à rendre ces évaluations plus pertinentes dans le chapitre 6.

2.2 Où, quand et avec qui

Je fais ici un rapide historique de mes travaux.

2.2.1 De 2008 à 2010, doctorat

J'ai commencé mes recherches en master dans le groupe de bioinformatique et biostatistiques de l'Institut Curie. Sous la direction de Philippe Hupé et Emmanuel Barillot j'ai découvert l'art très difficile de normaliser des données [1]. J'ai continué par un doctorat sous la direction d'un biologiste moléculaire, Thierry Dubois, un bioinformaticien, Emmanuel Barillot et un statisticien, Stéphane Robin. Sous la direction de Thierry, j'ai analysé des jeux de données omiques [24, 25]. Sous la direction d'Emmanuel, j'ai développé et utilisé des outils bioinformatiques pour l'analyse du nombre de copies d'ADN [19, 26, 27]. Sous la direction de Stéphane, j'ai étudié des modèles de détection de ruptures multiples [2-6].

2.2.2 En 2011, post-doctorat

J'ai effectué un post-doctorat dans l'équipe de bioinformatique et biostatistiques du NKI-AVL, aux Pays-Bas. Sous la direction de Lodewyk Wessels et avec Ian Majewski et Jorma de Ronde j'ai étudié et développé des outils statistiques pour la bioinformatique [7, 20].

2.2.3 De 2012 à aujourd'hui, maître de conférences puis chargé de recherche

En 2012, j'ai obtenu un poste de maître de conférences à l'université d'Évry avec une chaire d'excellence INRA. Le poste était entre deux laboratoires :

1. l'équipe bioinformatique de l'URGV (Unité de Recherche en Génomique Végétale),
2. l'équipe Stat & Génome du LaMME (Laboratoire de Mathématiques et Modélisation d'Évry).

En 2017, j'ai réussi le concours de CR1 blanc de l'INRA pour travailler dans les mêmes équipes. En 2016, l'URGV a déménagé et a intégré l'IPS2 (Institut des Sciences des Plantes de Paris-Saclay) et l'équipe de

bioinformatique est devenue l'équipe Genomic Networks. À Évry, depuis 2012, j'ai élargi mes thématiques de recherche.

En 2012, j'ai débuté une collaboration avec Julien Chiquet sur les techniques de régression régularisée. L'objectif initial était de développer des méthodes pour l'inférence de réseaux de gènes en génomique. Pour prendre en compte des structures de groupe, nous avons développé des méthodes adaptées à la classification et l'analyse différentielle [8, 9, 43-45]. Dans le cadre de ce projet j'ai co-encadré la thèse de Trung Ha et je co-encadre la thèse de Martina Sundqvist.

Depuis 2012, je m'intéresse également à l'évaluation pratique de méthodes d'analyse en génomique. J'ai en particulier travaillé sur les outils pour l'analyse du nombre de copies d'ADN avec Toby Hocking, Pierre Neuvial et Morgane Pierre-Jean [10, 17, 18, 21] et l'analyse différentielle de données RNAseq avec Marie-Laure Martin-Magniette et Etienne Delannoy [11].

De 2012 à aujourd'hui j'ai continué à m'impliquer dans l'analyse de jeux de données omiques en collaboration avec des biologistes, bioinformaticiens et statisticiens, notamment Etienne Delannoy, Richard Berthomé, Eric Jencewksi et Thierry Dubois [22, 28-35].

Après mon recrutement, je ne pensais pas retravailler sur la détection de ruptures. À l'époque, je peinais à publier des travaux entrepris seul sur ce sujet à la fin de ma thèse [5, 41]. Toutefois, en 2014 suite à deux conférences j'ai initié deux collaborations sur le sujet. L'une était avec Robert Maidstone, Paul Fearnhead et Toby Hocking [12-15, 17] et l'autre avec Morgane Pierre-Jean, Guillemette Marrot et Alain Celisse de l'université de Lille [16]. En 2017, j'ai obtenu un financement de la Génopole pour travailler sur cette thématique (voir ci-dessous). J'ai ainsi pu recruter Vincent Runge comme post-doctorant entre 2017 et 2019 jusqu'à ce qu'il devienne maître de conférences à l'université d'Évry en 2019.

2.3 Activités liées à mes recherches

2.3.1 Projets structurants

Depuis 2012 je suis impliqué dans plusieurs projets de recherche et j'ai bénéficié de diverses sources de financement.

Chaire d'excellence 2012-2017. En 2012, j'ai obtenu une chaire d'excellence de l'INRA. Ce financement et la décharge d'enseignements associée ont été particulièrement favorables au développement de mes recherches.

ATIGE 2017-2020. En 2017 j'ai obtenu une bourse ATIGE de 250K euros pour le développement de modèles de détection de ruptures complexes pour étudier les régulations de l'ARN chloroplastique. Le financement m'a permis de recruter Vincent Runge comme post-doctorant.

Saturne 2018. En 2018, dans le cadre de mon projet ATIGE, j'ai obtenu un financement de 70K euros de la Génopole pour équiper le LaMME d'un nouveau serveur de calcul. Ce serveur est partagé avec un laboratoire du CNG (le Centre National de Génotypage).

BAP-Starter. En 2016, le département INRA BAP (Biologie et Amélioration des Plantes) a financé un projet, coordonné par Claire Lurin de l'IPS2, qui vise à identifier des sites de stérilité mâle cytoplasmique (CMS en anglais [103]) à partir de données de RNA-seq mitochondrial et chloroplastique. Depuis je suis impliqué dans l'analyse des données.

Sonata-Stat. De 2012 à 2015, j'ai participé au projet Sonata-Stat qui portait sur l'étude des gènes orphelins d'*Arabidopsis thaliana* impliqués dans la réponse aux stress biotiques et abiotiques.

2.3.2 Conférences et groupes de travail

Depuis 2012, je m'implique dans l'organisation de plusieurs groupes de travail.

Rencontres Paris-Lancaster sur la détection de ruptures. En 2015, 2017 et 2019 j'ai co-organisé avec Idris Eckley, Paul Fearnhead et Stéphane Robin, des journées sur la détection de ruptures. Ces journées ont regroupé à chaque fois une vingtaine de statisticiens de Lancaster et Paris. L'édition de 2019 s'est tenue à l'Institut des Systèmes Complexes à Paris. Elle a été suivie de deux jours de conférences sur la détection de ruptures. Près de 40 chercheurs y ont participé. La majorité venait de France ou d'Angleterre, mais des chercheurs sont aussi venus de Belgique, de Suisse, d'Allemagne et des États-Unis.

Réseau méthodologique Netbio. Depuis 2012 je participe au réseau méthodologique NETBIO [90]. Depuis 2015 je suis également co-organisateur de ce groupe avec Marie-Laure Martin-Magniette, Julien Chiquet, Etienne Delannoy, Françoise Monéger et Nathalie Vialaneix. La vocation de NETBIO est de créer un espace où échanger sur les méthodes statistiques et bioinformatiques pour la construction et l'analyse de graphes et discuter de l'adéquation de la modélisation par rapport aux problématiques biologiques. Il y a 228 inscrits à ce réseau et une quarantaine participe aux réunions annuelles.

Participation à l'édition d'un livre pour le GDR Bim. Depuis 2018, je participe, avec Marie-Laure Martin-Magniette à l'édition de quatre chapitres relatifs aux aspects statistiques de l'intégration de données pour un livre commandité par le GdR Bim [69].

2.3.3 Enseignements

Écoles d'été. J'ai enseigné dans plusieurs écoles d'été autour de l'utilisation de méthodes statistiques pour l'analyse de données omiques :

- en 2016 à l'école « From gene expression to network » organisée par Marie-Laure Martin Magniette et Etienne Delannoy [97],
- en 2017, 2018 et 2019 à l'école d'été de la Génopole en bioinformatique et biostatistiques [70],
- en 2018 à l'école d'été (JC)2BIM organisée par le GDR Bim [69].

Ces écoles d'été s'adressaient essentiellement à des biologistes et bioinformaticiens ayant quelques connaissances en biostatistiques.

Module Biologie des Systèmes 1. Je suis responsable d'un module intitulé Biologie des Systèmes 1 pour le master 1 Biologie et Santé de l'université Paris-Saclay. J'ai fait évolué le module avec les années. En 2014, le module faisait un panorama, non-exhaustif, de techniques omiques pour analyser la cellule. Aujourd'hui, une part importante du module est consacrée à l'analyse des données et à l'importance de cette analyse pour l'interprétation des données.

Enseignements en statistiques. De 2012 à 2018 j'ai enseigné les statistiques à des étudiants en biologie, bioinformatique et statistiques de l'université d'Évry.

Chapitre 3

Détection de ruptures multiples

Rien n'a changé. J'ai tout revu : l'humble tonnelle
De vigne folle avec les chaises de rotin...
Le jet d'eau fait toujours son murmure argentin
Et le vieux tremble sa plainte sempiternelle.

Poèmes saturniens - Paul Verlaine

Résumé

Les modèles de détection de ruptures multiples sont souvent utilisés en génomique. Avec un certain nombre de collaborateurs, j'ai proposé une classe d'algorithmes de programmation dynamique pour maximiser une vraisemblance pénalisée. Ces algorithmes exploitent une récurrence sur le paramètre continu du dernier segment. Cette récurrence est proche de celle de l'algorithme de Viterbi, utilisée pour les chaînes de Markov cachées. Dans ce chapitre, je présente cette récurrence pour l'inférence de modèles gaussiens. J'explique ensuite comment l'étendre pour inférer des modèles avec des dépendances entre les segments successifs. Mon objectif est de donner une intuition précise du mode opératoire de ces algorithmes. Je présenterai les formules de récurrence qu'ils exploitent sans les justifier ici, ni rentrer dans les détails de leur implémentation. Je terminerai par quelques perspectives.

3.1 La détection de ruptures multiples

La détection de ruptures multiples est un domaine de recherche extrêmement vaste. C'est un problème fréquent dans beaucoup de domaines d'application. En génomique il semble naturel de visualiser les données le long du génome pour identifier des régions au comportement homogène. La détection d'altérations du nombre de copies d'ADN [92] et la détection de domaines compositionnels [104] sont des exemples. Je ne ferai pas une revue de ce domaine en pleine évolution. À ce titre, je recommande la lecture de l'HDR d'Émilie Lebarbier [84] ou de la revue [98].

Dans ce chapitre, je présente des algorithmes maximisant exactement une vraisemblance pénalisée. Cela restreint grandement le sujet. La majorité des méthodologies existantes ne maximise pas une vraisemblance. Je liste ci-dessous quelques-unes des nombreuses stratégies alternatives :

— la segmentation binaire et ses variantes comme la « Wild Binary Segmentation » [66],
— des approches multi-échelles comme SMUCE [65],
— des approches basées sur la variation totale [72].

3.1.1 Intérêt du maximum de vraisemblance pénalisée

Maximiser une vraisemblance pénalisée est une approche digne d'intérêt pour la détection de ruptures multiples. Je liste ci-après quelques arguments pour en débattre :

1. C'est une démarche classique en statistique. Elle a fait ses preuves pour beaucoup d'autres modèles.
2. C'est une démarche assez naturelle et simple conceptuellement.
3. Statistiquement, les garanties sur l'estimation du signal ou la détection des ruptures sont bonnes asymptotiquement [58, 108] et non-asymptotiquement [47, 68, 85].
4. Pour des modèles simples au moins, les temps de calcul sont très compétitifs [4, 12].
5. Sur des simulations les performances sont souvent bonnes. Par exemple, nous avons montré dans [12], sur les simulations proposées par [66], que la pénalité de [108], proposée il y a déjà 40 ans, garde encore tout son intérêt devant des approches récentes comme WBS ou SMUCE [65, 66].
6. Sur certaines applications les résultats sont souvent très satisfaisants. Ils sont même l'état de l'art dans plusieurs applications génomiques [18, 60, 74, 83].

3.1.2 Mes travaux sur la détection de ruptures

Mes travaux sur la détection de ruptures sont de trois natures différentes :

1. Dans un cadre Bayésien, j'ai proposé des algorithmes [3, 6] pour calculer l'entropie de l'espace des segmentations et un critère ICL (Integrated Completed Likelihood en anglais [54]). J'ai réalisé ces travaux durant ma thèse avec Stéphane Robin, Emilie Lebarbier, Alice Cleynen, The Minh Luong et Grégory Nuel. Je n'en parlerai pas ici.
2. J'ai développé des algorithmes exacts et rapides pour maximiser une vraisemblance pénalisée [4, 5, 12-16, 42]. J'ai débuté ces travaux seul à la fin de ma thèse [41]. J'ai ensuite collaboré avec beaucoup de chercheurs et chercheuses sur le sujet. Je parlerai de ces travaux dans ce chapitre.
3. J'ai proposé des modèles et méthodologies de segmentation pour des applications en génomique [1, 2, 7, 17]. J'évoquerai rapidement certains de ces travaux dans ce chapitre.

3.1.3 Rapide tour d'horizon du chapitre

L'idée principale des algorithmes que j'ai proposés avec mes collaborateurs pour maximiser une vraisemblance pénalisée est de construire une récurrence sur la moyenne du dernier segment. Je parlerai de fonctionnalisation ou d'algorithme fonctionnel [12].

Voilà un plan de ce chapitre :

— Dans la section 3.2 je présenterai le modèle de détection de ruptures multiples et le problème de vraisemblance pénalisée associé.
— Dans la section 3.3 je décrirai des algorithmes de programmation dynamique pour inférer le modèle, en commençant par les plus classiques, utilisant une récurrence sur la dernière rupture (3.3.1) et en terminant par les fonctionnels que j'ai développés (3.3.2).
— Dans la section 3.4 je parlerai des algorithmes de programmation dynamique fonctionnelle que j'ai développés pour prendre en compte des structures de dépendances.
— Dans la section 3.5 je conclurai avec quelques perspectives.

3.2 Modèle et vraisemblance pénalisée

3.2.1 Le modèle

Pour parler de vraisemblance il faut définir un modèle. On considère des données $Y_1, Y_2, ..., Y_n$ et $K-1$ ruptures $\tau_1 < \cdots < \tau_{K-1}$ entre 0 et n. Par convention on définit aussi $\tau_0 = 0$ et $\tau_K = n$. Toutes ces ruptures définissent K segments $\rrbracket \tau_k, \tau_{k+1} \rrbracket = \{\tau_k + 1, \cdots, \tau_{k+1}\}$. Dans chaque segment on suppose que les Y_i sont indépendants et de loi gaussienne. Je présente schématiquement ce modèle sur la figure 3.1. Mathématiquement il s'écrit :

$$\forall\, i \ \in \rrbracket \tau_k, \tau_{k+1} \rrbracket \quad Y_i \sim \mathcal{N}(\theta_k, \sigma^2) \quad i.i.d.$$

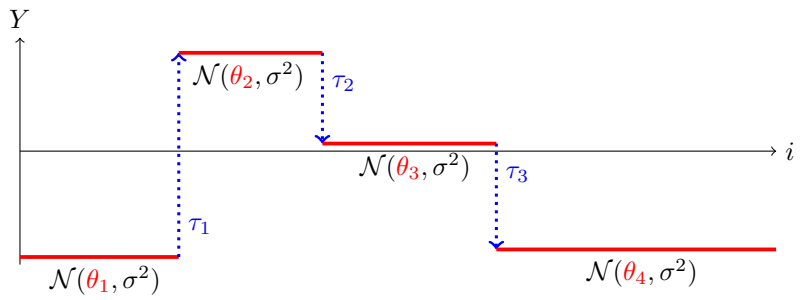

FIGURE 3.1 – Présentation schématique du modèle de segmentation gaussien avec ses paramètres. Le k-ième segment est délimité par deux ruptures τ_{k-1} et τ_k. Dans ce segment les données Y_i sont indépendantes et de même loi gaussienne de moyenne θ_k et de variance σ^2.

Ce modèle gaussien, inféré par maximum de vraisemblance pénalisée, est l'état de l'art pour l'analyse du nombre de copies d'ADN [18, 83, 92]. Dans ce cas, les Y_i correspondent aux mesures le long du génome du log-ratio du nombre de copies d'ADN entre deux types cellulaires. Un exemple de profil de nombre de copies d'ADN est donné figure 2.1 à la page 14.

Évidemment, il est préférable de choisir un modèle adapté à la nature des données. Par exemple, pour des données de RNA-seq on préférera souvent un modèle de Poisson ou négatif binomial [5, 61]. Les récurrences et algorithmes que je vais présenter fonctionnent aussi pour ces modèles. L'implémentation est légèrement plus difficile que dans le cas gaussien. Je n'en parlerai pas d'avantage dans ce chapitre.

3.2.2 Vraisemblance pénalisée

Si le nombre de segments (K) est connu, la log-vraisemblance du modèle gaussien s'écrit :

$$-\frac{1}{2\sigma^2}\sum_{k=1}^{K}\sum_{i=\tau_{k-1}+1}^{\tau_k}(y_i-\theta_k)^2 \quad - \quad \frac{n}{2}\log(2\pi\sigma^2). \tag{3.1}$$

En dérivant par rapport au paramètre σ^2, on obtient que pour maximiser la vraisemblance il convient de minimiser la quantité $C_{K,n}$ suivante [48, 53, 64] :

$$C_{K,n} = \min_{\substack{\tau_1,\dots\tau_{K-1}\\ \theta_1,\dots\theta_K}} \left\{ \sum_{k=1}^{K}\sum_{i=\tau_{k-1}+1}^{\tau_k}(y_i-\theta_k)^2 \right\} \tag{3.2}$$

$$= \min_{\tau_1,\dots\tau_{K-1}} \left\{ \sum_{k=1}^{K}\sum_{i=\tau_{k-1}+1}^{\tau_k}(y_i-\bar{y}_{\tau_{k-1}+1:\tau_k})^2 \right\}, \tag{3.3}$$

où $\bar{y}_{\tau_{k-1}+1:\tau_k}$ est la moyenne empirique du segment $]\!]\tau_{k-1},\tau_k]\!]$:

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

Le nombre de segments, K, n'est généralement pas connu. Sans pénalité la plus petite valeur de $C_{K,n}$ est toujours obtenue pour $K=n$ segments et vaut 0. Pour obtenir un nombre de segments plus parcimonieux il est classique de pénaliser la vraisemblance. De nombreuses pénalités ont été proposées et étudiées [47, 51, 55, 68, 85, 108]. La pénalité est une fonction croissante du nombre de ruptures. En général, elle dépend aussi de n et de σ^2. Je l'écrirai $pen(K,n,\sigma^2)$.

L'une des plus simples et plus anciennes est celle proposée dans [108]. Elle est linéaire en K et vaut $pen(K,n,\sigma^2) = 2K\sigma^2\log(n)$. Celle proposée dans [85] a de meilleures propriétés statistiques. Elle est toutefois un peu plus complexe et vaut $pen(K,n,\sigma^2) = K\sigma^2(5+2\log(n/K))$.

3.2.3 Pénalités linéaires et non-linéaires

Pour maximiser la vraisemblance on exploite des formules de récurrence que je décris dans la section 3.3. En algorithmique, on parle de programmation dynamique. En fonction de la nature exacte de la pénalité, le nombre de récurrences à mettre en œuvre est plus ou moins grand. Le cas le plus simple est une pénalité linéaire en K. Les pénalités non-linéaires sont plus complexes à traiter.

3.2.3.1 Minimisation avec une pénalité linéaire

La très populaire pénalité $pen(K, n, \sigma^2) = 2K\sigma^2 \log(n)$ proposée par [108] est linéaire en K. Pour un profil donné, de taille n et de variance σ^2, elle s'écrit comme une constante λ fois K : λK. Pour optimiser la vraisemblance ainsi pénalisée on peut résoudre par programmation dynamique le problème suivant [12, 77, 82] :

$$P_{\lambda,n} = \min_{\substack{K \\ \tau_1,\dots\tau_{K-1} \\ \theta_1,\dots\theta_K}} \left\{ \sum_{k=1}^{K} \left[\sum_{i=\tau_{k-1}+1}^{\tau_k} (y_i - \theta_k)^2 \right] + \lambda K \right\} \tag{3.4}$$

$$= \min_{\substack{K \\ \tau_1,\dots\tau_{K-1}}} \left\{ \sum_{k=1}^{K} \left[\sum_{i=\tau_{k-1}+1}^{\tau_k} (y_i - \bar{y}_{\tau_{k-1}+1:\tau_k})^2 \right] + \lambda K \right\}. \tag{3.5}$$

3.2.3.2 Minimisation avec une pénalité quelconque

Pour des pénalités non-linéaires comme celles de [85] le problème est légèrement plus complexe. Il faut procéder en deux temps. Deux stratégies sont envisageables.

Première stratégie, classique. La stratégie la plus classique [85] procède de la manière suivante :

1. L'utilisateur donne un nombre de ruptures K_{max} supérieur au nombre de ruptures attendu.
2. On calcule par programmation dynamique tous les $C_{K,n}$ pour K dans $\{1, \cdots, K_{max}\}$.
3. Connaissant les $C_{K,n}$, on trouve le minimum de $C_{K,n} + pen(K, n, \sigma^2)$.

Deuxième stratégie, plus récente. Revenons un instant à la pénalité linéaire. Si l'on résout le problème $P_{\lambda,n}$ de l'équation (3.4) on obtient une segmentation avec $K(\lambda)$ segments. On démontre simplement que cette segmentation n'est autre que la segmentation optimale en $K(\lambda)$ segments : $C_{K(\lambda),n}$. La fonction $K(\lambda)$ est décroissante et constante par morceaux. Sur un intervalle $[\lambda_{min}, \lambda_{max}]$ donné la stratégie CROPS [73] permet de reconstruire efficacement $K(\lambda)$. Pour fonctionner, CROPS a seulement besoin d'une routine calculant $P_{\lambda,n}$ comme PELT [82] ou FPOP [12] que je décris plus loin.

La deuxième stratégie pour des pénalités non-linéaires exploite CROPS :

1. L'utilisateur donne un intervalle $[\lambda_{min}, \lambda_{max}]$.
2. On récupère les $C_{K(\lambda),n}$ pour λ dans $[\lambda_{min}, \lambda_{max}]$ avec CROPS.
3. Connaissant les $C_{K(\lambda),n}$, on trouve le minimum de $C_{K(\lambda),n} + pen(K(\lambda), n, \sigma^2)$.

Cette deuxième stratégie est particulièrement intéressante quand le nombre de ruptures attendu est grand. En effet pour un intervalle $[\lambda_{min}, \lambda_{max}]$ suffisamment bien choisi on n'explorera pas des segmentations avec peu de segments. La première stratégie, elle, doit parcourir tous les nombres de ruptures de 1 à K_{max}. C'est très coûteux en temps. D'autant que les modèles avec peu de ruptures ne sont pas utiles au final.

3.3 Un peu de programmation dynamique

Dans cette section j'explique comment calculer par programmation dynamique $C_{K,n}$ et $P_{\lambda,n}$ définis aux équations (3.2) et (3.4). Dans tous les cas, le calcul repose sur une récurrence. Deux types de récurrences existent dans la littérature. La première, classique, considère toutes les positions possibles de la dernière rupture t. La seconde considère toutes les moyennes possibles du dernier segment μ. Dans la suite, je présente cette seconde récurrence comme une version continue de l'algorithme de Viterbi utilisé pour les chaînes de Markov cachées (HMM en anglais). Cette présentation est assez différente des papiers la décrivant [4, 12, 41]. Je pense qu'elle est mathématiquement plus intuitive. Elle donne une idée plus précise de ce que manipule l'algorithme.

3.3.1 Une récurrence sur la dernière rupture

3.3.1.1 La récurrence

Dans $C_{K,n}$ et $P_{\lambda,n}$ définis aux équations (3.2) et (3.4) à la page 21 le terme $\sum_{i=\tau_{k-1}+1}^{\tau_k}(y_i-\theta_k)^2$ ne dépend que des observations y_i du segment $]\!]\tau_{k-1},\tau_k]\!]$. On dit que $C_{K,n}$ et $P_{\lambda,n}$ sont additives en les segments. De ce fait, on peut appliquer le principe de Bellman [53, 64]. Plus simplement, si nous connaissions la dernière rupture t pour calculer $C_{K,n}$ et $P_{\lambda,n}$ il nous suffirait de :

1. calculer le coût optimal du dernier segment $]\!]t,n]\!]$: $c_{t+1:n}=\sum_{i=t+1}^{n}(y_i-\bar{y}_{t+1:n})^2$;
2. calculer le coût de la meilleure segmentation jusqu'à t : $C_{K-1,t}$ ou $P_{\lambda,t}$.

Ne connaissant pas la dernière rupture nous considérons toutes les positions possibles et prenons la meilleure. Mathématiquement, on obtient les récurrences suivantes :

$$C_{K,n} = \min_{t<n}\{C_{K-1,t} + c_{t+1:n}\}, \tag{3.6}$$

$$P_{\lambda,n} = \min_{t<n}\{P_{\lambda,t} + c_{t+1:n}\} + \lambda. \tag{3.7}$$

3.3.1.2 Quelques détails sur l'implémentation

Boucles Sur le plan algorithmique, pour obtenir $C_{K,n}$ il faudra exécuter la récurrence (3.6) pour tous les $C_{k,t}$ avec $k \leq K$ et $t \leq n$. On le fera à l'aide d'une double boucle sur k et t. Pareillement, pour obtenir $P_{\lambda,n}$ il faudra exécuter la récurrence (3.7) pour tous les $P_{\lambda,t}$ avec $t \leq n$. Une simple boucle sur t suffira.

Calculs préliminaires Pour implémenter efficacement les récurrences (3.6) et (3.7) il faut calculer à la volée le coût de n'importe quel segment $c_{t_1:t_2}$. Pour cela, définissons la somme des observations à la puissance j entre t_1 et t_2 :

$$s^{(j)}_{t_1:t_2} = \sum_{i=t_1}^{t_2} y_i^j.$$

La somme entre t_1 et t_2 s'obtient facilement à partir des sommes entre 1 et t_1 et 1 et t_2 :

$$s^{(j)}_{t_1:t_2} = s^{(j)}_{1:t_2} - s^{(j)}_{1:(t_1-1)}. \tag{3.8}$$

On peut ré-écrire le coût comme :

$$c_{t_1:t_2} = s^{(2)}_{t_1:t_2} - \frac{[s^{(1)}_{t_1:t_2}]^2}{t_2-t_1-1}. \tag{3.9}$$

$s^{(1)}_{1:t}$ et $s^{(2)}_{1:t}$ sont des sommes cumulées. On peut calculer toutes les $s^{(1)}_{1:t}$ et $s^{(2)}_{1:t}$ pour t inférieur à n avec une complexité de $O(n)$ en temps et en espace. Moyennant ce calcul préliminaire, en combinant (3.8) et (3.9) on calcule tout $c_{t_1:t_2}$ à la volée en $O(1)$.

Complexité. Moyennant un calcul efficace des $c_{t_1:t_2}$, on obtient des complexités de :
— $O(K_{max}n^2)$ en temps et $O(K_{max}n)$ en espace pour calculer tous les $C_{K,n}$ avec K dans $\{1, ..., K_{max}\}$.
— $O(n^2)$ en temps et $O(n)$ en espace pour calculer $P_{\lambda,n}$.

Des astuces similaires permettent d'obtenir des algorithmes de complexité identique pour des modèles multi-paramétriques [48]. Avec quelques co-auteurs nous avons étendu ce résultat à des modèles non-paramétriques basés sur des noyaux. Ces derniers modèles permettent notamment de détecter des changements dans la distribution du signal et pas seulement dans un paramètre ou un moment de la distribution [47].

3.3.1.3 Élagage des dernières ruptures

À chaque étape de la récurrence (3.7) il faut parcourir toutes les ruptures t possibles avant n. Imaginons qu'il y ait une vraie rupture dans le signal à la position t_1. Dans ce cas, les segmentations avec une dernière rupture avant t_1 sont probablement associées à une vraisemblance faible. Parcourir toutes ces ruptures avant t_1 dans la récurrence (3.7) semble une perte de temps.

Les auteurs de [82] ont démontré que cette intuition est correcte. À chaque étape de la récurrence on peut, sans faire d'erreur, éliminer définitivement toutes les ruptures t telles que :

$$P_{\lambda,t} + c_{t+1:n} > P_{\lambda,n}. \tag{3.10}$$

L'algorithme PELT [82] implémente cette idée d'élagage. Il utilise deux récurrences, la première sur $P_{\lambda,n}$ et la seconde sur l'ensemble des ruptures à considérer $R_{\lambda,n}$:

$$P_{\lambda,n} = \min_{t \in R_{\lambda,n-1}} \{P_{\lambda,t} + c_{t+1:n}\} + \lambda \tag{3.11}$$

$$R_{\lambda,n} = \{t \in R_{\lambda,n-1} | P_{\lambda,t} + c_{t+1:n} \leq P_{\lambda,n}\} \cup \{n\}. \tag{3.12}$$

L'ensemble $R_{\lambda,n-1}$ est inclus dans $\{1, \cdots n\}$. Si le nombre de vraies ruptures croît linéairement avec n, l'espérance du temps de calcul de PELT est $O(n)$ [82] ; intuitivement, l'espérance du cardinal de $R_{\lambda,n-1}$ est bornée par une constante. S'il n'y a pas ou peu de ruptures la complexité reste quadratique.

Un élagage générique ! L'élagage à la mode PELT est qualifié de « basé sur des inégalités » dans [12]. Il est particulièrement générique. Il dépend assez peu de la forme du coût des segments. Informellement, cet élagage est juste si découper un segment en deux est toujours bénéfique en termes de coût. Formellement, il suffit que pour toutes positions $t_1 \leq t_2 \leq t_3$ et une certaine constante κ :

$$c_{t_1:t_2} + c_{t_2+1:t_3} + \kappa \leq c_{t_1:t_3}. \tag{3.13}$$

Cette propriété est vraie pour un grand nombre de coûts [82]. Par exemple elle est vraie, avec $\kappa = 0$, si l'on considère l'estimation par maximum de vraisemblance pénalisée d'un modèle paramétrique.

3.3.2 Une récurrence sur le paramètre du dernier segment

Je passe maintenant au deuxième type de récurrences pour calculer $C_{K,n}$ et $P_{\lambda,n}$ définis aux équations (3.2) et (3.4). Ces récurrences considèrent la moyenne du dernier segment μ plutôt que la position de la dernière rupture t.

3.3.2.1 Fonctionnalisation

L'idée est de considérer $C_{K,n}$ ou $P_{\lambda,n}$ en fonction du paramètre du dernier segment, μ. Cette idée est qualifiée « d'élagage fonctionnel » dans [12]. Formellement, on définit les deux fonctions de μ, $\widetilde{C}_{K,n}(\mu)$ et

$\widetilde{P}_{\lambda,n}(\mu)$, comme suit :

$$\widetilde{C}_{K,n}(\mu) = \min_{\substack{\tau_1,\dots,\tau_{K-1} \\ \theta_1,\dots,\theta_K=\mu}} \left\{ \sum_{k=1}^{K} \sum_{i=\tau_{k-1}+1}^{\tau_k} (y_i - \theta_k)^2 \right\} \tag{3.14}$$

et

$$\widetilde{P}_{\lambda,n}(\mu) = \min_{\substack{K,\tau_1,\dots\tau_{K-1} \\ \theta_1,\dots\theta_K=\mu}} \left\{ \sum_{k=1}^{K} \sum_{i=\tau_{k-1}+1}^{\tau_k} (y_i - \theta_k)^2 + \lambda K \right\}. \tag{3.15}$$

$\widetilde{C}_{K,n}(\mu)$ et $\widetilde{P}_{\lambda,n}(\mu)$ s'interprètent comme le coût optimal sachant la moyenne μ du dernier segment. On retrouve $C_{K,n}$ et $P_{\lambda,n}$ en minimisant $\widetilde{C}_{K,n}(\mu)$ et $\widetilde{P}_{\lambda,n}(\mu)$ en μ.

Polynômes par morceaux. $\widetilde{P}_{\lambda,n}(\mu)$ et $\widetilde{C}_{K,n}(\mu)$ sont polynomiales par morceaux ou intervalles. En effet, par définition, elles sont le minimum d'un nombre fini de fonctions polynomiales. Chaque intervalle est associé à une dernière rupture permettant d'atteindre ce coût optimal. La figure 3.2, reprise de [12] donne un exemple de $\widetilde{C}_{K,n}(\mu)$.

Cette fonctionnalisation semble complexifier le problème. Dans une certaine mesure c'est le cas ; il est algorithmiquement plus difficile de manipuler une fonction qu'une valeur dans $\mathbb{R}$. Toutefois dans beaucoup de situations le nombre de fonctions à manipuler pour représenter $\widetilde{C}_{K,n}$ ou $\widetilde{P}_{\lambda,n}$ est faible. Par exemple, sur la figure 3.2 il y a besoin de seulement 5 intervalles pour décrire parfaitement $\widetilde{C}_{2,43}(\mu)$. C'est nettement moins que les $85 = 43 \times 2 - 1$ nécessaires dans le pire des cas (voir le paragraphe complexité page 28).

3.3.2.2 Un rapide historique

Une idée récurrente. À ma connaissance cette idée de fonctionnaliser $C_{K,n}$ ou $P_{\lambda,n}$ est apparue dans une pré-publication arxiv d'avril 2010 dont je suis l'auteur [41] et dans la thèse de Nicholas Johnson [80] datée d'octobre 2010. Des idées proches ont également été proposées pour la régression isotonique par Günter Rote en 2012 [94, 95].

Fonctionnalisation pour des problèmes de fused-lasso. Dans sa thèse [80], Johnson présente cette idée de fonctionnalisation pour divers problèmes de fused-lasso. Le fused-lasso peut être vu comme une relaxation convexe du problème $\widetilde{P}_{\lambda,n}(\mu)$ et s'écrit formellement :

$$\tilde{F}_{\lambda,n}(\mu) = \min_{\theta_1,\dots\theta_n=\mu} \left\{ \sum_{k=1}^{K} \sum_{i=1}^{n} (y_i - \theta_i)^2 + \lambda \sum_{i=1}^{n-1} |\theta_i - \theta_{i+1}| \right\}.$$

On retombe sur le problème $\widetilde{P}_{\lambda,n}(\mu)$ en remplaçant $|\theta_i - \theta_{i+1}|$ par la fonction indicatrice $\mathbb{I}_{\theta_i \neq \theta_{i+1}}$.

Fonctionnalisation pour des problèmes de maximum de vraisemblance. Johnson a obtenu une complexité linéaire pour le problème de fused-lasso $F_{\lambda,n}(\mu)$ [80]. Il décrit rapidement un algorithme pour $\widetilde{P}_{\lambda,n}(\mu)$ mais sans garantie sur sa complexité. Cet algorithme est à peu de choses près l'algorithme FPOP décrit dans [12]. Dans [4] et [41] je décris un algorithme pour obtenir $\widetilde{C}_{K,n}(\mu)$ avec une complexité au pire $O(Kn^2)$ en temps et $O(Kn)$ en espace. Moyennant quelques ré-écritures la même preuve donne une complexité de $O(n^2)$ en temps et de $O(n)$ en espace pour l'algorithme FPOP pour le calcul de $\widetilde{P}_{\lambda,n}(\mu)$.

J'avoue n'avoir découvert les travaux de Johnson et Rote qu'assez tardivement, durant l'écriture de [12]. Je me rassure en constatant que les travaux publiés de Johnson et Rote [81, 95] ne citent ni les travaux de Rebecca Killick [82] ni les miens [41]. Faire la bibliographie sur un sujet n'est pas simple !

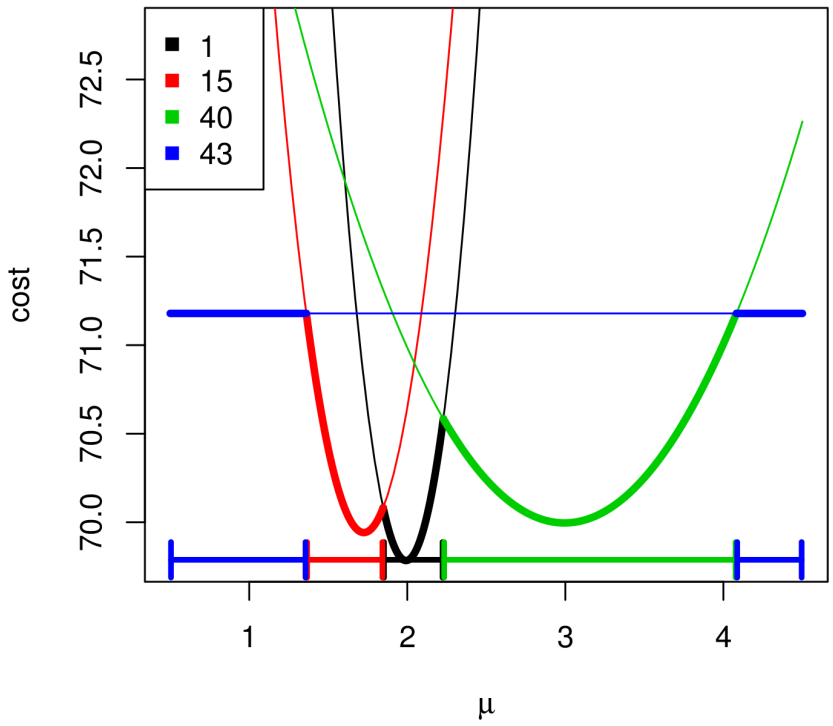

FIGURE 3.2 – Un exemple de fonction $\widetilde{C}_{K,n}(\mu)$. Exemple tiré de [12] pour un profil de 43 points. $\widetilde{C}_{2,43}(\mu)$ est le coût optimal en K segments sachant que la moyenne du dernier segment est μ (voir (3.14)). $\widetilde{C}_{2,43}(\mu)$ est un polynôme du second degré par morceaux ou intervalles. $\widetilde{C}_{2,43}(\mu)$ est représentée par une ligne épaisse qui change de couleur à chaque intervalle. Les lignes fines de couleur donnent la valeur des polynômes au delà de leurs intervalles d'optimalité respectifs. On obtient ici 5 intervalles. Chacun correspond à une rupture. Les deux intervalles bleus correspondent à une dernière rupture en 43, l'intervalle rouge à une dernière rupture en 15, l'intervalle noir à une dernière rupture en 1 et l'intervalle vert à une dernière rupture en 40. Les autres ruptures ne sont jamais optimales : pour tout μ leurs coûts sont supérieurs à $\widetilde{C}_{2,43}(\mu)$. Ainsi on peut lire sur ce graphe que pour une moyenne de 3, la meilleure segmentation a une dernière rupture en 40.

3.3.2.3 Ruptures et chaîne de Markov cachée

Avant d'en venir à la récurrence sur $\widetilde{P}_{\lambda,n}(\mu)$, il est intéressant de ré-écrire le modèle comme une chaîne de Markov cachée ou HMM (Hidden Markov Model en anglais). Le modèle est décrit ci-dessous et sa représentation graphique est sur la figure 3.3.

— Les variables cachées $Z_1, ..., Z_n$ sont continues dans $[a, b]$ avec a et b dans $\mathbb{R}$ et $a \leq b$.
— Les variables observées $Y_1, ..., Y_n$ sont, comme dans la description en 3.2.1, dans $\mathbb{R}$.
— La règle de chaînage entre Z_i et Y_i est $(Y_i|Z_i = \mu) \sim \mathcal{N}(\mu, \sigma^2)$.
— Le noyau de transition entre Z_i et Z_{i+1} est $k(x, y) \propto \mathbb{I}_{x=y} + e^{-\lambda}\mathbb{I}_{x\neq y}$. La constante de normalisation est omise par simplicité. Le terme $e^{-\lambda}\mathbb{I}_{x\neq y}$ modélise la probabilité d'une rupture et le terme $\mathbb{I}_{x=y}$ l'absence de rupture.

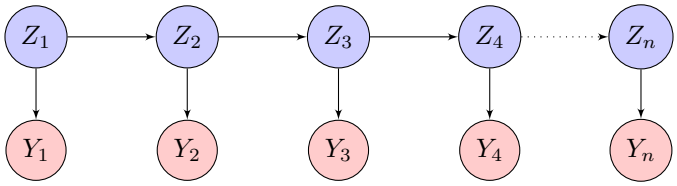

FIGURE 3.3 – Réseau bayésien du modèle de détection de ruptures multiples vu comme une chaîne de Markov cachée à espace d'états continu. Les variables cachées Z_i sont dans $[a, b]$. Les variables observées Y_i dans $\mathbb{R}$.

La log-vraisemblance du modèle s'écrit :

$$-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - z_i)^2 \; - \sum_{i=1}^{n-1} e^{-\lambda}\mathbf{I}_{z_i \neq z_{i+1}}. \tag{3.16}$$

À une constante près et en changeant de variables on retrouve la log-vraisemblance de l'équation (3.1) page 21 du modèle de ruptures multiples. Pour le changement de variable il faut :

— définir K comme $\sum_{i=1}^{n-1} \mathbb{I}_{z_i \neq z_{i+1}}$,
— définir les ruptures τ_k comme les K positions i telles que $z_i \neq z_{i+1}$,
— définir les segments et θ_k à partir des ruptures,
— pour tous les i dans $[\![\tau_{k-1}, \tau_k]\!]$ remplacer z_i par θ_k.

Dans 3.3.2.4 je décris la récurrence de l'algorithme de Viterbi pour ce modèle HMM. Dans 3.3.2.5 j'explique comment l'appliquer même si l'espace d'états est continu.

3.3.2.4 Récurrence fonctionnelle sur le paramètre de moyenne

Décrivons maintenant la récurrence pour mettre à jour $\widetilde{P}_{\lambda,n}(\mu)$. À l'instar de l'algorithme de Viterbi deux événements peuvent se produire :

1. Soit au temps $n-1$ nous étions déjà dans l'état μ. Dans ce cas il faut considérer la probabilité de rester dans l'état μ au temps n. Elle est donnée par le noyau de transition.
2. Soit au temps $n-1$ nous n'étions pas dans l'état μ. Dans ce cas il y a une rupture. Il faut sauter du meilleur état μ' au temps $n-1$ puis considérer la probabilité de changer d'état au temps n.

Dans les deux cas, il reste à ajouter la log-vraisemblance associée à l'observation y_n, c'est-à-dire $(y_n - \mu)^2$. Formellement, on obtient la récurrence suivante [12] :

$$\widetilde{P}_{\lambda,n}(\mu) = \min\{\; \widetilde{P}_{\lambda,n-1}(\mu),\; P_{\lambda,n-1} + \lambda \;\} \; + \; (y_n - \mu)^2. \tag{3.17}$$

De manière analogue on obtient pour $\widetilde{C}_{K,n}(\mu)$ la récurrence suivante [41] :

$$\widetilde{C}_{K,n}(\mu) = \min\{\; \widetilde{C}_{K,n-1}(\mu),\; C_{K-1,n-1} \;\} \; + \; (y_n - \mu)^2. \tag{3.18}$$

3.3.2.5 Implémentation de la récurrence fonctionnelle

Intervalle par intervalle. L'espace d'états est infini (contrairement à l'algorithme de Viterbi classique). Une application point par point de la récurrence n'est pas viable, car il y en a une infinité. L'astuce consiste à appliquer la récurrence intervalle par intervalle en faisant de l'analyse. Je donne quelques détails ci-dessous.

Les fonctions $\widetilde{C}_{K,n-1}(\mu)$ et $\widetilde{P}_{\lambda,n-1}(\mu)$ sont définies par morceaux. Sur chaque morceau ou intervalle on a une fonction quadratique définie par trois coefficients : $a_0 + a_1\mu + a_2\mu^2$. Si l'on reprend l'équation (3.17) il nous faut donc calculer sur cet intervalle la fonction $\min\{a_0 + a_1\mu + a_2\mu^2, P_{\lambda,n-1} + \lambda\} + (y_n - \mu)^2$. On l'obtient en calculant le déterminant et les racines du polynôme $(a_0 - P_{\lambda,n-1} - \lambda) + a_1\mu + a_2\mu^2$.

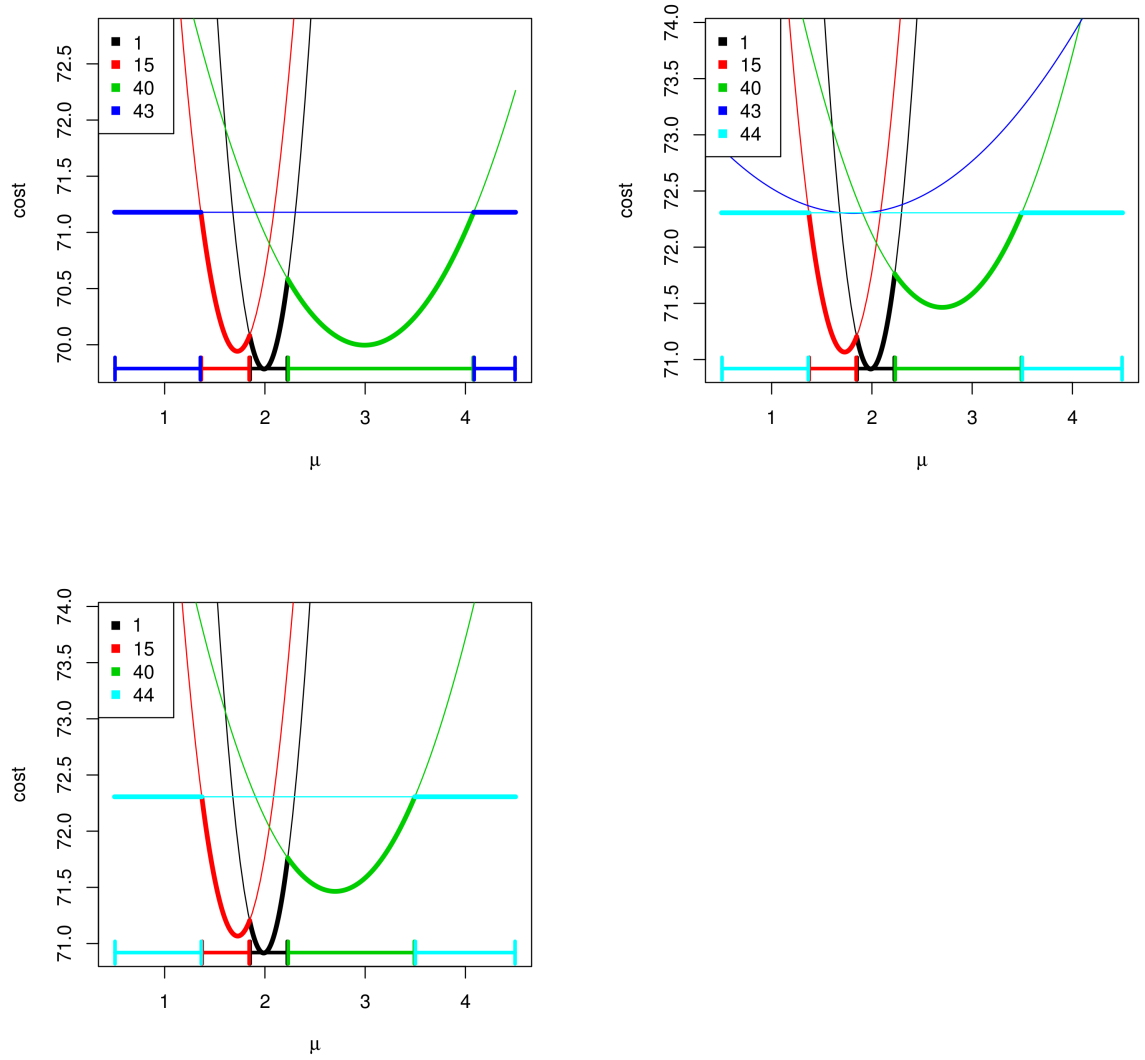

FIGURE 3.4 – Exemple d'exécution de la mise à jour de $\widetilde{C}_{K,n}(\mu)$. Exemple tiré de [12]. $\widetilde{C}_{K,n}(\mu)$ est quadratique par morceaux. En haut à gauche on retrouve le graphe de la figure 3.2. En haut à droite on voit la fonction après l'ajout de l'observation y_{44} : $(y_{44} - \mu)^2$ et la comparaison avec $C_{1,44}$ représenté par la ligne horizontale bleu ciel. On constate que la fonction en bleu, correspondant à une rupture en 43, est battue pour toutes les valeurs de μ. Sur la figure en bas à gauche cette courbe est donc éliminée. Elle n'appartient pas à $\widetilde{C}_{2,44}(\mu)$.

Boucles. Sur le plan algorithmique, pour obtenir $\widetilde{C}_{K,n}(\mu)$ il faudra exécuter la récurrence (3.18) pour tous les $C_{k,t}$ avec $k \leq K$ et $t \leq n$. On le fera à l'aide d'une double boucle sur k et t. J'ai appelé l'algorithme implémentant cette double boucle pDPA (« pruned Dynamic Progaming Algorithm ») dans [41]. La figure 3.4 illustre l'évolution de $\widetilde{C}_{K,n}(\mu)$ sur deux étapes de l'algorithme pDPA.

Pour obtenir $\widetilde{P}_{\lambda,n}(\mu)$ il faudra exécuter la récurrence (3.17) pour tous les $P_{\lambda,t}$ avec $t \leq n$. Une simple boucle sur t suffira. Robert Maidstone, Toby Hocking, Paul Fearnhead et moi-même avons appelé l'algorithme implémentant cette boucle FPOP dans [12].

Complexité. On pourrait craindre que le nombre d'intervalles explose avec n. Ce n'est pas le cas. J'ai démontré que le nombre d'intervalles au temps n est inférieur à $2n - 1$ [41]. La preuve repose sur la convexité des fonctions $\mu \to (y_i - \mu)^2$. De cette borne sur le nombre d'intervalles on déduit une complexité

au pire en $O(Kn^2)$ pour pDPA et $O(n^2)$ pour FPOP. Il est possible de construire des signaux pour lesquels le nombre d'opérations est effectivement en $O(n^2)$. Toutefois, pour de nombreux signaux les temps de calcul sont log-linéaires en n : $O(Kn\log(n))$ ou $O(n\log(n))$.

Répétitions. Dans des données de séquençage il y a souvent des répétitions dans le signal ; plusieurs observations successives sont égales : $y_i = y_{i+1}$. Avec Alice Cleynen, Emilie Lebarbier, Michel Koskas et Stéphane Robin, nous avons demontré dans [5] que les ruptures optimales n'étaient pas dans ces zones répétées. Ce résultat permet de compresser les données tout en garantissant l'exactitude de l'algorithme. Si le taux de compression est important, cela réduit grandement les temps de calcul.

Autres modèles à 1 paramètre. La récurrence fonctionnelle s'étend à des fonctions de perte autres que quadratiques. Elle est vraie pour une vraisemblance de Poisson, binomiale et négative binomiale [5, 12, 41, 81]. La récurrence fonctionne aussi pour des pertes robustes comme la perte biweight [13]. La perte biweight est la perte quadratique tronquée, définie comme suit :

$$\text{Si } |y_i - \theta| \leq K \ : \ \theta \to (y_i - \theta)^2, \quad \text{ et sinon } \ : \ \theta \to K^2$$

Ces pertes sont plus complexes à manipuler. Appliquer la récurrence (3.17) sur chaque intervalle est plus difficile. Néanmoins, la complexité reste en pratique souvent log-linéaire.

Modèles avec plusieurs paramètres. Les formules de récurrence (3.17) et (3.18) sont vraies pour des modèles multi-paramétriques. Leur implémentation est nettement plus difficile. Les ensembles sur lesquels $\widetilde{C}_{K,n}(\mu)$ et $\widetilde{P}_{\lambda,n}(\mu)$ sont définis ne sont plus des intervalles dans $\mathbb{R}$. Ce sont des ensembles non-convexes de $\mathbb{R}^p$. Pour le modèle gaussien p-dimensionnel, par exemple, il faut considérer l'intersection de boules privée de l'union d'autres boules. Je reviens sur ce sujet dans les perspectives.

3.3.2.6 Récurrence sur la dernière rupture et récurrence fonctionnelle

Je considère ici uniquement des modèles où les observations et paramètres sont indépendants. Je parlerai de modèles avec dépendances dans la section 3.4. Pour rappel, l'algorithme PELT [82] élague et accélère la récurrence sur la dernière rupture en se basant sur des inégalités décrites en 3.3.1.3.

PELT est plus générique que FPOP. S'il est possible d'appliquer FPOP, les conditions de PELT de l'équation (3.13) sont vraies avec $\kappa = 0$. S'il est par ailleurs possible de calculer le coût d'un segment efficacement comme expliqué en 3.3.1.2 page 23, PELT est donc applicable. PELT est plus générique que FPOP et s'applique sans difficulté sur des modèles multi-paramétriques et non-paramétriques.

FPOP s'applique sur des pertes robustes. Pour des pertes robustes comme la perte Huber ou Biweight, utile en présence d'*outliers*, le calcul du coût n'est pas simple. Même si l'équation (3.13) est vraie, PELT ne s'applique pas directement. J'ai montré avec Paul Fearnhead que FPOP s'applique sans difficulté [13].

FPOP est plus rapide que PELT. Nous avons démontré dans [12] que la complexité de FPOP est toujours inférieure à celle de PELT. La figure 3.5 représente les temps de calcul de plusieurs approches de détection de ruptures. Les temps de calcul dépendent de nombreux détails de l'implémentation. Il n'est pas toujours facile d'interpréter les différences. Toutefois, on constate sur la partie gauche que FPOP (en rose) est plus rapide que PELT (en orange) en particulier quand le nombre de ruptures est petit. FPOP et PELT sont implémentés en C ou C++.

Plus généralement, on constate sur la figure 3.5 que FPOP est plus rapide que les récentes approches WBS et SMUCE [65, 66]. Sur la figure 3.6 on observe que FPOP a un temps de calcul comparable à la

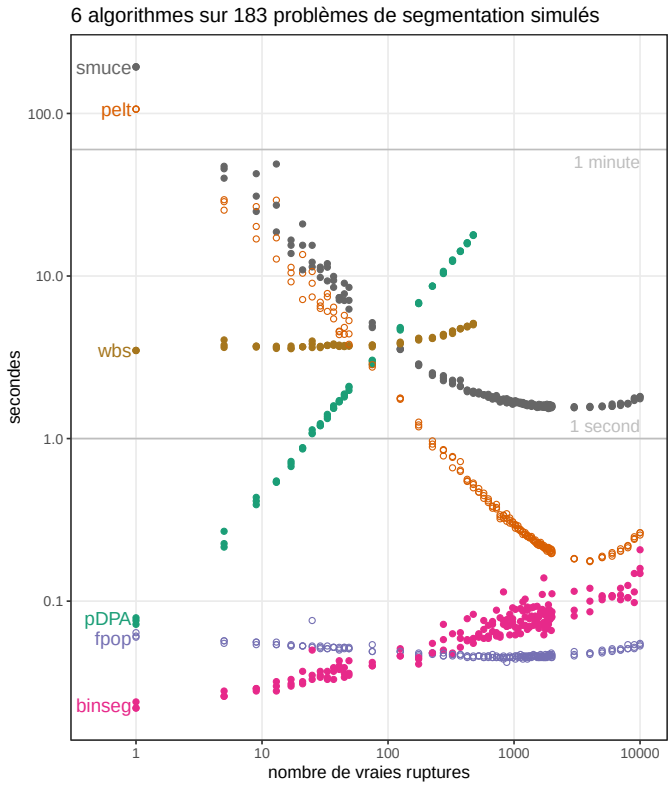

FIGURE 3.5 – Temps de calcul pour quelques méthodes de détection de ruptures. C'est une reprise de la figure 7 du papier [12] pour des profils de tailles $n = 2.10^5$. Certaines approches sont exactes algorithmiquement (au sens où elles résolvent exactement un problème d'optimisation défini) : SMUCE [65], PELT [82], FPOP [12], pDPA [41]. D'autres sont heuristiques sur le plan algorithmique : WBS (Wild Binary Segmentation) [66], binseg (la segmentation binaire). Les codes R pour réaliser ces figures sont disponibles en section supplémentaire de ce chapitre.

segmentation binaire même pour $n = 10^7$. Pour conclure, à l'heure actuelle et pour le modèle gaussien univarié, il ne me semble pas raisonnable d'écarter l'approche par maximum de vraisemblance pénalisée pour des raisons de temps de calcul.

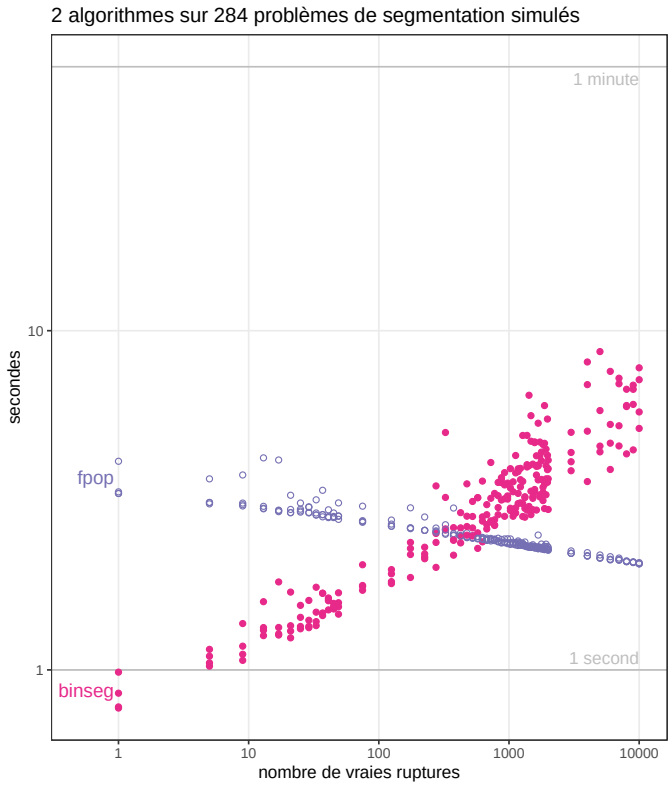

FIGURE 3.6 – Temps de calcul de FPOP et de la segmentation binaire (binseg). C'est une reprise de la figure 7 du papier [12] pour des profils de tailles $n = 10^7$. Les codes R pour réaliser ces figures sont disponibles en section supplémentaire de ce chapitre.

3.4 Récurrence fonctionnelle et dépendances entre paramètres

L'approche fonctionnelle, utilisant une récurrence sur le paramètre du dernier segment, permet de considérer des dépendances entre les paramètres des segments. Ce n'est pas le cas des récurrences sur la position de la dernière rupture. Un certain nombre d'applications justifie d'incorporer des dépendances et contraintes sur la nature du signal dans le modèle [14, 15, 17, 50, 63, 78, 79].

3.4.1 Une application, la détection de pics

Considérons par exemple la détection de pics. Détecter des pics peut être vu comme de la détection de ruptures avec des contraintes sur le sens des sauts. Si une rupture va vers le haut, la suivante doit aller vers le bas et inversement. Mathématiquement, cela se formalise sous forme d'inégalités. Pour tout segment pair $2k$ on a simultanément $\theta_{2k-1} \leq \theta_{2k}$ et $\theta_{2k} \geq \theta_{2k+1}$.

Avec Toby Hocking, nous avons proposé cette contrainte pour modéliser les données Chip-Seq [17]. Ce modèle est schématiquement représenté sur la figure 3.7. Heuristiquement nous avons proposé un algorithme utilisant une récurrence sur la position de la dernière rupture (équation (3.6)). Nous avons pu démontrer, à l'aide d'un contre-exemple, que l'algorithme était effectivement une heuristique : il n'est pas garanti de trouver la solution optimale. Plus tard nous avons proposé un algorithme exact [15], j'y reviendrai.

Statistiquement et intuitivement, ajouter des contraintes sur les paramètres des segments successifs devrait faciliter l'inférence car on restreint l'espace des possibles. Algorithmiquement toutefois, cela complique les choses. La récurrence sur la position de la dernière rupture n'est en général plus valide. L'approche fonctionnelle permet de considérer ces dépendances au prix de quelques difficultés addition-

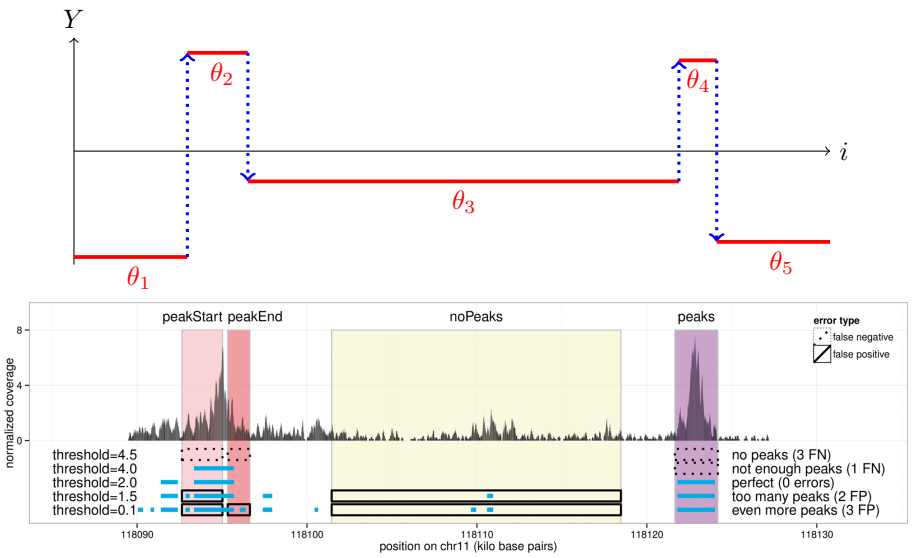

FIGURE 3.7 – En haut : Représentation d'une segmentation en 5 morceaux respectant une contrainte haut-bas : $\theta_{2k-1} \leq \theta_{2k}$ et $\theta_{2k+1} \leq \theta_{2k}$. En bas : Exemple d'un jeu de données Chip-Seq annoté tiré de l'article [17]. Le signal le long du génome est représenté en gris sous forme de barres. L'objectif est de détecter deux pics. Le premier doit commencer dans la zone rose clair et finir dans la zone rose foncé. Le second pic doit être dans la zone violette. Il ne faut pas détecter de pics dans la zone jaune.

nelles [15]. Conceptuellement, il y a deux difficultés supplémentaires à appréhender. Je vais présenter la première en étudiant un modèle avec une contrainte d'isotonicité (3.4.2). J'expliquerai la seconde avec le modèle haut-bas pour la détection de pics présenté ci-dessus (3.4.3). Je terminerai par une présentation informelle de l'algorithme général (3.4.4).

3.4.2 Récurrence fonctionnelle et isotonie

Le modèle. L'idée du modèle isotonique est de contraindre les moyennes des segments successifs à croître. Mathématiquement on écrit $\theta_k \leq \theta_{k+1}$. Ce modèle est schématiquement représenté sur la figure 3.8.

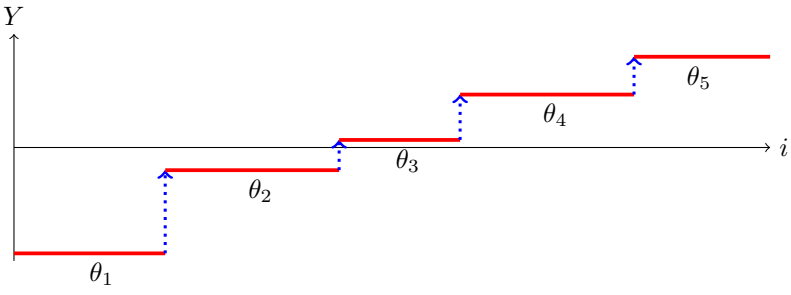

FIGURE 3.8 – Représentation d'une segmentation en 5 morceaux respectant une contrainte d'isotonicité : $\theta_k \leq \theta_{k+1}$.

Ce modèle isotonique est très étudié en statistique et connu sous le nom de régression isotonique [88]. Dans cette littérature, il n'y a en général pas de pénalité pour l'ajout d'une rupture. D'un point de vue algorithmique les problèmes à résoudre sont :

$$\widetilde{C}^{Iso}_{K,n}(\mu) = \min_{\substack{\tau_1,\dots\tau_{K-1}\\ \theta_1<\dots<\theta_K=\mu}} \left\{ \sum_{k=1}^{K} \sum_{i=\tau_{k-1}+1}^{\tau_k} (y_i-\theta_k)^2 \right\} \tag{3.19}$$

et

$$\widetilde{P}^{Iso}_{\lambda,n}(\mu) = \min_{\substack{K\\ \tau_1,\dots\tau_{K-1}\\ \theta_1<\dots<\theta_K=\mu}} \left\{ \sum_{k=1}^{K} \sum_{i=\tau_{k-1}+1}^{\tau_k} (y_i-\theta_k)^2 + \lambda K \right\}. \tag{3.20}$$

Pour une pénalité nulle ($\lambda = 0$), il existe des algorithmes très rapides pour retrouver la solution de $\widetilde{P}^{Iso}_{0,n}$. Par exemple, l'algorithme PAVA [88] a une complexité linéaire $O(n)$. L'algorithme de programmation dynamique que je vais présenter est plus lent. Sa récurrence est toutefois légèrement plus générique. Elle prend naturellement en compte une pénalité et s'étend à des pertes robustes non-convexes comme la perte biweight. Ce n'est pas le cas, à ma connaissance, de l'algorithme PAVA. Les simulations décrites dans la figure 2 de [49] montrent bien l'intérêt d'utiliser des pertes robustes non-convexes dans un contexte isotonique : elles permettent une bonne estimation du signal même avec 50% de données corrompues.

3.4.2.1 La récurrence

Essayons de construire une récurrence pour le problème (3.19) et (3.20). Pour les récurrences (3.17) et (3.18), en cas de saut, il faut considérer la meilleure valeur possible pour μ au temps $n-1$. Avec la contrainte, il semble naturel de se restreindre aux μ' inférieurs à μ. Cette intuition est juste. Mathématiquement on obtient :

$$\widetilde{C}^{Iso}_{K,n}(\mu) = \min \left\{ \widetilde{C}^{Iso}_{K,n-1}(\mu),\ \min_{\mu'\le\mu}\{\widetilde{C}^{Iso}_{K-1,n-1}(\mu')\} \right\} + (y_n-\mu)^2, \tag{3.21}$$

$$\widetilde{P}^{Iso}_{\lambda,n}(\mu) = \min \left\{ \widetilde{P}^{Iso}_{\lambda,n-1}(\mu),\ \min_{\mu'\le\mu}\{\widetilde{P}^{Iso}_{\lambda,n-1}(\mu')\} + \lambda \right\} + (y_n-\mu)^2. \tag{3.22}$$

Pour $\lambda = 0$ on obtient une récurrence très similaire à celle proposée par G. Rote pour la perte ℓ_1 [95].

3.4.2.2 Opérateur minimum à gauche

On définit l'opérateur minimum à gauche $\widetilde{\min}$ comme suit :

$$\widetilde{\min}(f)(\mu) = \min_{\mu'\le\mu} f(\mu'). \tag{3.23}$$

Pour effectuer la récurrence (3.21) ou (3.22) il faut calculer $\widetilde{\min}\widetilde{C}^{Iso}_{K-1,n-1}$ ou $\widetilde{\min}\widetilde{P}^{Iso}_{\lambda,n-1}$. On peut ré-écrire les récurrences :

$$\widetilde{C}^{Iso}_{K,n}(\mu) = \min \left\{ \widetilde{C}^{Iso}_{K,n-1}(\mu),\ \widetilde{\min}(\widetilde{C}^{Iso}_{K-1,n-1})(\mu) \right\} + (y_n-\mu)^2, \tag{3.24}$$

$$\widetilde{P}^{Iso}_{\lambda,n}(\mu) = \min \left\{ \widetilde{P}^{Iso}_{\lambda,n-1}(\mu),\ \widetilde{\min}(\widetilde{P}^{Iso}_{\lambda,n-1})(\mu) + \lambda \right\} + (y_n-\mu)^2. \tag{3.25}$$

On peut démontrer que $\widetilde{\min}\widetilde{C}^{Iso}_{K-1,n-1}$ et $\widetilde{P}^{Iso}_{\lambda,n-1}(\mu)$ sont convexes et quadratiques par morceaux [14, 15, 42] et qu'il est possible de les calculer efficacement. L'implémentation de l'opérateur $\widetilde{\min}$ est un peu fastidieuse. Je donne les étapes clés de ce calcul dans 3.4.2.3. Ces détails ne sont pas essentiels pour la suite de l'exposé et peuvent être ignorés. L'important est de comprendre qu'il est possible de calculer l'opérateur pour exécuter la récurrence.

Sur le plan mathématique, les récurrences (3.21) et (3.22) s'étendent à d'autres contraintes. Le calcul de l'opérateur est possible pour des contraintes linéaires d'égalité ou d'inégalité [14, 15, 42]. Pour des contraintes non-linéaires, l'opérateur obtenu est bien souvent difficile à calculer. Cela limite l'implémentation pratique de la récurrence. Les récurrences s'étendent également à des modèles non-gaussiens. Pour des pertes robustes (Huber ou biweight) les mêmes opérateurs peuvent être implémentés. Pour une vraisemblance Poisson, binomiale ou négative binomiale il y a des restrictions : tous les opérateurs ne se calculent pas facilement.

3.4.2.3 Détails sur l'implémentation de l'opérateur minimum à gauche

Considérons le calcul de $\widetilde{\min}(G)$ où G est une fonction convexe et quadratique par morceaux. Supposons que G soit définie sur M intervalles $\{I_m\}_{m\leq M}$. Sur chaque intervalle I_m on a $G = g_m$ avec g_m une fonction quadratique et convexe.

Quelques propriétés. Commençons par deux propriétés simples de l'opérateur $\widetilde{\min}$:

1. Si g est convexe on calcule $\widetilde{\min}(g)$ comme suit. On calcule $\mu^* = \arg\min_\mu g(\mu)$. Puis on a :

$$\begin{aligned}\widetilde{\min}(g)(\mu) &= \quad g(\mu), \quad \forall\ \mu \leq \mu^*,\\ \widetilde{\min}(g)(\mu) &= \quad g(\mu^*), \quad \forall\ \mu \geq \mu^*.\end{aligned}$$

2. Si g est quadratique et convexe par morceaux, $\widetilde{\min}(g)$ est aussi quadratique et convexe par morceaux.

Calcul en quatre étapes. Pour calculer $\widetilde{\min}(G)$ on procède en quatre étapes :

1. On calcule sur chaque intervalle I_m la fonction $\widetilde{\min}(g_m)$.
2. On calcule le minimum a_m de chaque fonction g_m sur son intervalle I_m :

$$a_m = \min_{\mu\in I_m} g_m(\mu).$$

3. On calcule le minimum à gauche de chaque intervalle I_m par récurrence : $b_m = \min\{b_{m-1}, a_m\}$, en initialisant b_0 à $+\infty$.
4. Sur chaque intervalle I_m on obtient alors $\widetilde{\min}(G)$ en comparant b_m et $\widetilde{\min}(g_m)$:

$$\forall\ \mu\ \in\ I_m \quad \widetilde{\min}(G) = \min\{b_m, \widetilde{\min}(g_m)(\mu)\}.$$

3.4.3 Récurrence fonctionnelle et pics

Il est temps d'aborder la deuxième difficulté pour prendre en compte des contraintes complexes dans la récurrence fonctionnelle. Considérons pour cela le cas d'une contrainte haut-bas. Pour rappel, il s'agit d'avoir une alternance de ruptures vers le haut et de ruptures vers le bas. Mathématiquement, pour tout k on a $\theta_{2k\pm1} \leq \theta_{2k}$. La première rupture va vers le haut. On dira que le segment initial est dans un état bas. Dans les formules on le notera $\downarrow$. Pour la détection de pics il est assez naturel que la dernière rupture soit vers le bas. L'état final est donc bas $\downarrow$. On aura un nombre impair de segments. Les notations deviennent un peu lourdes, les deux problèmes à résoudre sont formellement :

$$\widetilde{C}^{Pic,\downarrow}_{2K+1,n}(\mu) = \min_{\substack{\tau_1,\dots\tau_{2K}\\ \theta_{2k\pm1}\leq\theta_{2k}\\ \theta_{2K+1}=\mu}} \left\{ \sum_{k=1}^{2K+1} \sum_{i=\tau_{k-1}+1}^{\tau_k} (y_i-\theta_k)^2 \right\}, \tag{3.26}$$

$$\widetilde{P}^{Pic,\downarrow}_{\lambda,n}(\mu) = \min_{\substack{K\\ \tau_1,\dots\tau_{2K}\\ \theta_{2k\pm1}\leq\theta_{2k}\\ \theta_{2K+1}=\mu}} \left\{ \sum_{k=1}^{2K+1} \sum_{i=\tau_{k-1}+1}^{\tau_k} (y_i-\theta_k)^2 \ + \ \lambda K \right\}. \tag{3.27}$$

Pour propager la récurrence, l'idée consiste à considérer le même problème avec un état final vers le haut : $\widetilde{P}^{Pic,\uparrow}_{\lambda,n}$ et $\widetilde{C}^{Pic,\uparrow}_{2K,n}$. Voilà la définition formelle de ces deux fonctions :

$$\widetilde{C}^{Pic,\uparrow}_{2K,n}(\mu) = \min_{\substack{\tau_1,\dots\tau_{2K-1}\\ \theta_{2k\pm1}\leq\theta_{2k}\\ \theta_{2K}=\mu}} \left\{ \sum_{k=1}^{2K} \sum_{i=\tau_{k-1}+1}^{\tau_k} (y_i-\theta_k)^2 \right\}, \tag{3.28}$$

$$\widetilde{P}^{Pic,\uparrow}_{\lambda,n}(\mu) = \min_{\substack{K\\ \tau_1,\dots\tau_{2K-1}\\ \theta_{2k\pm1}\leq\theta_{2k}\\ \theta_{2K}=\mu}} \left\{ \sum_{k=1}^{2K} \sum_{i=\tau_{k-1}+1}^{\tau_k} (y_i-\theta_k)^2 \ + \ \lambda K \right\}. \tag{3.29}$$

Considérons la mise à jour de $\widetilde{P}^{Pic,\uparrow}_{\lambda,n}$ à l'étape n pour une valeur μ. Deux situations sont à envisager :

1. Il y a une rupture à l'étape n. La rupture est nécessairement vers le bas. Nous étions donc dans un état $(\downarrow,\mu')$ au temps $(n-1)$. Il faut chercher la meilleure solution. On l'obtient avec l'opérateur minimum à gauche $\widetilde{\min}(\widetilde{P}^{Pic,\downarrow}_{\lambda,n-1})(\mu)$ défini à l'équation (3.23).
2. La rupture n'est pas à l'étape n. Nous étions donc déjà dans l'état $(\uparrow,\mu)$ au temps $(n-1)$.

Formellement on démontre la récurrence suivante :

$$\widetilde{P}^{Pic,\uparrow}_{\lambda,n}(\mu) = \min\left\{ \ \widetilde{P}^{Pic,\uparrow}_{\lambda,n-1}(\mu),\ \widetilde{\min}(\widetilde{P}^{Pic,\downarrow}_{\lambda,n-1})(\mu)+\lambda \ \right\} \ + \ (y_n-\mu)^2.$$

Il faut également considérer la mise à jour de $\widetilde{P}^{Pic,\downarrow}_{\lambda,n}$. Pour cela nous aurons besoin de l'opérateur minimum à droite $\widehat{\min}$:

$$\widehat{\min}(f) = \min_{\mu'\geq\mu} f(\mu'),$$

On obtient alors une récurrence proche de (3.4.3) pour $\widetilde{P}^{Pic,\downarrow}_{\lambda,n}$:

$$\widetilde{P}^{Pic,\downarrow}_{\lambda,n}(\mu) = \min\left\{ \ \widetilde{P}^{Pic,\downarrow}_{\lambda,n-1}(\mu),\ \widehat{\min}(\widetilde{P}^{Pic,\uparrow}_{\lambda,n-1})(\mu)+\lambda \ \right\} \ + \ (y_n-\mu)^2.$$

On peut démontrer des formules analogues pour $\widetilde{C}^{Pic,\uparrow}_{2K,n}(\mu)$ et $\widetilde{C}^{Pic,\downarrow}_{2K+1,n}(\mu)$ [15]. Je ne les décris pas ici.

3.4.3.1 Retour sur le modèle HMM avec contraintes

Il est intéressant à ce stade de revenir à la modélisation HMM de ce modèle haut-bas. Par rapport au modèle sans contrainte décrit en sous-section 3.3.2.3 l'espace d'états est un peu plus grand. Appelons $\mathcal{S}$ l'ensemble $\{\uparrow,\downarrow\}$. L'espace d'états des Z_i est $\mathcal{S}\times\mathbb{R}$. La règle de chaînage entre les Y_i et les Z_i reste la même : $(Y_i|Z_i=\mu)\sim\mathcal{N}(\mu,\sigma^2)$. Il faut par contre modifier le noyau de transition pour prendre en compte $\mathcal{S}$ et les contraintes. Le noyau est une fonction de $(\mathcal{S}\times\mathbb{R})^2$ vers $\mathbb{R}$, $k((s,x),(s',y))$, définie comme suit :

— Pour tout x on a $k((\uparrow,x),(\uparrow,x))=1$ et $k((\downarrow,x),(\downarrow,x))=1$. Cela modélise la probabilité de rester dans le même état.

— Pour $x\geq y$ on a $k((\uparrow,x),(\downarrow,y))=e^{-\lambda}$. Cela modélise la probabilité d'une rupture vers le bas.

— Pour $x \leq y$ on a $k((\downarrow, x), (\uparrow, y)) = e^{-\lambda}$. Cela modélise la probabilité d'une rupture vers le haut.
— Dans tous les autres cas le noyau est nul. Cela interdit deux ruptures consécutives vers le haut ou vers le bas.

La figure 3.9 représente sous forme graphique ce noyau.

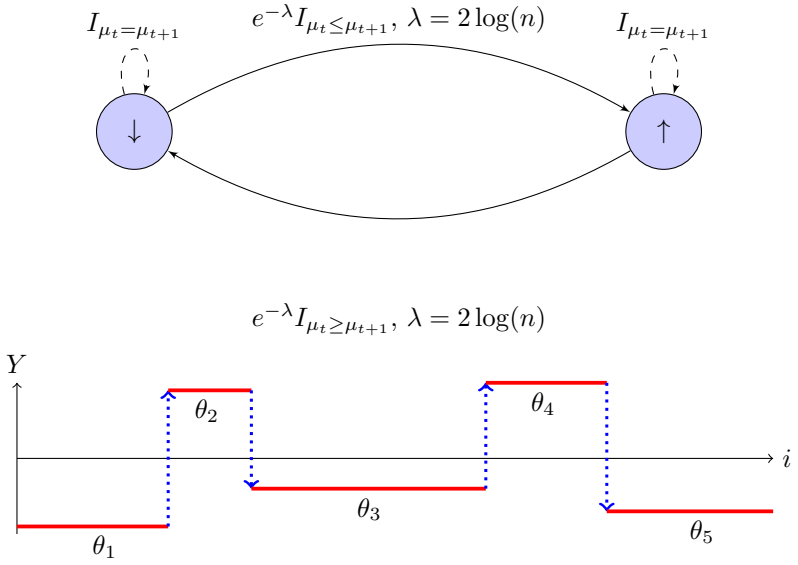

FIGURE 3.9 – En haut : Représentation graphique du noyau de transition du modèle haut-bas. L'espace d'états est le produit cartésien entre $\{\uparrow, \downarrow\}$ et $\mathbb{R}$. Les nœuds représentent des états $(\downarrow, \mu)$ et $(\uparrow, \mu)$. Les nœuds $\emptyset$ et . représentent les états initial et final. Le signal commence et finit dans un état $\downarrow$. Les arêtes décrivent la pénalité et les contraintes linéaires des transitions entre états. Par exemple, l'arête du nœud $\downarrow$ vers $\downarrow$ indique une contrainte d'égalité pour une pénalité nulle. L'arête du nœud $\downarrow$ vers $\uparrow$ indique une contrainte d'accroissement pour une pénalité de $\lambda = 2\log(n)$. En bas : Le signal représenté respecte les contraintes. Il démarre dans un état $\downarrow$ avec $\mu = \theta_1$. Il alterne bien entre $\downarrow$ et $\uparrow$. Il se termine bien dans un état $\downarrow$ avec $\mu = \theta_5$.

3.4.4 Récurrence fonctionnelle, généralisation

On peut généraliser cette modélisation et ce graphe de contraintes à des $\mathcal{S}$ plus grands. Dans [15, 42] nous avons proposé un algorithme pour l'inférence de ces modèles. L'algorithme fonctionne si les contraintes sont linéaires. Je ne vais pas présenter formellement cet algorithme.

Intuition de l'algorithme. De manière intuitive l'algorithme suit une fonctionnelle pour chaque état discret s. On peut démontrer une récurrence fonctionnelle pour chaque s. La récurrence implique de comparer plusieurs fonctionnelles transformées par un opérateur au temps $n-1$.

Mathématiquement, il est possible de formaliser ces récurrences en une seule et unique récurrence. L'algorithme résultant, gfpop, est décrit dans [15] et [42]. Une implémentation générique de l'algorithme, c'est-à-dire fonctionnant pour tout graphe de contraintes linéaires, est disponible sur la page github de Vincent Runge [99].

Temps de calcul. Actuellement, il n'existe aucune garantie sur la complexité au pire de l'algorithme gfpop. En pratique, les temps de calculs semblent dépendre assez fortement de la nature des données et des contraintes. Néanmoins, pour plusieurs modèles nous avons constaté empiriquement que les temps de calculs sont acceptables. Par exemple pour le modèle haut-bas avec la perte quadratique on obtient des temps de calcul inférieurs à 20 secondes pour $n = 10^6$.

3.5 Récurrence fonctionnelle, bilan et perspectives

Comme expliqué dans 3.3.2.6, le premier intérêt de la récurrence fonctionnelle est le temps de calcul réduit même quand il y a peu de ruptures. Le deuxième est sans aucun doute la possibilité d'inférer des modèles avec dépendances. La récurrence classique sur la dernière rupture ne le permet pas. À ma connaissance, l'inférence par maximum de vraisemblance de modèles avec structures de dépendances n'était jusque-là pas envisagée.

3.5.1 Perspectives

La récurrence fonctionnelle ouvre un certain nombre de perspectives sur le plan statistique et algorithmique mais aussi en termes de modélisation.

J'identifie au moins trois grandes problématiques :

1. l'inférence de ruptures en présence de dépendances,
2. l'inférence de ruptures sur des arbres ou des graphes,
3. la construction d'algorithmes pour des modèles multi-variés.

Les trois problématiques peuvent se mélanger. Par exemple en phylogénie il semble intéressant de considérer des modèles de ruptures multiples sur arbre avec une structure de dépendance [52].

Dépendances, quelques aspects méthodologiques. Les algorithmes gfpop [14, 42] et cpop [63] permettent l'inférence de modèle avec des dépendances entre les paramètres des segments successifs. Ils ouvrent la voie vers l'inférence par maximum de vraisemblance d'autres formes de dépendances, notamment dans le bruit. Par exemple on peut penser à un bruit auto-régressif [59]. Des résultats préliminaires obtenus avec Gaetano Romano, Vincent Runge et Paul Fearnhead me laissent penser que la récurrence fonctionnelle s'étend à des modèles avec un bruit auto-régressif.

Au-delà de cet exemple, sur le plan algorithmique, il s'agirait de déterminer la classe de modèles pour laquelle la récurrence fonctionnelle est opérante. Dans le cas gaussien, il faudrait savoir quels types de dépendances préservent la décomposition de la vraisemblance comme un polynôme par morceaux du paramètre du dernier segment. Il s'agirait aussi d'obtenir des garanties théoriques sur la complexité au pire et en espérance. Cela me semble possible au moins dans certains cas particuliers.

Sur le plan statistique, beaucoup d'outils restent à développer. En particulier la question de la sélection de modèles en présence de dépendances est peu développée dans la littérature. Une idée serait d'essayer d'étendre les résultats de Lebarbier [85]. Les travaux récents de Chakar [59] sur des approximations de la vraisemblance et de Gao [67] dans le cas isotonique sont prometteurs. De même, il manque des outils pour mesurer l'incertitude sur la position des ruptures. Une idée assez naturelle pour cela serait de considérer une approche bayesienne.

Dépendances, un peu de modélisation. Le noyau de transition du modèle de ruptures vu comme un HMM permet de modéliser des connaissances a priori sur la nature du signal. Sur le plan applicatif, pour choisir ces a priori il faudra, comme je l'expliquerai au chapitre 4, discuter avec les praticiens et remettre en cause de nombreuses fois le modèle. Voilà toutefois quelques pistes qui me semblent intéressantes pour des applications en biologie.

— On peut modéliser le sens des transitions. Une alternance haut-bas modélise des pics simples. On peut imaginer deux ruptures vers le haut suivies de deux vers le bas pour des structures de pic plus complexes. On peut considérer des schémas asymétriques. Par exemple, pour détecter des potentiels d'action en neurologie [78] on peut considérer un saut vers le haut suivi d'une décroissance isotonique [42].

— On peut modéliser une hauteur minimale pour les transitions. L'intérêt est de ne détecter que des ruptures suffisamment importantes et facilement interprétables. Cette idée est proposée dans un autre cadre statistique dans [62].
— On peut modéliser la longueur des segments. En particulier on peut contraindre les segments à être suffisamment longs. On évite ainsi la détection de très petits segments qui sont bien souvent très difficiles à interpréter.

Modèle multi-dimensionnel. Pour des modèles multivariés avec p paramètres par segment et des données avec un nombre de ruptures linéaire en la taille des données, l'algorithme PELT [82] permet d'obtenir le maximum de vraisemblance pénalisé rapidement. S'il y a peu de ruptures, la complexité de PELT est quadratique. Pour des profils avec plus de 10^6 observations il faut compter plusieurs heures de calcul. Cela limite son application en génomique où les profils ont souvent plus de 10^6 points.

Pour ces modèles multivariés la récurrence fonctionnelle est valide et mathématiquement elle doit conduire à un meilleur élagage que PELT [12]. Toutefois, comme expliqué à la fin de 3.3.2.5 l'implémentation de la récurrence n'est pas simple. Les morceaux sur lesquels le coût fonctionnel est défini sont des ensembles non-convexes de $\mathbb{R}^p$. Cela soulève des problèmes de géométrie et d'optimisation assez complexes [96]. Je ne rentrerai pas dans les détails. Malgré tout, au moins en petite dimension ($p = 3$), l'algorithme cpop [63] et des travaux préliminaires avec Vincent Runge et Paul Fearnhead me laissent penser qu'une implémentation rapide de la récurrence est possible.

Arbres et graphes. Il existe maintenant des algorithmes efficaces pour détecter des ruptures sur données ordonnées (le long du temps ou du génome) [4, 12, 82]. Au vu de ces algorithmes, il semble intéressant de reconsidérer le problème de segmentation sur des données partiellement ordonnées comme des arbres et même plus généralement sur un graphe. Sur le plan applicatif un certain nombre d'applications en écologie (sur des réseaux de rivières) et en phylogénie [52] pose explicitement le problème de détection de ruptures sur des arbres.

C'est un problème que j'ai commencé à étudier avec Solène Thépaut dans le cadre de sa thèse avec Christophe Giraud et Nicolas Verzelen. La difficulté essentielle sur des arbres par rapport à une chaîne est le nombre de segments. Dans une chaîne avec n nœuds il y a $n(n-1)/2$ segments. Dans un arbre avec n nœuds il peut y en avoir $2^{n-1} + (n-1)$. Cela pose un certain nombre de problèmes algorithmiques et statistiques. Sur le plan algorithmique en fonction du coût utilisé le problème peut devenir NP-difficile [89]. Sur le plan statistique un certain nombre de preuves de consistance repose sur le nombre limité de segments [66, 68]. Il semble possible de contourner un certain nombre de ces difficultés grâce à la récurrence fonctionnelle et en adaptant les travaux de [85] en sélection de modèle.

3.5.2 Conclusion

J'ai initié mes travaux sur la détection de ruptures il y a presque 10 ans. À la fin de ma thèse, j'ai commencé seul des travaux sur l'inférence rapide par maximum de vraisemblance pénalisé. Cela m'a conduit de fils en aiguilles plus loin que je ne l'aurais imaginé. Pour terminer ce chapitre, je tiens à remercier tous les algorithmiciens, statisticiens, bioinformaticiens et biologistes avec qui j'ai eu la chance de travailler sur ce sujet.

3.6 Codes supplémentaires

Ci-dessous les codes pour réaliser la figure 3.5. Le prochain chapitre est à la page 43.

3.6.1 Code pour des profils de cent-milles points

```
require(changepoint)
require(stepR)
require(wbs)
require(jointseg)
require(fpop)
########################################
seg.funs <-
  list(smuce=function(one.chrom, Kmax){
    system.time({
      smuceR(one.chrom)
    })[["user.self"]]
  },
pDPA=function(one.chrom, Kmax){
if(Kmax <= 500){
timings <- system.time(
Fpsn(one.chrom, Kmax)
   )[["user.self"]]
} else {
timings <- NA
}
return(timings)
  },
    wbs=function(one.chrom, Kmax){
          if(Kmax <= 500){
            system.time({
              w <- wbs(one.chrom)
              changepoints(w, Kmax=Kmax)
            })[["user.self"]]
          } else {
            NA
          }
  },
pelt=function(one.chrom, Kmax=NA){
if(Kmax >= 1){
timings <- system.time(
cpt.mean(one.chrom,
method="PELT", penalty="Manual",
              pen.value=log(length(one.chrom)))
   )[["user.self"]]
} else {
timings <- NA
}
return(timings)
  },
fpop=function(one.chrom, Kmax=NA){
system.time(
Fpop(one.chrom,
```

```
     log(length(one.chrom)))
)[["user.self"]]
  },
binseg=function(one.chrom, Kmax){
     system.time(
multiBinSeg(one.chrom, Kmax)
)[["user.self"]]
}
)

##########################################
n <-2*10^5
repet <- 3
K <- c(seq(1, 50, by =4), seq(75, 2000, by =50), seq(2*10^3, 10^4, by =1000))
set.seed(100)
signal.list <- list()
iS <- 1

for(iR in 1:repet){
for(iK in K){
signal.list[[iS]] <- list()
signal.list[[iS]]$Ktrue <- iK

if(iK > 1){
bkp <- sort(sample(1:(n-1), iK-1))
lg <- diff(c(0, bkp, n)) ### rep(n/K, K)
signal <- rep(rep(c(0, 1), length(lg))[1:length(lg)], lg)  [1:n]
} else {
signal <- rep(0, n)
}

signal.list[[iS]]$signal <- signal+ rnorm(n,sd=0.5)
iS <- iS+1
}
}


##########################################
systemtime.simulation <- NULL
for(pid.chr.i in seq_along(signal.list)){
  cat(pid.chr.i,  "%\n")
  pid.chr <- names(signal.list)[pid.chr.i]
  for(algorithm in names(seg.funs)){
    cat(algorithm,  "%\n")
    fun <- seg.funs[[algorithm]]
    seconds <- fun(one.chrom=signal.list[[pid.chr.i]]$signal,
   Kmax= signal.list[[pid.chr.i]]$Ktrue+1)

    systemtime.simulation <- rbind(systemtime.simulation,
   data.frame(algorithm, pid.chr.i,
           signal.list[[pid.chr.i]]$Ktrue, seconds)
     )
  }
```

```
}
colnames(systemtime.simulation)[2:4] <- c("id", "Ktrue", "seconds")

save(systemtime.simulation, file="systemtime.simulation.RData")
```

3.6.2 Code pour des profils de dix millions de points

```
require(fpop)
##########################################
seg.funs <-
  list(fpop=function(one.chrom, Kmax=NA){
system.time(
Fpop(one.chrom,
log(length(one.chrom)))
)[["user.self"]]
  },
binseg=function(one.chrom, Kmax){
     system.time(
multiBinSeg(one.chrom, Kmax)
)[["user.self"]]
}
)

##########################################
n <- 10^7
repet <- 4
K <- c(seq(1, 50, by =4), seq(75, 2000, by =50), seq(2*10^3, 2*10^4, by =1000))
set.seed(100)
signal.list <- list()
iS <- 1

systemtime.simulation <- NULL
for(iR in 1:repet){
for(iK in K){
cat(iR, iK, "\n")
### simu
if(iK > 1){
bkp <- sort(sample(1:(n-1), iK-1))
lg <- diff(c(0, bkp, n)) ### rep(n/K, K)
signal <- rep(rep(c(0, 1), length(lg))[1:length(lg)], lg)  [1:n]
} else {
signal <- rep(0, n)
}
signalToAnalyze <- signal+ rnorm(n,sd=0.5)
iS <- iS+1


# runtimes

for(algorithm in names(seg.funs)){
     cat(algorithm,  "%\n")
```

```
     fun <- seg.funs[[algorithm]]
       seconds <- fun(one.chrom=signalToAnalyze, Kmax= iK+1)

     systemtime.simulation <- rbind( systemtime.simulation,
data.frame(algorithm, iS, iK, seconds)
 )
   }


}
}
#################
colnames(systemtime.simulation)[2:4] <- c("id", "Ktrue", "seconds")

save(systemtime.simulation, file="systemtime.simulationLarge.RData")
```

Chapitre 4

Analyse de jeux de données omiques

C'est une occupation très jolie. C'est véritablement utile puisque c'est joli.

Le Petit Prince - Saint-Exupéry

Résumé

Une part importante de mon travail de recherche est d'analyser des jeux de données omiques en collaboration avec des biologistes, des bioinformaticiens et des statisticiens. Dans la majorité de ces analyses la question biologique peut s'exprimer simplement et les outils statistiques mis en œuvre sont faciles à utiliser. Toutefois bien interpréter la question biologique et bien exploiter ces outils statistiques n'est pas simple. Cela prend du temps. Conduire l'analyse est complexe, car il faut essayer d'être exigeant biologiquement, bioinformatiquement et statistiquement pour obtenir le meilleur résultat. Pour cela j'essaie d'instaurer un dialogue entre disciplines.

4.1 Quelques analyses de données omiques

J'ai été impliqué dans de nombreuses analyses omiques. Dans certains cas, j'ai analysé les données moi-même dans d'autres, j'ai simplement donné des conseils. Voilà une liste des techniques statistiques que j'ai souvent mises en œuvre :

- des tests de student, de Wilcoxon et de Fisher exact dans [20, 23, 27, 34, 36-39] ;
- des modèles linéaires et des modèles linéaires généralisés dans [22, 24, 25, 27-29, 31, 32, 36, 38] ;
- des modèles de détection de ruptures multiples dans [21, 26, 27, 30, 31, 33, 40] ;
- des techniques d'analyse exploratoire comme l'analyse en composante principale ou la classification hiérarchique ascendante.

En génomique les technologies évoluent rapidement et de nouveaux types de données apparaissent régulièrement. Des outils statistiques sophistiqués sont disponibles pour les analyser. Ces outils évoluent rapidement et permettent des analyses toujours plus efficaces et précises. Souvent ils sont en outre faciles à utiliser ou « user-friendly ». Ces nouveaux outils ne sont pas la seule clé d'une analyse réussie. Il faut aussi bien les mettre en œuvre. Pour cela il me semble important de développer une démarche statistique classique : il faut réfléchir à l'objectif biologique de l'analyse, modéliser statistiquement les données, penser à l'interprétation biologique du modèle.

J'ai toujours aimé analyser des jeux de données en collaboration avec des biologistes et des bioinformaticiens. Je trouve que c'est une jolie activité. Si l'on en croit le petit prince (voir en haut de page), c'est donc que c'est une activité utile, qui comprend un certain nombre de difficultés réelles. J'en présente quelques facettes dans la suite de ce chapitre.

4.2 Analyses et dialogues interdisciplinaires

J'ai eu l'occasion de discuter informellement avec de nombreux biologistes, bioinformaticiens et statisticiens. Je pense que certains sous-évaluent la difficulté de réaliser une analyse omique. A première vue, l'analyse semble en effet facile :

1. La question biologique est exprimée simplement. Par exemple « Je souhaite identifier les gènes sur-exprimés dans ce type de cancer ».
2. Les outils statistiques à mettre en œuvre sont faciles d'utilisation. Par exemple, quelques lignes de commande R suffisent pour lancer une analyse différentielle RNAseq avec edgeR [93].

Une question simple et des outils faciles, on croirait qu'il n'y a pas de problème. Pourtant, rappelons-nous que la question est une petite étape d'un projet de recherche en biologie ou en bioinformatique et qu'elle porte sur des milliers de gènes ou d'entités biologiques. Souvenons-nous que les packages R implémentent des stratégies mathématiques complexes. Justifier le choix d'une méthode pour analyser et interpréter biologiquement les données ne saurait être simple ! Pour bien faire les choses, je crois qu'il faut prendre le temps d'écouter les arguments de la biologie, de la bioinformatique et des statistiques.

J'essaie de l'illustrer dans la suite de ce chapitre au travers de deux analyses. La première, inspirée de mes recherches, parle de la difficulté de choisir une méthode plutôt qu'une autre même dans un cas en apparence simple. La seconde montre que l'interprétation biologique d'un modèle statistique n'est pas toujours évidente ou naturelle.

4.2.1 Hypothèses et choix de modèles

4.2.1.1 Un problème classique

Pour comparer deux populations, deux tests sont souvent considérés, le test de Student et le test de Wilcoxon. Sur wikipedia [105] on peut lire à propos du test de Wilcoxon :

> can be used as an alternative to the paired Student's t-test when the sample size is small and the population cannot be assumed to be normally distributed.

Il s'agirait donc de savoir si la distribution est gaussienne ou non. De nombreux livres, forums et articles reviennent sur ce choix entre test de Wilcoxon et test de Student. Je ne les ai pas tous lus et je ne prétends pas en faire la synthèse ici. Il me semble juste qu'un certain nombre résume le choix à la question suivante : Avez-vous le courage de supposer que vos données sont normales ?

Je pense - beaucoup d'autres l'ont dit avant moi, par exemple [87] - que la « bonne question » est plus interdisciplinaire. Il s'agit certes de savoir si le test est valide statistiquement mais aussi de vérifier que les hypothèses et quantités calculées sont réalistes ou pertinentes pour la question scientifique. À ce titre, l'hypothèse de normalité du test de Student n'est pas très réaliste. Les hypothèses du test de Wilcoxon sont mathématiquement plus faibles, mais elles ne sont pas toujours plus réalistes. Je propose d'étudier un cas pratique, inspiré de plusieurs analyses que j'ai faites dans le passé pour l'illustrer.

4.2.1.2 Un cas pratique

Imaginons la situation suivante. Une biologiste, que j'appellerai Maria, mesure par dPCR (digital PCR) l'expression du gène A dans une lignée cellulaire. Elle souhaite savoir s'il y a une différence d'expression entre les cellules traitées et non-traitées avec un médicament. Maria a réalisé $n = 3$ réplicats biologiques, ce qui est assez classique pour ce genre d'expérience. Elle nous demande d'analyser les données. Imaginons que nous hésitions entre le test de Student et le test de Wilcoxon. Par simplicité, écartons tout problème de normalisation des données et ne considérons que les versions par défaut des tests dans R : t.test(cell.line.ctrl, cell.line.trt) et wilcox.test(cell.line.ctrl, cell.line.trt).

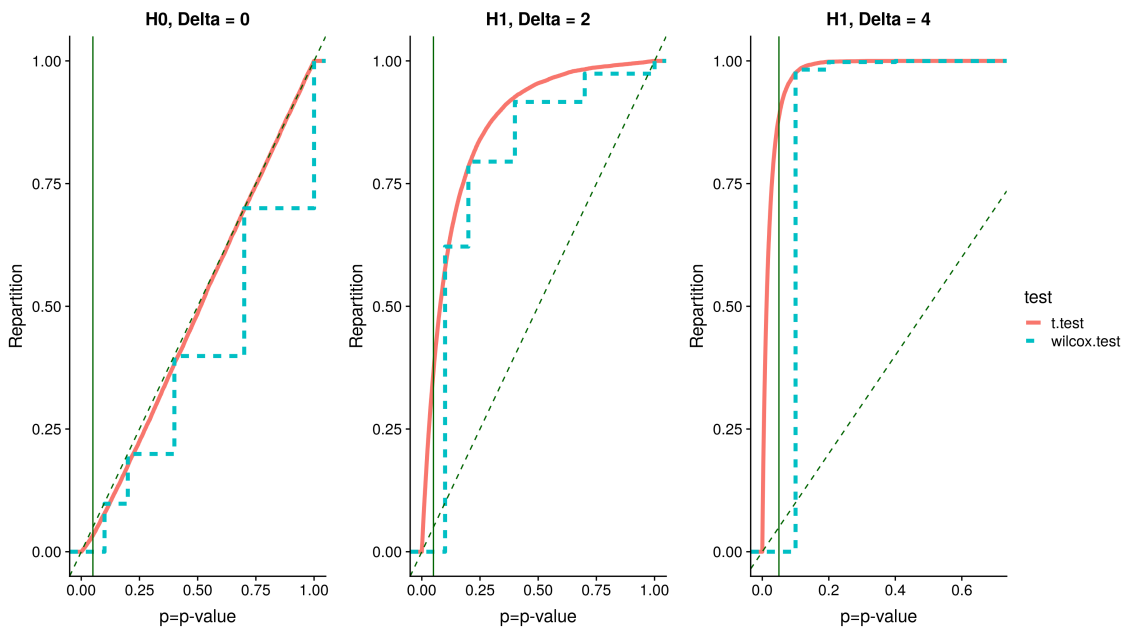

FIGURE 4.1 – Fonctions de répartition des p-valeurs d'un test de Student et de Wilcoxon pour 10^4 jeux de données simulés avec le modèle (4.1) avec $n = 3$, $\delta = 0, 2$ ou 4. La ligne $y = x$ est représentée par une ligne verte pointillée. La ligne verte verticale représente le seuil de 5%.

Statistiquement, notre choix doit être guidé par la capacité de ces deux tests à

— détecter de vraies différences : la puissance ;

— ne pas abusivement identifier une différence s'il n'y en a pas : le risque de première espèce.

Je propose d'étudier ces quantités par simulation.

Simulations statistiques. Voilà comment nous pourrions simuler notre jeu de données avec R :

```
n <- 3; delta=2;
cell.line.trt  <- rnorm(n) + delta
cell.line.ctrl <- rnorm(n)
```

n est le nombre de réplicats et delta (δ) est la différence d'expression entre les deux conditions. cell.line.trt et cell.line.ctrl contiennent 3 observations simulées avec une loi normale. Cela correspond au modèle statistique que voici :

$$X_{ci} = \mu_c + \varepsilon_{ci}, \qquad \varepsilon_{ci} \sim \mathcal{N}(0, 1) \quad \text{i.i.d.} \tag{4.1}$$

X_{ci} est l'expression du gène A dans la condition c (1 : traité ou 2 : non-traité/contrôle) et le réplicat i (1, 2 ou 3). Le bruit est gaussien et de variance 1. La différence d'expression moyenne entre les deux conditions est $\mu_1 - \mu_2 = \delta$. Répéter un grand nombre de fois cette simulation permet d'étudier la fonction de répartition des p-valeurs du test de Student et de Wilcoxon. Les résultats pour $\delta = 0, 2$ ou 4 sont sur la figure 4.1.

Contrôle du risque de première espèce. On voit à gauche sur la figure 4.1 que les deux tests contrôlent le risque de première espèce. En effet pour $\delta = 0$ la fonction de répartition des p-valeurs est sous la diagonale représentant la loi uniforme. En particulier, on a bien moins de 5% de chance d'obtenir une p-valeur inférieure à 5%.

Puissance des deux tests. On peut voir sur la figure 4.1, au milieu et à droite, que pour le test de Wilcoxon la probabilité d'obtenir une p-valeur inférieure à 5% est nulle pour $\delta = 2$ ou 4. La puissance est nulle. Autrement dit on ne peut pas détecter de différence à ce seuil. Pour le même seuil le test de Student a une puissance non négligeable. Elle est d'environ 30% pour $\delta = 2$. Bien évidemment, le bruit étant gaussien, il s'agit d'une configuration favorable au test de Student. Le lecteur intéressé trouvera quelques simulations non-gaussiennes dans la section supplémentaire 4.4.2.

Conclusion intermédiaire. Si nous n'avons pas le courage de supposer les données normales, nous allons devoir demander à Maria de réaliser des expériences supplémentaires, car une puissance nulle n'est pas acceptable. Ici, le choix entre Wilcoxon et Student n'est pas sans importance.

Pour des échantillons plus grands, la différence de puissance est moins importante. Le choix entre les deux tests est moins épineux. Ceci est illustré sur la figure supplémentaire 4.4 pour $n = 30$. Revenons au cas $n = 3$.

4.2.1.3 Hypothèses et modélisation

Avons-nous le courage de supposer les données normales ? Regardons précisément les hypothèses du test de Student, puis celles du test de Wilcoxon.

Hypothèses du test de Student. Par défaut, la fonction t.test de R utilise le test de Welch [106]. Sa dérivation mathématique suppose que les observations sont gaussiennes. Les variances ne sont pas supposées égales. Sans entrer dans trop de détails, il n'est pas possible de savoir si ces hypothèses sont vraies. Par ailleurs, il est difficile d'invoquer quelque chose comme le théorème central limite quand $n = 3$.

Sur l'absence d'hypothèses du test de Wilcoxon. Expliquons les principes du test de Wilcoxon à Maria. Pour commencer, nous dirons que contrairement au test de Student, le test de Wilcoxon ne fait pas d'hypothèses sur la distribution des erreurs. Le test ne considère que les rangs. Pour être plus clairs, nous illustrerons et ajouterons que le test donne le même résultat pour les deux jeux de données du tableau 4.1.

	Lignée cellulaire traitée			Lignée cellulaire contrôle			p-value
	x_{11}	x_{12}	x_{13}	x_{21}	x_{22}	x_{23}	
Jeu de données 1	10	10.1	10.2	13	13.1	13.2	0.1
Jeu de données 2	10	11	12	13	14	15	0.1

TABLE 4.1 – Le test de Wilcoxon ne prend en compte que les rangs. Il donne la même p-valeur sur les deux jeux de données.

À ce moment-là, Maria, qui nous a écoutés attentivement jusque-là, nous interrompra : « le test ne considère donc pas l'intensité du signal ? ». Nous dirons que oui. Puis Maria nous expliquera que la dPCR est, au moins partiellement, quantitative. Sur la page wikipedia sur la dPCR [107] on peut lire :

> It provides absolute quantification because dPCR measures [...]

Conclusion. Au final, les hypothèses du test de Wilcoxon ne paraissent pas plus réalistes que celles du test de Student. Ne pas faire d'hypothèses sur la distribution et considérer seulement les rangs est trop pauvre pour des mesures quantitatives. Les hypothèses du test de Student, à l'inverse, semblent trop fortes. Pour conclure, même dans cette situation simple le choix du test n'est pas évident. Il faut prendre le temps de considérer les spécificités de l'expérience pour prendre une décision.

4.2.2 Modèles statistiques et interprétation biologique

Souvent les paramètres d'un modèle statistique n'ont pas de sens biologique en tant que tels. Pourtant, on leur donne souvent des noms évocateurs comme, le signal, l'effet stress ou l'effet cancer. J'ai souvent constaté, lors d'analyses, qu'il est utile de remettre en cause l'interprétation biologique des paramètres statistiques. Je vais d'abord rapidement évoquer le cas des modèles linéaires qui est bien connu. Je donnerai ensuite un exemple moins classique en détection de ruptures.

4.2.2.1 Un exemple classique, les modèles linéaires

Quand on réalise une analyse différentielle en transcriptomique végétale avec un modèle linéaire on modélise souvent un effet génotype. Cet effet génotype prendra différentes valeurs en fonction du génotype de la plante, par exemple 3 pour le génotype sauvage et -2 pour le mutant. Ces valeurs dépendent des autres paramètres du modèle : l'ordonnée à l'origine, l'effet condition de culture, etc. On ne peut les interpréter telles que. Informellement, on ne comparera pas les valeurs des effets mutant et sauvage mais les estimations du modèle pour un mutant et un sauvage. Statistiquement, on parle de contrastes. Ce sont des combinaisons linéaires particulières des paramètres du modèle. De manière générale, déterminer le contraste approprié à la question biologique est loin d'être trivial. Les habitués des analyses différentielles le savent bien.

L'analyse et l'interprétation se compliquent encore quand on considère des modèles linéaires généralisés ou pénalisés. Dans ce dernier cas, par exemple, l'encodage des variables catégorielles est important [8]. Pour conclure, les paramètres d'un modèle linéaire n'ont en général pas de sens biologique en tant que tels.

4.2.2.2 Un exemple moins classique, ruptures statistiques et ruptures biologiques

J'ai longtemps travaillé sur les modèles de détection de ruptures (voir le chapitre 3). Bien souvent, la modélisation statistique du signal biologique semble aller de soi ; il ne semble pas y avoir d'arbitraire dans le choix des paramètres. En particulier, l'interprétation biologique des ruptures semble naturelle. Cependant dans des modèles de détection de ruptures un peu complexes l'interprétation n'est pas si évidente. Je vais l'illustrer sur un exemple.

Les données. Avec un certain nombre de collègues je travaille sur la détection et l'analyse de transcrits dans des génomes mitochondriaux et chloroplastiques. Chloroplastes et mitochondries sont le résultat de l'endosymbiose d'une cellule bactérienne par une cellule eucaryote. Autrement dit, des cellules eucaryotes ont intégré des bactéries et ces bactéries sont devenues des chloroplastes et des mitochondries. De ce fait, les génomes chloroplastiques et mitochondriaux sont proches de ceux de bactéries. Aussi, je me suis intéressé au travail [91] de Pierre Nicolas et ses co-auteurs pour étudier le paysage transcriptomique des bactéries.

Dans ce travail, ils modélisent le niveau d'expression de cellules bactériennes à l'aide d'un modèle de Markov caché. Ils analysent des données de puces tiling avec une résolution inférieure à 25 paires de bases. La puce contient plusieurs centaines de milliers de sondes que l'on ordonne en fonction de leur position sur le chromosome bactérien. Pour chaque sonde t on dispose d'une mesure d'expression ARN, Y_t, et d'une mesure d'ADN génomique, r_t, utilisée dans le modèle pour corriger un effet sonde. La figure 4.2 illustre une portion de profil analysé.

Un modèle de Markov caché pour la détection de transcrits chez les bactéries. Je décris, ci-dessous une version simplifiée du modèle proposé dans [91]. Par simplicité je ne donne pas toutes les écritures mathématiques du modèle de Markov.

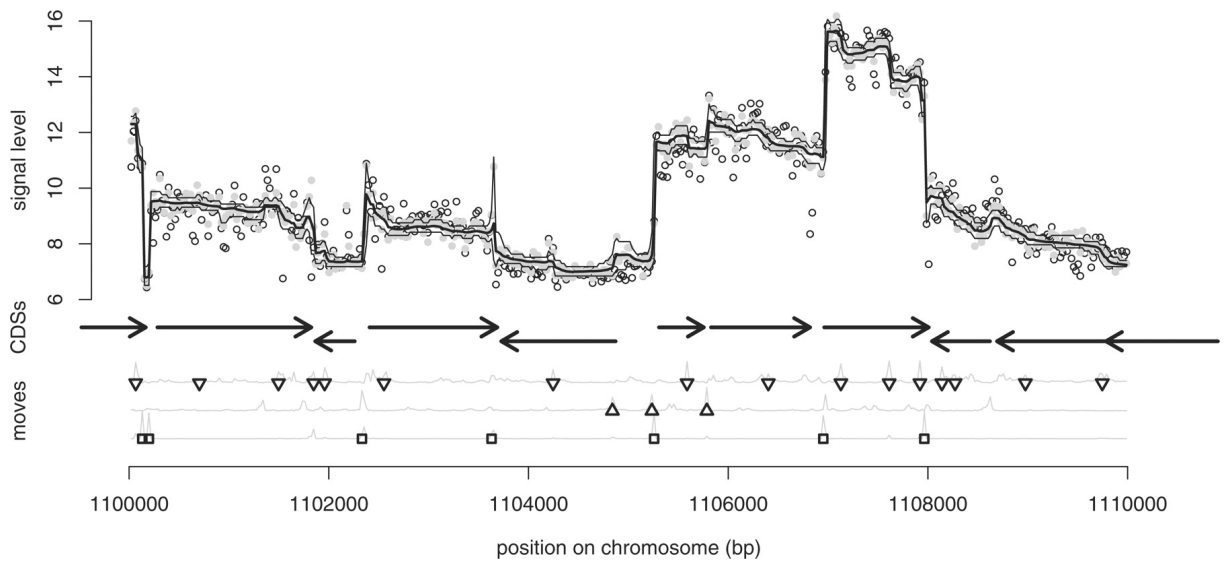

FIGURE 4.2 – Figure 3 de [91]. Je traduis ici la description du papier. Analyse du signal sur le brin (+) d'un segment de 10000 pb du chromosome de *Bacillus subtilis*. Partie supérieure : les cercles ouverts indiquent le signal d'origine. Les cercles gris fermés représentent le signal après « correction » avec la covariable d'ADN, r_t. La ligne noire épaisse montre l'estimation du niveau de transcription calculé avec le modèle de Markov caché. De fines lignes noires correspondent à l'intervalle de confiance à 95%. Partie centrale : les flèches horizontales indiquent les CDS (« coding sequences ») GenBank. Partie inférieure : les grandes transitions le long de la trajectoire la plus probable sont représentées par des carrés. Les petites transitions, de dérive, vers le haut et vers le bas sont indiquées respectivement par des triangles pointant vers le haut et vers le bas. Les probabilités de transitions sont représentées par des lignes grises.

L'espérance du signal Y_t est décrite comme la somme d'un signal u_t modélisé par une chaîne de Markov et d'un effet sonde ρr_t. Si les Y_t sont supposés gaussiens on obtient :

$$Y_t \sim \mathcal{N}(u_t + \rho r_t, \sigma^2) \quad i.i.d. \tag{4.2}$$

Dans le modèle complet de [91] ρ et σ dépendent de u_t ce qui complexifie les écritures et l'inférence. Par ailleurs, la distribution n'est pas gaussienne mais un mélange entre une gaussienne et une uniforme. La loi uniforme modélise la présence de points aberrants ou *outliers*.

Revenons sur la modélisation de u_t. Le signal u_t est modélisé par une chaîne de Markov. La modélisation de u_t est assez proche de celle de μ_t dans 3.3.2.3 à la page 26. Il y a deux différences principales.

— Premièrement, u_t est discrétisé. Cela permet d'utiliser l'algorithme de Viterbi classique pour l'inférence. En pratique 100 états équidistants sont utilisés dans [91].

— Deuxièmement, u_t est soumis à deux types de transitions. Il y a de grandes transitions correspondant aux ruptures de u_t et de petites transitions dites de dérive. Ces petites transitions modélisent de petites variations lisses de l'intensité du signal ; le signal n'est pas exactement constant par morceaux.

Interprétation biologique du modèle. Ce modèle vise une estimation fine du niveau de transcription le long du génome. Il doit faciliter l'interprétation des données et notamment l'identification de transcrits. [91] montre chez *Bacillus subtilis* que les transitions ou ruptures inférées avec le modèle sont proches des promoteurs et terminateurs prédits bioinformatiquement. Elles sont même plus proches que celles identifiées par des modèles de détection de ruptures simples.

Les résultats sont présentés dans la figure 5 de [91]. J'ai étudié en détail cette expérience, car je teste une modélisation alternative avec des collègues. Ce n'est pas mon propos ici. Je vais me concentrer

sur l'interprétation et l'exploitation du modèle de [91]. Je remercie Pierre Nicolas de m'avoir fourni ses codes R.

Commençons par quelques mots sur les promoteurs et les terminateurs. Un promoteur correspond en gros au début d'un gène. Au niveau de ce promoteur on s'attend donc à une augmentation du niveau de transcription : une transition vers le haut. Les terminateurs sont en quelque sorte des fins de gènes. Au niveau des terminateurs on s'attend donc à une baisse du niveau de transcription : une transition vers le bas.

Sur un jeu de données, on peut vérifier et mesurer ces augmentations et baisses du niveau de transcription. J'explique comment les auteurs de [91] le font pour les promoteurs ci-dessous. Par symétrie on peut définir des mesures équivalentes pour les terminateurs.

Mesure d'adéquation des transitions avec les promoteurs. Considérons pour l'instant une seule sonde t de la puce tiling.

— D'une part, on peut vérifier s'il y a ou non un promoteur à proximité (à moins de 22 paires de bases).

— D'autre part, on peut tester si le modèle infère une transition au niveau de la sonde : $u_t \neq u_{t+1}$. Pour savoir si le signal va vers le haut on regardera si la différence $d_t = \hat{Y}_{t+1} - \hat{Y}_t$ est strictement supérieure à 0.

On pourra alors compter le nombre de sondes avec une transition vers le haut qui sont proches d'un promoteur. [91] raffine cette mesure. Il compte le nombre, $R(\delta)$, de sondes avec une transition ($u_t \neq u_{t+1}$) et une différence d_t supérieure à un seuil δ donné. Puis parmi ces sondes il compte celles proches d'un promoteur : $M(\delta)$.

La figure 4.3 en haut, trace $M(\delta)$ en fonction de $R(\delta)$ pour diverses méthodes d'analyse. Dans la figure l'axe $R(\delta)$ est interprété comme un nombre de ruptures inférées au seuil δ et l'axe $M(\delta)$ comme un nombre de correspondances entre ruptures et promoteurs au seuil δ. Sur la figure, une méthode A est meilleure qu'une méthode B si la courbe A est au-dessus de la courbe B. Dans ce cas, pour le même nombre de ruptures inférées on a toujours plus de correspondances avec les promoteurs.

Les résultats du modèle de Markov avec dérive sont les meilleurs pour les promoteurs (courbe noire pleine en haut à gauche de la figure 4.3). Pour les terminateurs, ce n'est malheureusement pas le cas (voir en haut à droite de la figure 4.3). En termes de modélisation, le résultat sur les promoteurs est très satisfaisant. Il justifie bien la prise en compte de dérive dans le modèle.

Nouvelle interprétation biologique du modèle. Rechercher des promoteurs ou terminateurs là où le modèle infère des transitions ($u_t \neq u_{t+1}$) est naturel. La figure 5 de [91] (reproduite sur la figure 4.3 en haut) montre bien que les transitions les plus fortes ont tendance à être des promoteurs ou des terminateurs. Dans le cas des promoteurs on améliore même les performances par rapport à des modèles plus simples. Pourtant en utilisant le même modèle on peut faire encore mieux.

Initialement le mauvais résultat sur les terminateurs m'a surpris, d'autant plus que les prédictions de $\hat{Y}_t$ sur la figure 4.2 sont visuellement satisfaisantes. Si l'on revient sur le modèle, la prédiction $\hat{Y}_t$ est obtenue avec le calcul

$$\hat{Y}_t = \hat{u}_t + \hat{\rho} r_t$$

A cause de l'effet sonde ρr_t, les variations de Y_t ne sont pas identiques à celles de u_t. On a presque toujours $r_t \neq r_{t+1}$, aussi, même si $\hat{u}_t$ est égal à $\hat{u}_{t+1}$, $\hat{Y}_t$ est souvent différent de $\hat{Y}_{t+1}$. D'ailleurs, l'écart entre $\hat{Y}_t$ et $\hat{Y}_{t+1}$ est parfois assez grand même quand $\hat{u}_t = \hat{u}_{t+1}$. Un grand écart n'est-il pas le signe d'un promoteur ou d'un terminateur ?

Dans le calcul de $R(\delta)$ et $M(\delta)$, j'ai fini par considérer toutes les sondes où la différence $d_t = \hat{Y}_{t+1} - \hat{Y}_t$ était supérieure à δ et pas seulement celle où u_t changeait également. Comme on peut le voir en comparant

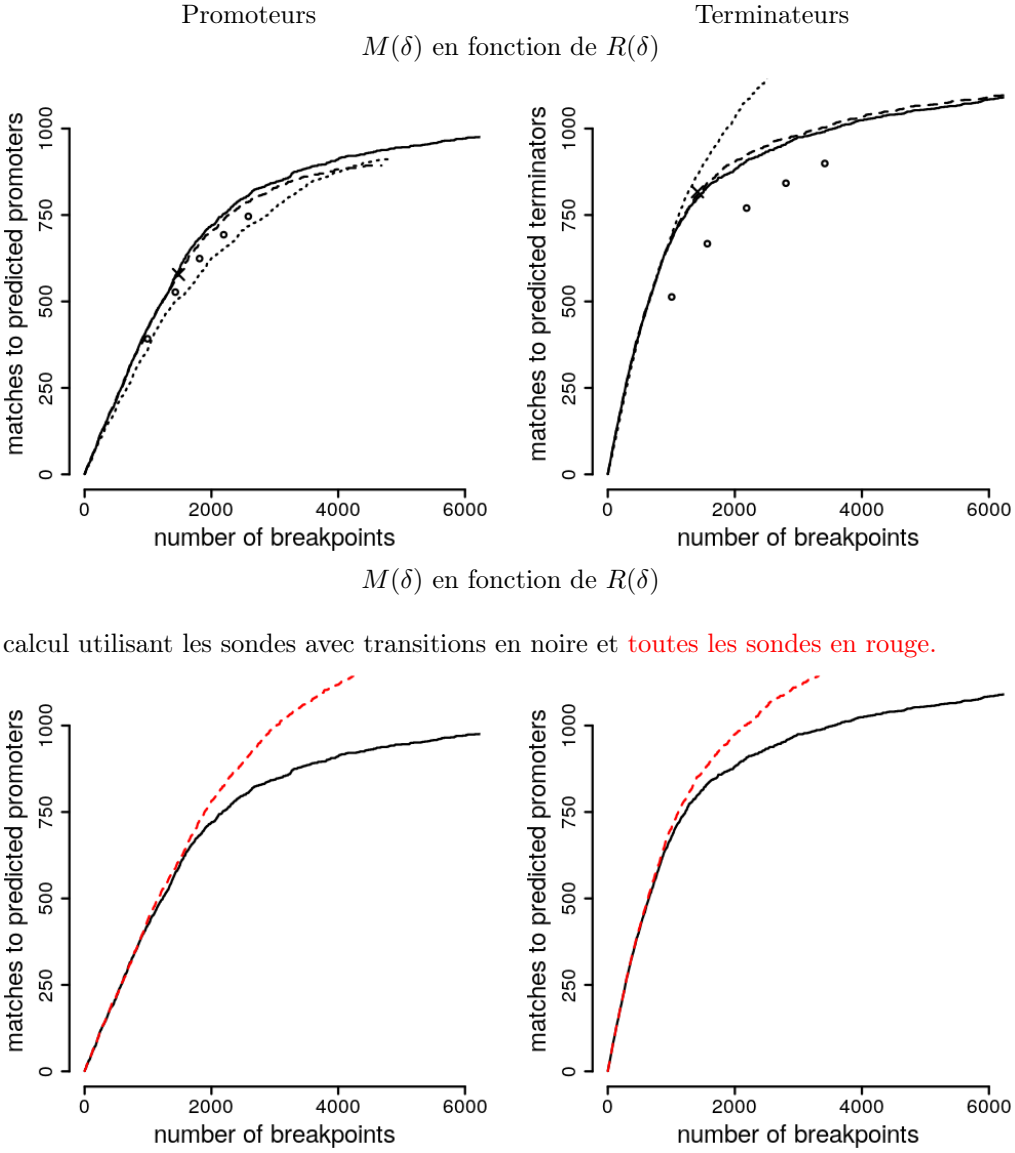

FIGURE 4.3 – En haut : Figure 5 de [91]. Nombre de transitions/ruptures correspondant aux promoteurs (à gauche) et terminateurs (à droite) prédits $M(\delta)$ en fonction du nombre de transitions inférées $R(\delta)$. Les lignes pleines, en tirets et en pointillés montrent les résultats obtenus avec la nouvelle méthode de [91], respectivement, avec le modèle complet, le modèle sans petites transitions et le modèle sans petites transitions et avec $\rho = 0$. Les cercles ouverts ($\circ$) indiquent le résultat pour un modèle de détection de ruptures constant par morceaux pour un nombre de ruptures de 1000, 1500, 2000, 2500 et 3000. En bas : Reprise de la figure 5 de [91]. Les lignes noires pleines sont identiques à celles des figures du haut. Les lignes rouges en pointillées tracent, pour le modèle complet, $M(\delta)$ en fonction de $R(\delta)$ calculés en utilisant toutes les sondes où la différence $d_t = \hat{Y}_{t+1} - \hat{Y}_t$ est supérieure à δ et pas seulement celles présentant une transition dans u_t : $u_t \neq u_{t+1}$.

les courbes rouges et noires sur la figure 4.3, en bas, cela améliore sensiblement les performances. Par exemple pour 2000 transitions vers le bas prédites on passe de 881 terminateurs à 974 terminateurs. Pour 2000 transitions vers le haut prédites on passe de 719 promoteurs à 781 promoteurs.

Conclusion. Modéliser de petites transitions dans le signal u_t améliore l'accord avec la réalité biologique. Par contre les transitions de $\hat{u}_t$ ne capturent pas idéalement cette réalité. Il est préférable de regarder une résultante : $\hat{Y}_t = \hat{u}_t + \hat{\rho} r_t$. Les ruptures statistiques ne correspondent pas parfaitement aux ruptures biologiques.

Je ne tenterai pas d'expliquer ce résultat. Nous obtenons des conclusions similaires avec le modèle que nous développons actuellement avec mes collègues. Je pense que Pierre Nicolas et ses collègues ont très bien fait leur travail. Leur modélisation du signal est naturelle et pertinente. Avec cet exemple je veux simplement illustrer qu'entre modélisation statistique et interprétation biologique il y a encore beaucoup de travail. Regarder un estimateur légèrement différent a parfois des conséquences importantes sur l'efficacité de l'analyse. Idéalement, il faudrait réfléchir et tester plusieurs alternatives. Nous n'en avons pas toujours le temps. Pour conclure, méfions-nous, parfois, de la beauté mathématique de nos équations et de nos modèles.

4.3 Interdisciplinarité, perspectives

J'espère que les deux exemples précédents illustrent bien les difficultés de traduction et d'interprétation des statistiques vers la biologie et vice-versa. Analyser un jeu de données omiques n'est pas un simple service de biostatistiques. Les outils statistiques ne sont pas pour autant une solution miraculeuse à l'analyse. Il faut raisonner leur utilisation et apprendre à bien s'en servir.

Analyser des données omiques c'est un art. Personnellement, ma vision de l'analyse a beaucoup changé ces dix dernières années. En sortie de thèse, j'hésitais entre simple service et miracle mathématique. Je sentais bien que ces deux visions opposées étaient réductrices. Elles ne correspondaient pas à mes premières expériences d'analyses de données omiques qui étaient complexes biologiquement et statistiquement. Peut-être, mon échantillon d'analyses n'était-il pas représentatif ? J'ai eu la chance de travailler avec des biostatisticiens, statisticiens, biologistes et bioinformaticiens hors du commun. J'ai pu poser des questions, proposer des réponses, reformuler des questions sereinement. J'ai découvert qu'analyser un jeu de données omiques est difficile.

Pour provoquer un peu les esprits, je dirais qu'il faut bien plus qu'un simple biologiste ou qu'un simple statisticien pour bien le faire. J'ai rencontré quelques personnes qui prétendent ne pas bien connaître la biologie, la bioinformatique et les mathématiques. Elles discutent pourtant facilement avec tous et ont un sens de l'analyse hors du commun. Ce sont des scientifiques accompli(e)s. Loin des projecteurs, elles restent modestes. Bienheureux, les biologistes et les statisticiens qui travaillent avec elles.

Modéliser et douter. Je n'ai pas vraiment de conseils précis à donner sur l'analyse de données omiques. Je pense qu'il faut régulièrement douter de la question biologique et du modèle statistique. N'est-il pas possible de formuler la question plus simplement ? Ne pourrait-on pas utiliser une mesure, un modèle ou une méthode plus simples, plus directs pour observer ou montrer la même chose ?

Cette recherche de simplicité et cet art du doute prennent du temps. On ne peut pas les mettre en œuvre en permanence. Petit à petit, avec chaque nouveau jeu de données, en se confrontant à des biologistes, des bioinformaticiens, des statisticiens, des informaticiens on améliore ses pratiques d'analyse. Progressivement on relève son niveau d'exigence biologique, statistique, bioinformatique et informatique. Ce poème de Boileau dans l'art poétique résume joliment la situation.

> Avant donc que d'écrire, apprenez à penser.
>
> Ce que l'on conçoit bien s'énonce clairement,
> Et les mots pour le dire arrivent aisément.
>
> Hâtez-vous lentement, et sans perdre courage,
> Vingt fois sur le métier remettez votre ouvrage,
> Polissez-le sans cesse, et le repolissez,
> Ajoutez quelquefois, et souvent effacez.

Promouvoir l'analyse de données. Très modestement, j'essaie depuis quelques années de promouvoir cette vision réaliste et sereine de l'analyse de données. Je donne quelques conseils statistiques. Je m'implique dans certaines analyses un peu complexes. Je participe à des réseaux méthodologiques d'animation scientifique comme NETbio [90]. Je continuerai à le faire.

Pour terminer ce chapitre, je tiens à remercier tous les biologistes, bioinformaticiens, biostatisticiens, statisticiens et informaticiens qui ont pris le temps de m'expliquer patiemment leurs idées et d'écouter les miennes.

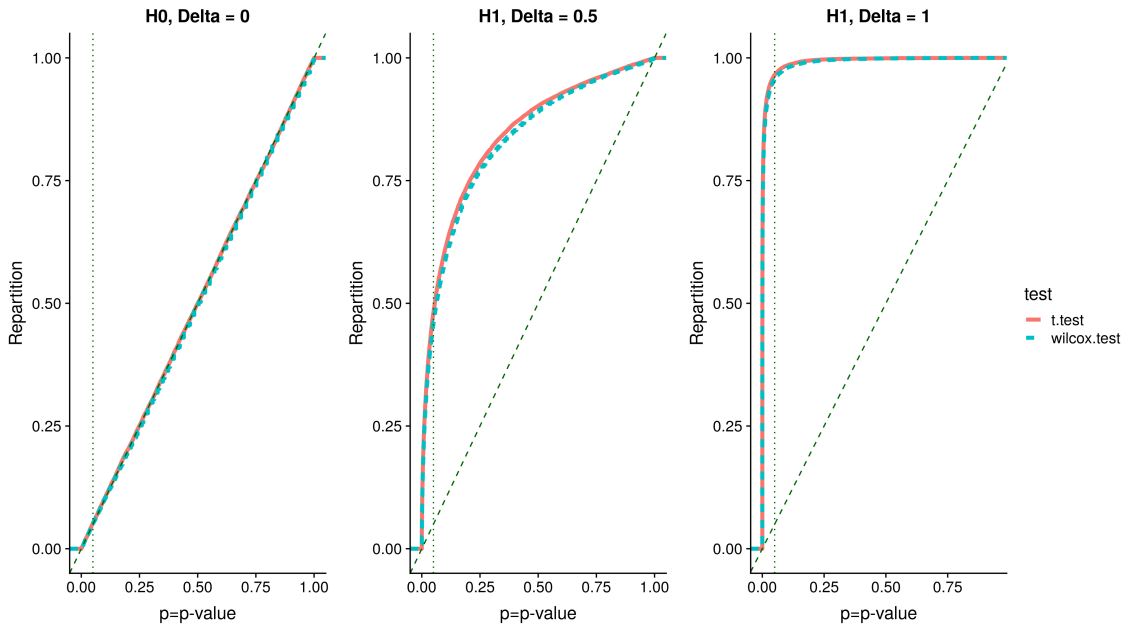

FIGURE 4.4 – Fonctions de répartition des p-valeurs d'un test de Student et de Wilcoxon pour 10^4 jeux de données simulés avec le modèle (4.1) avec $n = 30$, $\delta = 0, 0.5$ ou 1. La ligne $y = x$ est représentée par une ligne verte pointillée. La ligne verte verticale représente le seuil de 5%.

4.4 Quelques figures supplémentaires

Dans cette section se trouvent quelques simulations supplémentaires sur le test de Wilcoxon et de Student (voir la section 4.2.1). Je laisse le lecteur intéressé étudier et interpréter ces figures. Les autres peuvent passer au chapitre suivant à la page 57.

4.4.1 Pour un grand échantillon de 30 éléments

La figure 4.4 présente les fonctions de répartitions obtenues pour 10^4 simulations des p-valeurs du test de Wilcoxon et de Student sous le modèle de l'équation (4.1) avec $n = 30$, $\delta = 0, 0.5$ et 1.

4.4.2 Pour des bruits non-gaussiens et un échantillon de 3 éléments

Les dernières figures du chapitre présentent les fonctions de répartition obtenues pour 10^4 simulations des p-valeurs du test de Wilcoxon et de Student sous le modèle de l'équation (4.1) avec $n = 3$, $\delta = 0, 2$ et 4 et pour des bruits ε_{ci}

— suivant une loi de Student de degré 3 divisé par $\sqrt{3}$ (pour avoir une variance de 1) dans la figure 4.5 ;
— suivant une loi du χ^2 de degré 1 divisé par $\sqrt{2}$ (pour avoir une variance de 1) dans la figure 4.6 ;
— suivant une loi de Cauchy dans la figure 4.7.

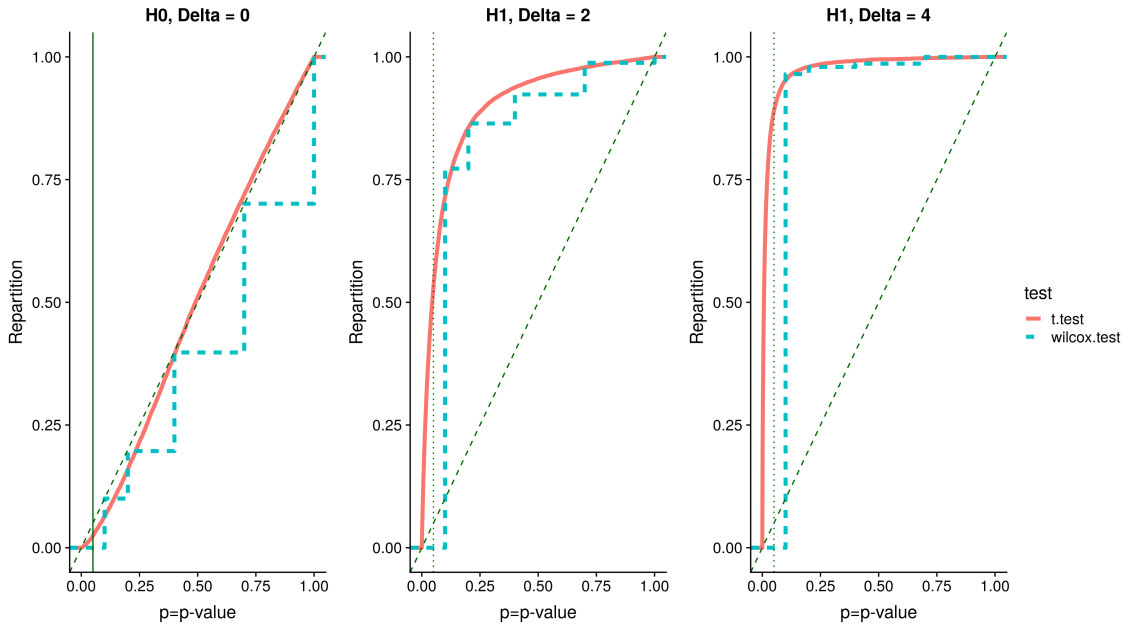

FIGURE 4.5 – Fonctions de répartition des p-valeurs d'un test de Student et de Wilcoxon pour 10^4 jeux de données simulés avec le modèle (4.1) avec $n = 3$, $\delta = 0, 2$ ou 4 et avec un bruit suivant une loi de student de degré 3 divisé par $\sqrt{3}$ plutôt qu'une loi $\mathcal{N}(0, 1)$. La ligne $y = x$ est représentée par une ligne verte pointillée. La ligne verte verticale représente le seuil de 5%.

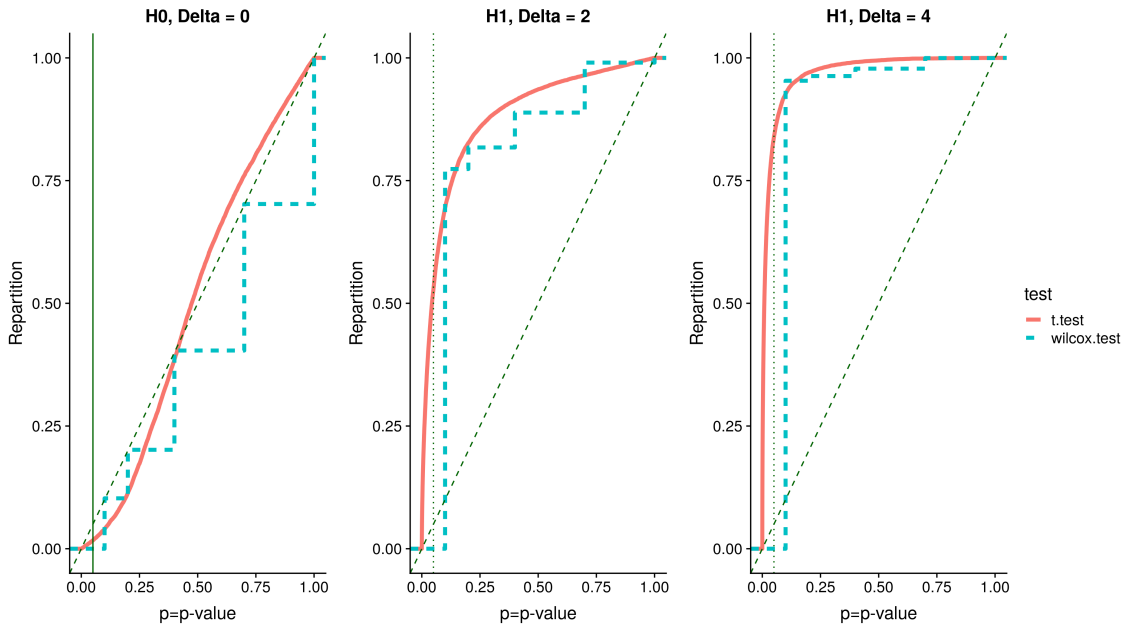

FIGURE 4.6 – Fonctions de répartition des p-valeurs d'un test de Student et de Wilcoxon pour 10^4 jeux de données simulés avec le modèle (4.1) avec $n = 3$, $\delta = 0, 2$ ou 4 et avec un bruit suivant une loi du χ^2 de degré 1 divisé par $\sqrt{2}$ plutôt qu'une loi $\mathcal{N}(0, 1)$. La ligne $y = x$ est représentée par une ligne verte pointillée. La ligne verte verticale représente le seuil de 5%.

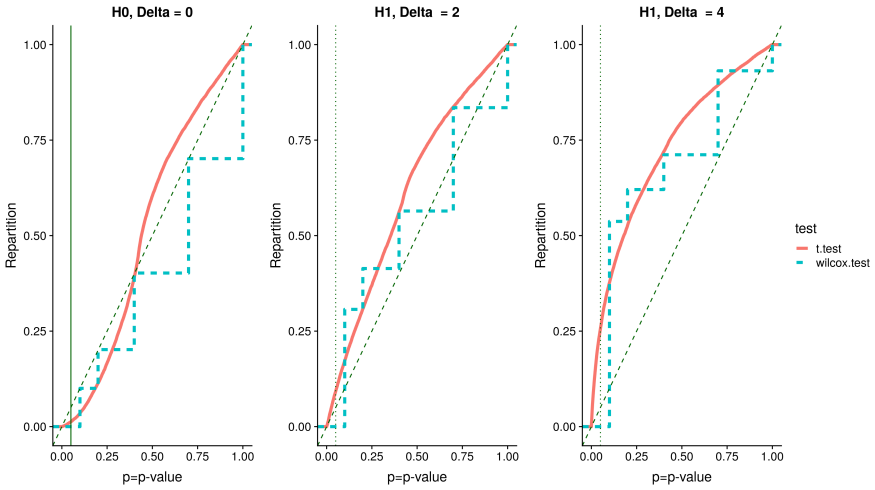

FIGURE 4.7 – Fonctions de répartition des p-valeurs d'un test de Student et de Wilcoxon pour 10^4 jeux de données simulés avec le modèle (4.1) avec $n = 3$, $\delta = 0,2$ ou 4 et avec un bruit suivant une loi de Cauchy plutôt qu'une loi $\mathcal{N}(0,1)$. La ligne $y = x$ est représentée par une ligne verte pointillée. La ligne verte verticale représente le seuil de 5%.

Chapitre 5

Classification régularisée

L'ombre des arbres dans la rivière embrumée
Meurt comme de la fumée,
Tandis qu'en l'air, parmi les ramures réelles,
Se plaignent les tourterelles.

Romance sans paroles - Paul Verlaine

Résumé

Je décris dans ce chapitre quelques contributions méthodologiques sur la classification régularisée.

5.1 Classification et régularisation

Initialement, vers 2012, je me suis intéressé aux méthodes de régression régularisée comme [109] pour l'inférence de réseaux de gènes à partir de données transcriptomiques. En termes de modélisation ces approches ne prennent pas directement en compte une structure de groupes entre les échantillons. Elles supposent les individus indépendants et identiquement distribués. Pour prendre en compte cette structure de groupe et régulariser le problème une idée naturelle est de rajouter une pénalité de fusion [75]. Sur le plan méthodologique, je me suis intéressé dans un premier temps à la classification régularisée indépendamment du problème d'inférence de réseaux.

5.1.1 Classification non-supervisée et relaxation convexe

Nous observons pour un certain nombre d'individus i dans $\{1, ..., n\}$ un vecteur Y_i dans $\mathbb{R}^p$, où p est le nombre de variables mesurées par individu. L'objectif est de trouver une classification de ces n individus en G classes. Pour cela il est classique [75] de chercher la solution du problème suivant :

$$\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^{np}} \left\{ ||Y_i - \beta_i||^2 \right\},$$
$$\text{sous la contrainte } \sum_{i<j\leq n} \mathbb{I}_{\beta_i \neq \beta_j} \leq t,$$

où $\mathbb{I}_{\beta_i \neq \beta_j}$ vaut 1 si $\beta_i \neq \beta_j$ et 0 sinon. Deux individus seront classés dans la même classe si $\hat{\beta}_i = \hat{\beta}_j$.

Ce problème n'est pas convexe. [75] en propose une relaxation en remplaçant l'indicatrice par une norme Ω. Il généralise également le problème en ajoutant un poids w_{ij}. On obtient ainsi le problème dit du « clusterpath » :

$$\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^{np}} \left\{ ||Y_i - \beta_i||^2 \right\},$$
$$\text{sous la contrainte } \sum_{i<j\leq n} w_{ij} \Omega(\beta_i - \beta_j) \leq t.$$

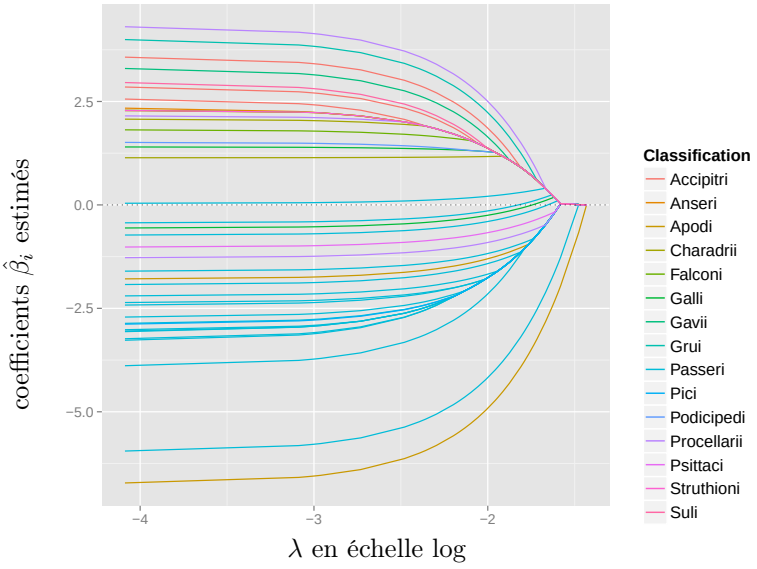

FIGURE 5.1 – Exemple d'arbre de clusterpath sur un jeu de données sur le poids des oiseaux à la naissance. Les poids des oiseaux sont log-transformés. Les branches de l'arbre sont colorées en fonction du sous-ordre taxonomique des oiseaux.

Il revient au même de considérer la formulation Lagrangienne :

$$\hat{\beta}(\lambda) = \arg\min_{\beta \in \mathbb{R}^{np}} \left\{ ||Y_i - \beta_i||^2 \right\} + \lambda \sum_{i<j\leq n} w_{ij}\Omega(\beta_i - \beta_j). \tag{5.1}$$

La valeur de l'estimateur dépend de λ et c'est pour cela que l'on note $\hat{\beta}(\lambda)$. Pour $\lambda = 0$ on a $\hat{\beta}_i(0) = Y_i$. Pour $\lambda = \infty$ tous les $\hat{\beta}_i(\infty)$ sont égaux. [75] étudie les valeurs de $\hat{\beta}_i$ en fonction de λ. Pour la norme ℓ_1 et pour des poids unitaires, $w_{ij} = 1$, les $\hat{\beta}_i$ fusionnent toujours quand λ croît. Mathématiquement, on a la propriété suivante :

$$\text{Si } \hat{\beta}_i(\lambda) = \hat{\beta}_j(\lambda) \text{ alors, pour tout } \delta \geq 0 \quad \hat{\beta}_i(\lambda + \delta) = \hat{\beta}_j(\lambda + \delta). \tag{5.2}$$

Autrement dit, il n'y a pas de fissions. Si i et j sont dans la même classe pour λ ils le seront encore pour $\lambda + \delta$. Si l'on trace tous les $\hat{\beta}_i$ en fonction de λ on obtient un arbre. Je présente un exemple sur un jeu de données sur le poids des oiseaux à la naissance dans la figure 5.1. On peut déduire de cet arbre une structure de classification hiérarchique sur les individus.

[75] dérive un algorithme homotopique en $O(pn\log(n))$ pour obtenir toutes les valeurs de $\hat{\beta}_i$ en fonction de λ. À partir de là les auteurs obtiennent la classe de chaque individu i pour une valeur de λ donnée en seulement $O(pn)$. Si le nombre de variables p est supérieur à 2, pour construire l'arbre et toute la structure hiérarchique entre les individus, il faut réaliser cette dernière opération de l'ordre de n fois. Le coût global est alors de $O(pn^2)$ ce qui est trop important en pratique si n est grand (de l'ordre de 10^4 ou plus).

5.1.2 Quelques contributions

5.1.2.1 Des poids décroissants de la distance, fused-anova

Le problème. Dans [9] nous étudions, avec Julien Chiquet et Pierre Gutierrez, une généralisation du clusterpath que nous appelons fused-anova. En plus des variables du clusterpath, nous nous donnons une partition initiale des données en K groupes. Ces groupes sont donnés par la fonction κ de $\{1, ..., n\}$ dans $\{1, ..., K\}$.

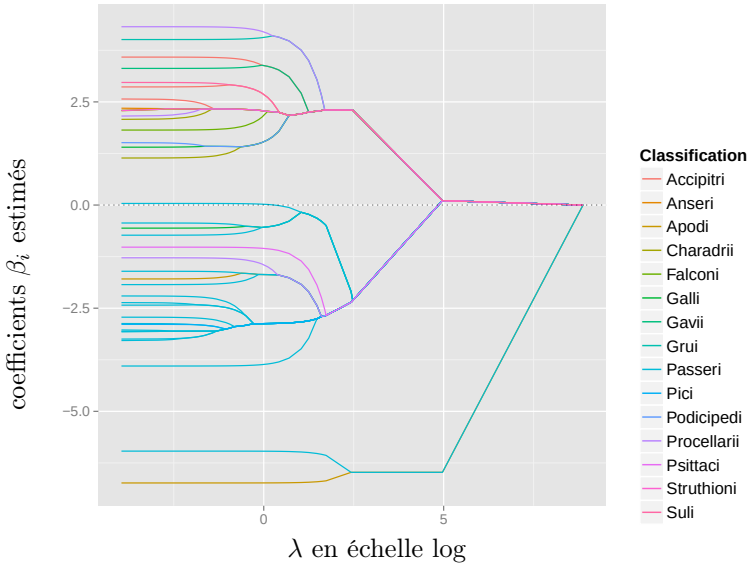

FIGURE 5.2 – Exemple de sortie de fused-anova avec des poids décroissants de la distance sur un jeu de données sur le poids des oiseaux à la naissance. Les poids des oiseaux sont log-transformés. Les branches de l'arbre sont colorées en fonction du sous-ordre taxonomique des oiseaux. Ce sont les mêmes données que sur la figure 5.1.

Dans un contexte de classification non-supervisée, cette partition initiale permet de modéliser des données répétées. Dans ce cadre K sera grand. La partition initiale κ permet également de faire le lien avec l'anova multivariée et l'approche Cas-anova [56] qui est une régularisation de l'analyse de la variance (anova). Dans ce cadre le nombre de groupes K sera souvent assez petit devant n. Le problème fused-anova s'écrit :

$$\min_{\beta \in \mathbb{R}^{Kp}} \left\{ ||Y_i - \beta_{\kappa(i)}||^2 \right\} + \lambda \sum_{k<\ell\leq K} w_{k\ell} \Omega(\beta_k - \beta_\ell). \tag{5.3}$$

On retrouve le problème du clusterpath pour $\kappa(i) = i$ et $K = n$.

Contributions. Nos principales contributions sont les suivantes :

1. Nous étendons le résultat de [75] sur l'absence de fissions
   — à toutes les normes ℓ_q,
   — à la norme ℓ_1 pour des poids w_{ij} décroissants de la distance entre les observations Y_i et Y_j.
2. Pour la norme ℓ_1 et certains poids décroissants de la distance nous dérivons un algorithme homotopique en $O(pn\log(n))$ pour obtenir toutes les valeurs de $\hat{\beta}_i$ en fonction de λ. Avec Julien Chiquet et Pierre Gutierez nous avons implémenté cet algorithme dans le package R univarclust [43].
3. Statistiquement, dans un cadre anova et pour certains poids décroissants de la distance nous démontrons des propriétés de consistance équivalentes à celles de Cas-anova [56]. Dans les détails, l'estimateur fused-anova est valide pour une plus grande plage de λ que Cas-anova et il se comporte mieux sur des simulations.

La figure 5.2 donne un exemple de structure hiérarchique obtenue avec fused-anova sur le jeu de données utilisé pour la figure 5.1.

5.1.2.2 Agrégation d'arbres

Pour de grandes valeurs de n et un nombre de variables $p \geq 2$ l'algorithme du clusterpath [75] et fused-anova [9] seul ne permettent pas de retrouver efficacement la structure d'arbre et la hiérarchie globale entre les individus i. Le temps de calcul est de $O(pn^2)$.

De manière très schématique clusterpath et fused-anova obtiennent une hiérarchie pour chaque variable en $O(pn\log(n))$. Pour obtenir la hiérarchie globale il faut les agréger. Une stratégie possible consiste à évaluer les classes des individus pour n valeurs de λ. Chaque évaluation est réalisée en $O(pn)$. La complexité globale est alors de $O(pn^2)$.

Cette stratégie n'exploite pas la structure d'arbre des hiérarchies. Dans le cadre de la thèse d'Audrey Hulot, nous avons montré avec Julien Chiquet et Florence Jaffrézic dans [**hulot_inprep**] qu'il était possible de reconstruire ou d'agréger la hiérarchie globale en seulement $O(pn\log(n))$ à partir des hiérarchies de chaque dimension. Pour tout p et pour la perte ℓ_1 on retrouve donc la hiérarchie globale du problème fused-anova pour une complexité réduite à $O(pn\log(n))$ au lieu de $O(pn^2)$. Avec Audrey Hulot et Julien Chiquet, nous avons implémenté l'algorithme dans le package R mergeTree [45]. De manière plus générale ce package permet d'agréger efficacement plusieurs hiérarchies.

5.1.2.3 Miscellanées

Encodage et régularisation. Suite au travail sur fused-anova nous avons montré, avec Julien Chiquet et Yves Grandvalet, que pour les méthodes de régression régularisée intégrant des variables catégorielles l'encodage des variables était important [8]. Par exemple, dans une approche de type anova il est classique de choisir une catégorie comme référence. En l'absence de régularisation les prédictions du modèle ne dépendent pas de ce choix arbitraire. Avec une régularisation de type ℓ_1, ce n'est plus le cas et les prédictions peuvent être très différentes [8].

Mesures de proximité entre classifications. Sur le plan applicatif, il est souvent intéressant de mesurer la proximité entre deux classifications. Une mesure classique est l'« adjusted-rand-index » mais il y en a beaucoup d'autres [100]. De nombreuses implémentations de ces critères calculent une matrice de contingence entre les deux classifications. Si le nombre de groupes est important la matrice de contingence est grande mais aussi très creuse ou sparse. Dans ce cas, le calcul explicite de la matrice est inefficace algorithmiquement. Dans le package aricode [44], développé avec Valentin Dervieux et Julien Chiquet, nous évitons ce calcul. Grâce à un tri préalable des données en $O(n)$ nous obtenons une complexité au pire en $O(n)$.

5.2 Quelques perspectives

Ces travaux ouvrent un certain nombre de pistes de recherche que je détaille brièvement ci-dessous.

Données discrètes. Concernant le modèle fused-anova, nous n'avons considéré que des variables continues et implicitement des données gaussiennes. Pour l'analyse de données de séquencage haut-débit, il faudrait étendre ces résultats à des variables discrètes, notamment pour un modèle de Poisson ou négatif binomial.

Inférence de réseaux de gènes, régression et classification/anova régularisées. Dans le cadre de la thèse de Trung Ha, que j'ai co-encadré avec Marie-Laure Martin-Magniette et Julien Chiquet, nous avons exploré plusieurs pistes pour combiner fused-anova et la régularisation ℓ_1 (classiquement utilisée pour l'inférence de réseaux de gènes). Certaines pistes sont prometteuses. Prendre en compte le réseau de gènes permet dans certaines situations d'améliorer la détection de gènes différentiellement exprimés [71].

Classification et stabilité. Les mesures de proximité entre classifications sont souvent utilisées pour évaluer la stabilité d'une classification et déterminer le nombre de groupes [101]. Ces approches sont souvent coûteuses algorithmiquement, car elles impliquent de calculer un très grand nombre de classifications sur des échantillons bootstraps et nécessitent de comparer ces classifications. En combinant des idées d'aricode [44] et d'agrégation d'arbre [**hulot_inprep**] il semble possible d'accélérer les calculs.

Par ailleurs, les mesures de proximité entre classifications sont encore peu étudiées d'un point de vue statistique. Les calculs proposés ne prennent pas toujours en compte le caractère aléatoire des individus échantillonnés et les possibles erreurs de classification [100]. Martina Sundqvist, en thèse avec Julien Chiquet, Thierry Dubois et moi-même, travaille sur le sujet.

Chapitre 6

Évaluation de méthodologies omiques

> La filosofia è scritta in questo grandissimo libro, che continuamente ci sta aperto innanzi agli occhi (io dico l'Universo), ma non si può intendere, se prima non il sapere a intender la lingua, e conoscer i caratteri ne quali è scritto. Egli è scritto in lingua matematica, e i caratteri son triangoli, cerchi ed altre figure geometriche, senza i quali mezzi è impossibile intenderne umanamente parola ; senza questi è un aggirarsi vanamente per un oscuro labirinto.
>
> Il Saggiatore, cap. 6 - Gallileo

Résumé

Certains problèmes d'analyse de données omiques sont particulièrement fréquents : par exemple l'analyse différentielle de données transcriptomiques ou encore la détection d'altérations du nombre de copies d'ADN. La majorité des outils pour réaliser ces analyses dépend d'un très grand nombre de paramètres et d'hypothèses explicites ou implicites. Il est en général difficile de juger de l'importance pratique de ces paramètres et hypothèses. Malgré tout il faut choisir une approche pour faire l'analyse. Je décris succinctement dans ce chapitre quelques travaux visant à éclairer ce choix.

6.1 Évaluation, simulations statistiques et expériences biologiques

Quand on analyse un jeu de données omiques il faut choisir une méthode d'analyse. Idéalement, nous aimerions pouvoir justifier notre choix. Comme je l'ai expliqué dans le chapitre 4 je pense qu'il faut, autant que possible, viser l'adéquation entre la question biologique et les hypothèses de la méthode utilisée. Toutefois, pour des analyses complexes il est souvent difficile de bien mesurer cette adéquation. Il y a plusieurs raisons à cela. Tout d'abord, il n'est pas toujours facile de bien juger de l'importance et de la robustesse de certaines hypothèses. C'est le cas notamment de l'hypothèse de normalité dans les données de nombre de copies d'ADN. Par ailleurs, certaines étapes de l'analyse exploitent des heuristiques et approximations statistiques ou calculatoires dont il est difficile de bien comprendre la portée et la pertinence biologique. Je pense par exemple à la normalisation des données RNA-seq.

Il y a aussi un certain nombre de difficultés liées à des biais de publications développées dans [57]. Je n'en parlerai pas beaucoup plus dans le reste du chapitre, mais on peut constater qu'il y a finalement assez peu d'articles focalisés sur la comparaison de méthodes. La majorité des publications parle du développement de nouvelles méthodologies et on peut, parfois, douter de la neutralité des comparaisons présentées dans ces publications. Anne-Laure Boulesteix, Sabine Lauer et Manuel Eugster dans [57] mettent en avant le besoin de comparaisons neutres et en identifient trois caractéristiques importantes :

1. La publication doit porter sur la comparaison en elle-même.
2. Les auteurs doivent être raisonnablement neutres.

3. Les jeux de données et les critères d'évaluation doivent être choisis de manière rationnelle.

Il ne me semble pas toujours simple de répondre à ces trois critères simultanément. Il faut toutefois essayer.

Revenons à l'évaluation de méthodologies omiques. Supposons que nous ayons deux méthodologies à comparer. Comment pourrions nous évaluer leurs performances ? Les statistiques et la biologie nous donnent quelques réponses.

Statistiquement, nous pouvons faire des simulations. Sur un jeu de données simulées il est facile d'évaluer les performances car la vérité est connue. On peut répéter le processus un très grand nombre de fois et explorer de nombreuses configurations. Il est toutefois clair que ces simulations ne peuvent rivaliser avec la complexité des données biologiques. En tous cas, les simulations ne mesurent pas l'écart entre le modèle et les données. De bonnes performances sur des données simulées ne sont pas le gage de bonnes performances pratiques. Néanmoins, il ne faut pas écarter trop vite ces simulations, car il est douteux qu'une méthode qui fonctionne mal sur des simulations simples soit performante sur des données biologiques complexes.

Biologiquement, nous pouvons valider expérimentalement les résultats des deux méthodologies avec d'autres techniques de biologie moléculaire comme la qRT-PCR. Ici, il n'y a pas à douter du réalisme des données. Néanmoins, les validations sont souvent coûteuses et prennent beaucoup de temps. Par conséquent elles sont souvent peu nombreuses et nous pouvons douter de leur représentativité. Les conclusions ne sont pas forcément simples à généraliser.

Pour résumer, les validations statistiques peuvent être nombreuses mais sont peu réalistes et les validations expérimentales sont réalistes mais peu nombreuses. En utilisant des techniques de visualisation ou de ré-échantillonage il est possible d'obtenir des jeux de données plus réalistes que des simulations statistiques simples où la vérité est partiellement connue. C'est l'objet des techniques dont je vais parler dans ce chapitre.

6.2 Annotation de profils génomiques

Pour l'analyse de profils de nombre de copies d'ADN ou de Chip-Seq les bio-analystes ont l'habitude de visualiser les données le long du génome. Ils jugent des performances d'une méthode en vérifiant si les ruptures identifiées sont plausibles à leurs yeux. Pour automatiser ce processus, il est assez naturel d'annoter les profils en indiquant des intervalles du génome où l'on pense qu'il y a des ruptures et des intervalles où l'on pense qu'il n'y en a pas. La figure 6.1 donne un exemple sur des données ChIP-seq.

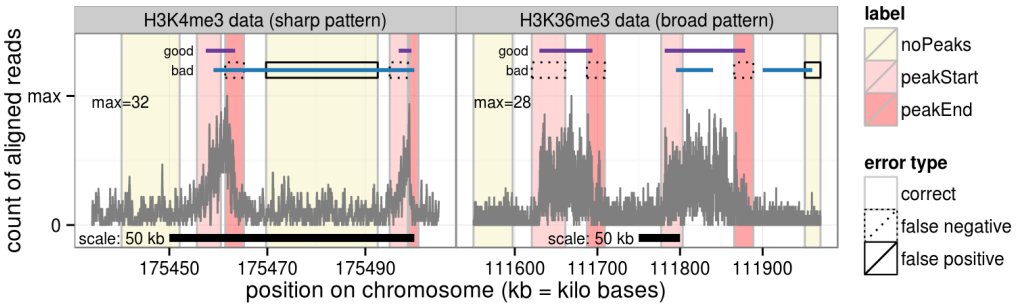

FIGURE 6.1 – Figure tirée de [17]. Exemple de profil ChIP-seq annoté. Il y a trois types d'annotations : absence de pic (jaune : noPeaks), début de pic (rose : peakStart) et fin de pic (rouge : peakEnd). On souhaite identifier des pics (représentés par des barres horizontales) cohérents avec l'annotation (violet : good) et on ne veut pas de pics incohérents avec les annotations (bleu : bad). Les pics à gauche sont plutôt courts (sharp pattern), les pics à droite plus longs (broad pattern).

On peut évaluer si une méthode d'analyse est en accord avec les annotations en comparant pour chaque intervalle le nombre de ruptures annotées et le nombre de ruptures identifiées. Par exemple, si dans un intervalle l'annotation n'indique aucune rupture et qu'une rupture y est détectée on compte une erreur. Cette idée, inspirée des approches de vision par ordinateur, a été proposée par Toby Hocking pour les données CGH (Comparative Genomic Hybridization en anglais) [76]. Le désaccord entre une méthode d'analyse et l'annotation est appelé dans cet article l'erreur d'annotation. On peut ramener cette erreur à un pourcentage d'annotations mal inférées.

6.2.1 Annotations, quelques inquiétudes statistiques

En tant que statisticien on peut douter de la qualité des annotations et s'inquiéter de discordances entre annotateurs. Pourquoi se fier à l'œil humain ? Dans [76] il est montré que les annotateurs sont raisonnablement cohérents. Par ailleurs, on peut tester que sur des données simulées les annotateurs ne se trompent pas [18].

On peut aussi craindre que les annotations soient dans des zones faciles, des profils génomiques ne permettant pas de discriminer les méthodes d'analyse. C'est une crainte respectable, mais elle n'est pas justifiée empiriquement : dans [76] les taux d'erreur des méthodes varient entre 2% et 83%.

6.2.2 Annotations et calibration

Le papier [76] va plus loin que la simple évaluation des méthodes. Il propose de calibrer les approches sur la base des annotations. L'idée est simple : trouver les valeurs des paramètres qui minimisent l'erreur d'annotation. Plus précisément, étant donné un profil Y, des annotations A_Y et une segmentation $S(Y, \lambda)$ obtenue avec le paramètre λ, on peut calculer l'erreur d'annotation en fonction de λ : $Err(A_Y, S(Y, \lambda))$. Sur un jeu de données de P profils $Y^{(1)}, ...Y^{(P)}$ l'objectif est de trouver le λ minimisant la somme des erreurs sur les P profils :

$$\arg\min_{\lambda} \left\{ \sum_{p} Err(A_{Y^{(p)}}, S(Y^{(p)}, \lambda)) \right\}. \tag{6.1}$$

Si λ est univarié on résout le problème assez simplement en calculant l'erreur sur une grille de valeurs de λ. On définit ainsi un problème d'apprentissage. Pour bien faire les choses on peut découper le jeu de données de P profils en un jeu d'apprentissage et un jeu test. On apprend alors λ en minimisant l'erreur sur le jeu d'apprentissage. Puis on évalue les performances de ce λ sur le jeu test. On peut également faire de la cross-validation. Je ne rentrerai pas plus dans les détails qui se trouvent dans [76].

6.3 Quelques contributions

6.3.1 Apprentissage de pénalité

Dans [76] pour les méthodes maximisant la vraisemblance il est proposé d'optimiser le coefficient α d'une pénalité par rupture de la forme αn_p, avec n_p la longueur du profil analysé (voir 3.2.2). Le problème (6.1) se ré-écrit :

$$\arg\min_{\alpha} \left\{ \sum_{p} Err(A_{Y^{(p)}}, S(Y^{(p)}, pen = \alpha K n_p)) \right\}.$$

Les résultats statistiques de [85, 108] suggèrent des pénalités plus complexes. La pénalité dépend notamment de la variance du signal σ_p, de la log longueur $\log(n_p)$ et du nombre de ruptures. Il faut remplacer $\alpha K n_p$ par une fonction f de plusieurs paramètres dont les coefficients β ne sont pas connus :

$$\arg\min_{\beta} \left\{ \sum_{p} Err(A_{Y^{(P)}}, S(Y^{(p)}, pen = f(\beta, n_p, \sigma_p, K, ...)) \right\}.$$

L'erreur d'annotation n'étant pas convexe ou dérivable il n'est pas évident de résoudre le problème si β n'est pas univarié, car une recherche par grille est rapidement coûteuse. Nous proposons d'utiliser une approximation convexe de l'erreur d'annotation. Nous obtenons alors un problème d'apprentissage plus classique que nous résolvons avec des techniques d'optimisation assez standard. Nous montrons qu'apprendre cette fonction f et ses coefficients β permet d'améliorer l'erreur d'annotation par rapport à [76].

Approximation convexe de l'erreur d'annotation. Pour un profil Y l'erreur d'annotation est une fonction constante par morceau de la pénalité pen : $Err(A_Y, S(Y, pen))$. Très schématiquement, voilà comment nous en obtenons une approximation convexe. Nous identifions l'intervalle de pénalités pen sur lequel l'erreur est minimale, $[\underline{\mathrm{L}}_Y, \bar{L}_Y]$ et remplaçons l'erreur d'annotation par une fonction constante sur cet intervalle et quadratique au delà de cet intervalle. Dans les détails c'est un peu plus complexe, nous définissons :

$$Approx(A_Y, S(Y, pen)) = \phi\left(\frac{pen - \underline{\mathrm{L}}_Y}{\delta}\right) + \phi\left(\frac{pen - \bar{L}_Y}{\delta}\right),$$

où ϕ est la fonction $x \to (x-1)^2$ et δ une constante donnée.

6.3.2 Ré-échantillonage de profils de nombre de copies d'ADN

L'utilisation d'annotations sur les ruptures n'est pas complètement satisfaisante sur le plan statistique. Notamment, il n'est pas possible de simuler de nouveaux profils avec des configurations différentes en termes de niveau de bruit ou de longueur des segments. Dans [10] plutôt que d'annoter la présence ou l'absence de ruptures dans un intervalle du profil génomique, nous proposons d'annoter le nombre de copies d'ADN de l'intervalle.

À partir de ces annotations nous simulons facilement de nouveaux profils. Par exemple pour simuler un segment de 100 points et deux copies d'ADN nous tirons au hasard 100 points de régions annotées deux copies. Il suffit ensuite de combiner plusieurs segments pour obtenir un profil complet. Ce processus de simulation nous permet de comparer diverses méthodes dans un grand nombre de configurations et d'identifier les avantages et inconvénients de chacune. Un package R jointseg implémente ce processus de simulation [46]. Je l'ai utilisé avec Paul Fearnhead pour montrer l'intérêt d'une méthode robuste aux outliers [13]. Cette comparaison ne remplit par le premier critère d'une comparaison neutre [57].

6.3.3 Ré-échantillonage de données transcriptomiques

Le ré-échantillonage d'un jeu de données transcriptomiques pour évaluer une méthode d'analyse différentielle est plus complexe. Pour pouvoir évaluer les performances nous avons besoin :

1. de gènes où nous savons qu'il y a une différence d'expression entre deux conditions expérimentales,
2. de gènes où nous savons qu'il n'y a pas de différence d'expression.

Pour identifier de vraies différences nous ne pouvons pas visualiser les données comme dans le cas des données de nombre de copies d'ADN, nous avons besoin de validations expérimentales (par exemple par qRT-PCR). À la condition d'avoir de telles validations nous pouvons artificiellement intégrer ces gènes véritablement différentiels dans un nouveau jeu de données.

Pour obtenir des gènes où il n'y a aucune différence d'expression nous pouvons utiliser des réplicats biologiques. Par exemple, si nous disposons de quatre réplicats, la comparaison des premier et troisième réplicats au deuxième et quatrième réplicats ne doit donner aucune différence. Là encore nous pouvons artificiellement intégrer ces gènes véritablement non-différentiels dans un nouveau jeu de données.

Dans [11] nous avons proposé de combiner ces deux idées pour obtenir un jeu de données synthétiques. L'idée est représentée très schématiquement sur la figure 6.2. On part d'un jeu de données avec des réplicats et d'un jeu de données très différencié (comparant des boutons floraux et des feuilles). Un certain

nombre des différences du deuxième jeu de données a été validé par qRT-PCR. Il ne reste alors plus qu'à combiner ces deux jeux de données. On peut contrôler la proportion de gènes différenciés en variant les proportions des deux jeux de données initiaux. Cette idée nous a permis d'évaluer différentes méthodes d'analyse différentielle et d'évaluer l'importance de plusieurs paramètres de cette analyse. Beaucoup de travaux se sont intéressés à la modélisation et l'estimation de la surdispersion dans les modèles négatifs binomiaux [86, 93]. Dans notre étude, de manière surprenante sans doute, la modélisation de la moyenne a un impact plus important sur la qualité de l'analyse différentielle que l'estimation de cette surdispersion. Cela confirme ce que je disais au début du chapitre : il n'est pas toujours facile de bien juger de l'importance et de la robustesse de certaines hypothèses.

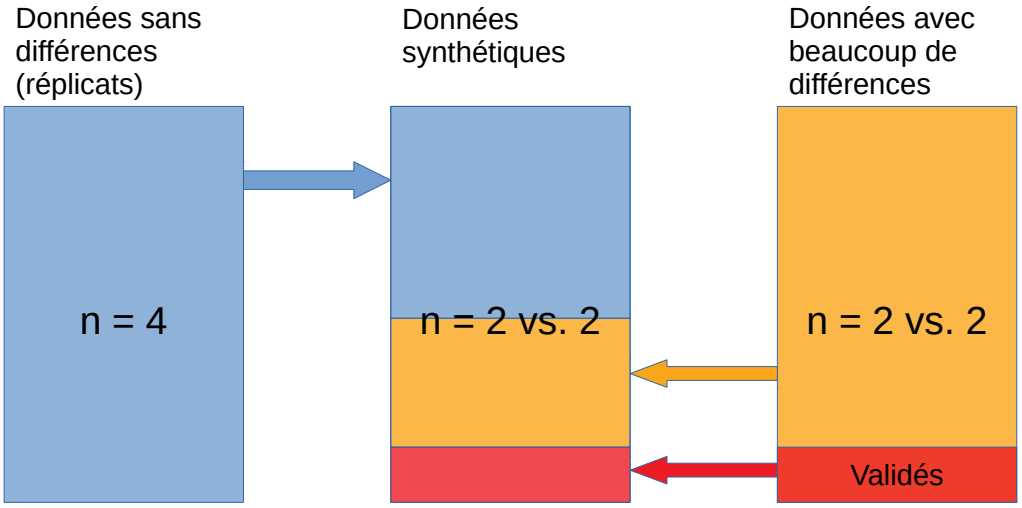

FIGURE 6.2 – Construction d'un jeu de données synthétiques en transcriptomique, vision schématique. Le jeu est obtenu en tirant au hasard des données d'un jeu de données sans différences et d'un jeu de données avec beaucoup de différences dont certaines sont validées expérimentalement. On peut faire varier les proportions des deux jeux de données initiaux.

6.4 Quelques perspectives

L'analyse de données omiques est un problème récurrent en biologie moléculaire. Il semble critique de bien évaluer les performances des outils existants pour les choisir judicieusement lors d'analyses. Au delà de l'analyse, l'évaluation doit aussi guider la modélisation statistique et mathématique des données. À ce titre, pour le développement de modèles de détection de ruptures complexes pour l'analyse de données génomiques, il me semble essentiel de promouvoir l'annotation de profils par des biologistes. J'ai commencé à le faire pour l'analyse de profils mitochondriaux et chloroplastiques.

Références

[1]Guillem Rigaill, P. Hupé, A. Almeida, P. La Rosa, J.-P. Meyniel, C. Decraene et E. Barillot, « ITALICS : an algorithm for normalization and DNA copy number calling for Affymetrix SNP arrays », **Bioinformatics 24**, 768-774 (2008).

[2]F. Picard, E. Lebarbier, M. Hoebeke, Guillem Rigaill, B. Thiam et S. Robin, « Joint segmentation, calling, and normalization of multiple CGH profiles », **Biostatistics** (2011).

[3]Guillem Rigaill, É. Lebarbier et S. Robin, « Exact posterior distributions and model selection criteria for multiple change-point detection problems », **Statistics and Computing 22**, 917-929 (2012).

[4]Guillem 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**, 150-175 (2015).

[5]A. Cleynen, M. Koskas, E. Lebarbier, Guillem Rigaill et S. Robin, « Segmentor3IsBack : an R package for the fast and exact segmentation of Seq-data. », **Algorithms for Molecular Biology 9**, 6 (2014).

[6]A. Cleynen, T. M. Luong, Guillem Rigaill et G. Nuel, « Fast estimation of the Integrated Completed Likelihood criterion for change-point detection problems with applications to Next-Generation Sequencing data », **Signal Processing 98**, 233-242 (2014).

[7]Guillem Rigaill, S. Cadot, R. J. Kluin, Z. Xue, R. Bernards, I. J. Majewski et L. F. Wessels, « A regression model for estimating DNA copy number applied to capture sequencing data », **Bioinformatics 28**, 2357-2365 (2012).

[8]J. Chiquet, Y. Grandvalet et Guillem Rigaill, « On coding effects in regularized categorical regression », **Statistical Modelling 16**, 228-237 (2016).

[9]J. Chiquet, P. Gutierrez et Guillem Rigaill, « Fast tree inference with weighted fusion penalties », **Journal of Computationnal and Graphical Statistics** (2015).

[10]M. Pierre-Jean, Guillem Rigaill et P. Neuvial, « Performance evaluation of DNA copy number segmentation methods », **Briefings in bioinformatics**, bbu026 (2014).

[11]Guillem Rigaill, S. Balzergue, V. Brunaud, E. Blondet, A. Rau, O. Rogier, J. Caius, C. Maugis-Rabusseau, L. Soubigou-Taconnat, S. Aubourg, L. Claire, M.-M. Marie-Laure et D. Etienne, « Synthetic data sets for the identification of key ingredients for RNA-seq differential analysis », **Briefings in bioinformatics** (2016).

[12]R. Maidstone, T. Hocking, Guillem Rigaill et P. Fearnhead, « On optimal multiple changepoint algorithms for large data », **Statistics and Computing**, 1-15 (2016).

[13]P. Fearnhead et Guillem Rigaill, « Changepoint detection in the presence of outliers », **Journal of the American Statistical Association**, 1-15 (2018).

[14]T. D. Hocking, Guillem Rigaill, P. Fearnhead et G. Bourque, « Generalized Functional Pruning Optimal Partitioning (GFPOP) for Constrained Changepoint Detection in Genomic Data », **Journal of Statistical Software (accepté)** (2020).

[15]T. D. Hocking, Guillem Rigaill, P. Fearnhead et G. Bourque, « A log-linear time algorithm for constrained changepoint detection », **JMLR** (2020).

[16]A. Celisse, G. Marot, M. Pierre-Jean et Guillem Rigaill, « New efficient algorithms for multiple change-point detection with reproducing kernels », **Computational Statistics & Data Analysis 128**, 200-220 (2018).

[17]T. D. Hocking, Guillem Rigaill et G. Bourque, « PeakSeg : constrained optimal segmentation and supervised penalty learning for peak detection in count data », in Proceedings of The 32nd International Conference on Machine Learning (2015), p. 324-332.

[18]Guillem Rigaill, T. Hocking, J.-P. Vert et F. Bach, « Learning sparse penalties for change-point detection using max margin interval regression », in Proceedings of The 30th International Conference on Machine Learning (2013), p. 172-180.

[19]T. Popova, E. Manié, D. Stoppa-Lyonnet, Guillem Rigaill, E. Barillot et M.-H. Stern, « Genome Alteration Print (GAP) : a tool to visualize and mine complex cancer genomic profiles obtained by SNP arrays », **Genome Biology 10**, R128-R128 (2009).

[20]J. J. de Ronde, Guillem Rigaill, S. Rottenberg, S. Rodenhuis et L. F. Wessels, « Identifying subgroup markers in heterogeneous populations », **Nucleic acids research**, gkt845 (2013).

[21]T. D. Hocking, V. Boeva, Guillem Rigaill, G. Schleiermacher, I. Janoueix-Lerosey, O. Delattre, W. Richer, F. Bourdeaut, M. Suguro, M. Seto et F. Bach, « SegAnnDB : interactive Web-based genomic segmentation », **Bioinformatics 30**, 1539-1546 (2014).

[22]B. Malbert, Guillem Rigaill, V. Brunaud, C. Lurin et E. Delannoy, « Bioinformatic Analysis of Chloroplast Gene Expression and RNA Posttranscriptional Maturations Using RNA Sequencing », in **Plastids** (Springer, 2018), p. 279-294.

[23]R. Zaag, J. P. Tamby, C. Guichard, Z. Tariq, Guillem Rigaill, E. Delannoy, J.-P. Renou, S. Balzergue, T. Mary-Huard, S. Aubourg, M. Martin-Magniette et V. Brunaud, « GEM2Net : from gene expression modeling to-omics networks, a new CATdb module to investigate Arabidopsis thaliana genes involved in stress response », **Nucleic acids research**, gku1155 (2014).

[24]F. Lizárraga, R. Poincloux, M. Romao, G. Montagnac, G. Le Dez, I. Bonne, Guillem Rigaill, G. Raposo et P. Chavrier, « Diaphanous-related formins are required for invadopodia formation and invasion of breast tumor cells », **Cancer research 69**, 2792-2800 (2009).

[25]A. Toullec, D. Gerald, G. Despouy, B. Bourachot, M. Cardon, S. Lefort, M. Richardson, Guillem Rigaill, M.-C. Parrini, C. Lucchesi, D. Bellanger, M. Stern, T. Dubois, X. Sastre-Garau, O. Delattre, A. Vincent-Salomon et F. Mechta-Grigoriou, « Oxidative stress promotes myofibroblast differentiation and tumour spreading », **EMBO molecular medicine 2**, 211-230 (2010).

[26]M. A. Bollet, N. Servant, P. Neuvial, C. Decraene, I. Lebigot, J.-P. Meyniel, Y. De Rycke, A. Savignoni, Guillem Rigaill, P. Hupé, A. Fourquet, B. Sigal-Zafrani, E. Barillot et J. Thiery, « High-resolution mapping of DNA breakpoints to define true recurrences among ipsilateral breast cancers », **Journal of the National Cancer Institute 100**, 48-58 (2008).

[27]B. Marty, V. Maire, E. Gravier, Guillem Rigaill, A. Vincent-Salomon, M. Kappler, I. Lebigot, F. Djelti, A. Tourdès, P. Gestraud, P. Hupé, E. Barillot, F. Cruzalegui, G. Tucker, M. Stern, J. Thiery, J. Hickman et T. Dubois, « Frequent PTEN genomic alterations and activated phosphatidylinositol 3-kinase pathway in basal-like breast cancer cells », **Breast Cancer Res 10**, R101 (2008).

[28]A.-S. Dumas, L. Taconnat, E. Barbas, Guillem Rigaill, O. Catrice, D. Bernard, A. Benamar, D. Macherel, A. El Amrani et R. Berthomé, « Unraveling the early molecular and physiological mechanisms involved in response to phenanthrene exposure », **BMC genomics 17**, 818 (2016).

[29]D. Guillaumot, M. Lopez-Obando, K. Baudry, A. Avon, Guillem Rigaill, A. F. de Longevialle, B. Broche, M. Takenaka, R. Berthomé, G. De Jaeger et al., « Two interacting PPR proteins are major Arabidopsis editing factors in plastid and mitochondria », **Proceedings of the National Academy of Sciences 114**, 8877-8882 (2017).

[30]J.-T. Brandenburg, T. Mary-Huard, Guillem Rigaill, S. J. Hearne, H. Corti, J. Joets, C. Vitte, A. Charcosset, S. D. Nicolas et M. I. Tenaillon, « Independent introductions and admixtures have contributed to adaptation of European maize and its American counterparts », **PLoS genetics 13**, e1006666 (2017).

[31]A. Lloyd, A. Blary, D. Charif, C. Charpentier, J. Tran, S. Balzergue, E. Delannoy, Guillem Rigaill et E. Jenczewski, « Homoeologous exchanges cause extensive dosage-dependent gene expression changes in an allopolyploid crop », **New Phytologist** (2018).

[32]E. Albert, R. Duboscq, M. Latreille, S. Santoni, M. Beukers, J.-P. Bouchet, F. Bitton, J. Gricourt, C. Poncet, V. Gautier, J. M. Jiménez-Gómez, Guillem Rigaill et M. Causse, « Allele specific expression and genetic determinants of transcriptomic variations in response to mild water deficit in tomato », **The Plant Journal** (2018).

[33]M.-A. Lemay, D. Torkamaneh, Guillem Rigaill, B. Boyle, A. O. Stec, R. M. Stupar et F. Belzile, « Screening populations for copy number variation using genotyping-by-sequencing : a proof of concept using soybean fast neutron mutants », **BMC genomics 20**, 1-16 (2019).

[34]C. Baldeyron, A. Brisson, B. Tesson, F. Némati, S. Koundrioukoff, E. Saliba, L. De Koning, E. Martel, M. Ye, Guillem Rigaill, D. Meseure, A. Nicolas, D. Gentien, D. Decaudin, M. Debatisse, S. Depil, F. Cruzalegui, A. Pierré, S. Roman-Roman, G. Tucker et T. Dubois, « TIPIN depletion leads to apoptosis in breast cancer cells », **Molecular oncology** (2015).

[35]V. Maire, F. Mahmood, Guillem Rigaill, M. Ye, A. Brisson, F. Némati, D. Gentien, G. C. Tucker, S. Roman-Roman et T. Dubois, « LRP8 is overexpressed in estrogen-negative breast cancers and a potential target for these tumors », **Cancer medicine** (2018).

[36]V. Maire, F. Némati, M. Richardson, A. Vincent-Salomon, B. Tesson, Guillem Rigaill, E. Gravier, B. Marty-Prouvost, L. De Koning, G. Lang, D. Gentien, A. Dumont, E. Barillot, E. Marangoni, D. Decaudin, S. Roman-Roman, A. Pierré, F. Cruzalegui, S. Depil, G. Tucker et T. Dubois, « Polo-like kinase 1 : a potential therapeutic option in combination with conventional chemotherapy for the management of patients with triple-negative breast cancer », **Cancer research 73**, 813-823 (2013).

[37]V. Maire, C. Baldeyron, M. Richardson, B. Tesson, A. Vincent-Salomon, E. Gravier, B. Marty-Prouvost, L. De Koning, Guillem Rigaill, A. Dumont, D. Gentien, E. Barillot, S. Roman-Roman, S. Depil, F. Cruzalegui, A. Pierré, G. Tucker et T. Dubois, « TTK/hMPS1 is an attractive therapeutic target for triple-negative breast cancer », **Plos One 8** (2013).

[38]S. Maubant, B. Tesson, V. Maire, M. Ye, Guillem Rigaill, D. Gentien, F. Cruzalegui, G. C. Tucker, S. Roman-Roman et T. Dubois, « Transcriptome analysis of Wnt3a-treated triple-negative breast cancer cells », **PloS one 10**, e0122333 (2015).

[39]S. Maubant, B. Tahtouh Tania, Amélie, V. Maire, F. Némati, B. Tesson, M. Ye, Guillem Rigaill, M. Noizet, A. Dumont, D. Gentien, M.-P. Bérengère, de Koning Leanne, S. F. Mahmood, D. Decaudin, F. Cruzalegui, T. G. C, S. Roman-Roman et T. Dubois, « LRP5 regulates the expression of STK40, a new potential target in triple-negative breast cancers », **Oncotarget** (2018).

[40]A. Vincent-Salomon, V. Benhamo, E. Gravier, Guillem Rigaill, N. Gruel, S. Robin, Y. de Rycke, O. Mariani, G. Pierron, D. Gentien, F. Reyal, P. Cottu, A. Fourquet, R. Rouzier, X. Sastre-Garau et O. Delattre, « Genomic instability : a stronger prognostic marker than proliferation for early stage luminal breast carcinomas », **PloS one 8** (2013).

[41]Guillem Rigaill, « Pruned dynamic programming for optimal multiple change-point detection », **arXiv preprint arXiv :1004.088** (2010).

[42]V. Runge, T. D. Hocking, G. Romano, F. Afghah, P. Fearnhead et Guillem Rigaill, « gfpop : an R Package for Univariate Graph-Constrained Change-point Detection », **arXiv preprint arXiv :2002.03646** (2020).

[43]P. Gutierrez, G. Rigaill et J. Chiquet, *Fused-Anova*, `https://r-forge.r-project.org/projects/fusedanova/`, This package adjusts a penalized ANOVA model with Fused-LASSO penality., 2013.

[44]J. Chiquet, V. Dervieux et G. Rigaill, *AriCode*, `https://CRAN.R-project.org/package=aricode`, aricode : a package for efficient computations ofstandard clustering comparison measures., 2018.

[45]A. Hulot, J. Chiquet et G. Rigaill, *MergeTree*, `https://CRAN.R-project.org/package=mergeTress`, mergeTrees : Aggregating Trees., 2018.

[46]M. Pierre-Jean, P. Neuvial et G. Rigaill, *jointSeg*, `https://CRAN.R-project.org/package=jointSeg`, jointseg : Joint Segmentation of Multivariate (Copy Number) Signals, 2015.

[47]S. Arlot, A. Celisse et Z. Harchaoui, « A kernel multiple change-point algorithm via model selection », **arXiv preprint arXiv :1202.3878** (2012).

[48]I. E. Auger et C. E. Lawrence, « Algorithms for the optimal identification of segment neighborhoods », **Bulletin of mathematical biology 51**, 39-54 (1989).

[49]F. Bach, « Efficient algorithms for non-convex isotonic regression through submodular optimization », in Advances in Neural Information Processing Systems (2018), p. 1-10.

[50]R. Baranowski, Y. Chen et P. Fryzlewicz, « Narrowest-over-threshold detection of multiple change points and change-point-like features », **Journal of the Royal Statistical Society : Series B (Statistical Methodology) 81**, 649-672 (2019).

[51]Y. Baraud, C. Giraud, S. Huet et al., « Gaussian model selection with an unknown variance », **The Annals of Statistics 37**, 630-672 (2009).

[52]P. Bastide, M. Mariadassou et S. Robin, « Detection of adaptive shifts on phylogenies by using shifted stochastic processes on a tree », **Journal of the Royal Statistical Society : Series B (Statistical Methodology) 79**, 1067-1093 (2017).

[53]R. Bellman et B. Kotkin, *ON THE APPROXIMATION OF CURVES BY LINE SEGMENTS USING DYNAMIC PROGRAMMING. II*, rapp. tech. (RAND CORP SANTA MONICA CALIF, 1962).

[54]C. Biernacki, G. Celeux et G. Govaert, « Assessing a mixture model for clustering with the integrated completed likelihood », **IEEE transactions on pattern analysis and machine intelligence 22**, 719-725 (2000).

[55]L. Birgé et P. Massart, « Gaussian model selection », **Journal of the European Mathematical Society 3**, 203-268 (2001).

[56]H. D. Bondell et B. J. Reich, « Simultaneous factor selection and collapsing levels in ANOVA », **Biometrics 65**, 169-177 (2009).

[57]A.-L. Boulesteix, S. Lauer et M. J. Eugster, « A plea for neutral comparison studies in computational sciences », **PloS one 8**, e61562 (2013).

[58]L. Boysen, A. Kempe, V. Liebscher, A. Munk, O. Wittich et al., « Consistencies and rates of convergence of jump-penalized least squares estimators », **The Annals of Statistics 37**, 157-183 (2009).

[59]S. Chakar, E. Lebarbier, C. Lévy-Leduc, S. Robin et al., « A robust approach for estimating change-points in the mean of an AR(1) process », **Bernoulli 23**, 1408-1447 (2017).

[60]A. Cleynen, S. Dudoit et S. Robin, « Comparing segmentation methods for genome annotation based on rna-seq data », **Journal of Agricultural, Biological, and Environmental Statistics 19**, 101-118 (2014).

[61]A. Cleynen et E. Lebarbier, « Segmentation of the Poisson and negative binomial rate models : a penalized estimator », **ESAIM : Probability and Statistics 18**, 750-769 (2014).

[62]H. Dette et D. Wied, « Detecting relevant changes in time series models », **Journal of the Royal Statistical Society : Series B (Statistical Methodology) 78**, 371-394 (2016).

[63]P. Fearnhead, R. Maidstone et A. Letchford, « Detecting Changes in Slope With an L 0 Penalty », **Journal of Computational and Graphical Statistics 28**, 265-275 (2019).

[64]W. D. Fisher, « On grouping for maximum homogeneity », **Journal of the American statistical Association 53**, 789-798 (1958).

[65]K. Frick, A. Munk et H. Sieling, « Multiscale change point inference », **Journal of the Royal Statistical Society : Series B (Statistical Methodology) 76**, 495-580 (2014).

[66]P. Fryzlewicz et al., « Wild binary segmentation for multiple change-point detection », **The Annals of Statistics 42**, 2243-2281 (2014).

[67]C. Gao, F. Han et C.-H. Zhang, « On estimation of isotonic piecewise constant signals », **arXiv preprint arXiv :1705.06386** (2017).

[68]D. Garreau, S. Arlot et al., « Consistent change-point detection with kernels », **Electronic Journal of Statistics 12**, 4440-4486 (2018).

[69]GDR-Bim, *"GDR Bioinformatique Moléculaire"*, (2019) `http://www.gdr-bim.cnrs.fr/` (visité le 21/11/2019).

[70]Génopole, *Bioinformatics and Biostatistical tools in medical genomics*, (2019) https://www.genopole.fr/spip.php?page=rubrique_event&id_rubrique=1108&event=1108&lang=fr#.Xeke5tHjK-E (visité le 27/12/2019).

[71]T. Ha, « A multivariate learning penalized method for a joined inference of gene expression levels and gene regulatory networks », thèse de doct. (Paris Saclay, 2016).

[72]Z. Harchaoui et C. Lévy-Leduc, « Multiple change-point estimation with a total variation penalty », **Journal of the American Statistical Association 105**, 1480-1493 (2010).

[73]K. Haynes, I. A. Eckley et P. Fearnhead, « Computationally efficient changepoint detection for a range of penalties », **Journal of Computational and Graphical Statistics 26**, 134-143 (2017).

[74]T. D. Hocking, P. Goerner-Potvin, A. Morin, X. Shao, T. Pastinen et G. Bourque, « Optimizing ChIP-seq peak detectors using visual labels and supervised machine learning », **Bioinformatics 33**, 491-499 (2016).

[75]T. D. Hocking, A. Joulin, F. Bach et J.-P. Vert, « Clusterpath : an algorithm for clustering using convex fusion penalties », in Proceedings of the 28th International Conference on International Conference on Machine Learning (Omnipress, 2011), p. 745-752.

[76]T. D. Hocking, G. Schleiermacher, I. Janoueix-Lerosey, V. Boeva, J. Cappo, O. Delattre, F. Bach et J.-P. Vert, « Learning smoothing models of copy number profiles using breakpoint annotations », **BMC bioinformatics 14**, 164 (2013).

[77]B. Jackson, J. D. Scargle, D. Barnes, S. Arabhi, A. Alt, P. Gioumousis, E. Gwin, P. Sangtrakulcharoen, L. Tan et T. T. Tsai, « An algorithm for optimal partitioning of data on an interval », **IEEE Signal Processing Letters 12**, 105-108 (2005).

[78]S. Jewell, T. D. Hocking, P. Fearnhead et D. Witten, « Fast nonconvex deconvolution of calcium imaging data », **arXiv preprint arXiv :1802.07380** (2018).

[79]S. Jewell et D. Witten, « Exact spike train inference via ℓ_0 optimization », **The annals of applied statistics 12**, 2457 (2018).

[80]N. A. Johnson, *EFFICIENT MODELS AND ALGORITHMS FOR PROBLEMS IN GENOMICS*, (2011) https://stacks.stanford.edu/file/druid:jq411pj0455/thesis_sig_page_removed-augmented.pdf (visité le 19/08/2019).

[81]N. A. Johnson, « A dynamic programming algorithm for the fused lasso and l 0-segmentation », **Journal of Computational and Graphical Statistics 22**, 246-260 (2013).

[82]R. Killick, P. Fearnhead et I. A. Eckley, « Optimal detection of changepoints with a linear computational cost », **Journal of the American Statistical Association 107**, 1590-1598 (2012).

[83]W. R. Lai, M. D. Johnson, R. Kucherlapati et P. J. Park, « Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data », **Bioinformatics 21**, 3763-3770 (2005).

[84]E. Lebarbier, « Contribution à la segmentation de processus », 113 pages, Habilitation à diriger des recherches en mathématiques (Université Paris Sud, nov. 2018).

[85]É. Lebarbier, « Detecting multiple change-points in the mean of Gaussian process by model selection », **Signal processing 85**, 717-736 (2005).

[86]M. I. Love, W. Huber et S. Anders, « Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2 », **Genome biology 15**, 550 (2014).

[87]T. Lumley, P. Diehr, S. Emerson et L. Chen, « The importance of the normality assumption in large public health data sets », **Annual review of public health 23**, 151-169 (2002).

[88]P. Mair, K. Hornik et J. de Leeuw, « Isotone optimization in R : pool-adjacent-violators algorithm (PAVA) and active set methods », **Journal of statistical software 32**, 1-24 (2009).

[89]M. Maravalle, B. Simeone et R. Naldini, « Clustering on trees », **Computational Statistics & Data Analysis 24**, 217-234 (1997).

[90]NetBio, *Réseau méthodologique MIA "Inférence de réseaux (biologiques)"*, (2019) https://mia.toulouse.inra.fr/NETBIO (visité le 29/08/2019).

[91]P. Nicolas, A. Leduc, S. Robin, S. Rasmussen, H. Jarmer et P. Bessières, « Transcriptional landscape estimation from tiling array data using a model of signal shift and drift », **Bioinformatics 25**, 2341-2347 (2009).

[92]F. Picard, S. Robin, M. Lavielle, C. Vaisse et J.-J. Daudin, « A statistical approach for array CGH data analysis », **BMC bioinformatics 6**, 27 (2005).

[93]M. D. Robinson, D. J. McCarthy et G. K. Smyth, « edgeR : a Bioconductor package for differential expression analysis of digital gene expression data », **Bioinformatics 26**, 139-140 (2010).

[94]G. Rote, *Isotonic Regression by Dynamic Programming*, (2012) `https://pdfs.semanticscholar.org/700e/3b2034cefef7b3044e2f82b3266a494fa732.pdf` (visité le 19/08/2019).

[95]G. Rote, « Isotonic regression by dynamic programming », in 2nd Symposium on Simplicity in Algorithms (SOSA 2019) (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018).

[96]V. Runge, « Is a Finite Intersection of Balls Covered by a Finite Union of Balls in Euclidean Spaces ? », **arXiv preprint arXiv :1804.06699** (2018).

[97]SPS, *From gene expression to genomic network*, (2016) `https://www6.inra.fr/saclay-plant-sciences_eng/Teaching-and-training/Summer-schools/Summer-School-2016` (visité le 27/12/2019).

[98]C. Truong, L. Oudre et N. Vayatis, « A review of change point detection methods », **arXiv preprint arXiv :1801.00718** (2018).

[99]R. Vincent, *gfpop*, (2018) `https://github.com/vrunge/gfpop` (visité le 19/08/2019).

[100]N. X. Vinh, J. Epps et J. Bailey, « Information theoretic measures for clusterings comparison : Variants, properties, normalization and correction for chance », **Journal of Machine Learning Research 11**, 2837-2854 (2010).

[101]U. Von Luxburg et al., « Clustering stability : an overview », **Foundations and Trends® in Machine Learning 2**, 235-274 (2010).

[102]Wikipedia, *Comparative genomic hybridization*, (2019) `https://en.wikipedia.org/wiki/Comparative_genomic_hybridization` (visité le 22/10/2019).

[103]Wikipedia, *Cytoplasmic male sterility*, (2019) `https://en.wikipedia.org/wiki/Cytoplasmic_male_sterility` (visité le 29/08/2019).

[104]Wikipedia, *Compositional domain*, (2019) `https://en.wikipedia.org/wiki/Compositional_domain` (visité le 22/10/2019).

[105]Wikipedia, *Wilcoxon signed-rank test*, (2019) `https://en.wikipedia.org/wiki/Wilcoxon_signed-rank_test` (visité le 29/08/2019).

[106]Wikipedia, *Welch's t-test*, (2019) `https://en.wikipedia.org/wiki/Welch%27s_t-test` (visité le 29/08/2019).

[107]Wikipedia, *Digital polymerase chain reaction*, (2019) `https://en.wikipedia.org/wiki/Digital_polymerase_chain_reaction#Comparison_between_dPCR_and_Real-Time_PCR_(qPCR)` (visité le 29/08/2019).

[108]Y.-C. Yao et S.-T. Au, « Least-squares estimation of a step function », **Sankhyā : The Indian Journal of Statistics, Series A**, 370-381 (1989).

[109]T. Zhao, H. Liu, K. Roeder, J. Lafferty et L. Wasserman, « The huge package for high-dimensional undirected graph estimation in R », **Journal of Machine Learning Research 13**, 1059-1062 (2012).