# Problèmes de clustering liés à la synchronie en écologie : estimation de rang effectif et détection de ruptures sur les arbres

Solène Thépaut

NNT : 2019SACLS477

# THÈSE DE DOCTORAT

de

L'Université Paris-Saclay

École doctorale de mathématiques Hadamard (EDMH, ED 574)

*Établissement d'inscription :* Université Paris-Sud

*Laboratoire d'accueil :* Laboratoire de mathématiques d'Orsay, UMR 8628 CNRS

*Spécialité de doctorat :* Mathématiques aux interfaces

**Solène THEPAUT**

## Problèmes de clustering liés à la synchronie en écologie : estimation de rang effectif et détection de ruptures sur les arbres

*Date de soutenance :* 06 Décembre 2019

*Après avis des rapporteurs :* Ghislaine GAYRAUD (Université technologique de Compiègne)
Franck PICARD (CNRS)

*Jury de soutenance :*

| | |
|---|---|
| Sylvain ARLOT | (Université Paris-Sud) Président du jury |
| Ghislaine GAYRAUD | (Université technologique de Compiègne) Rapporteur |
| Christophe GIRAUD | (Université Paris Sud) Directeur de thèse |
| Émilie LEBARBIER | (Université Paris Nanterre) Examinateur |
| Franck PICARD | (CNRS) Rapporteur |
| Nicolas VERZELEN | (INRA) Codirecteur de thèse |

# REMERCIEMENTS

Je voudrais tout d'abord remercier infiniment mes deux directeurs de thèse Christophe Giraud et Nicolas Verzelen. Leur bienveillance, leur disponibilité même lorsqu'ils avaient un emploi du temps impossible, leur patience et leurs encouragements m'ont permis d'aller au bout des trois longues années que dure une thèse de mathématiques. Je réalise la chance que j'ai eue de pouvoir être supervisée par deux chercheurs aussi talentueux et ce fut un honneur de pouvoir travailler avec eux. Je suis extrêmement reconnaissante de l'énergie et du temps qu'ils ont dépensés du début de mon doctorat jusqu'à la soumission de mon manuscrit.

Je tiens également à remercier Guillem Rigaill, avec lequel j'ai pu travailler sur le dernier chapitre de ma thèse. Il a toujours été très encourageant et positif concernant les sujets sur lesquels j'ai eu la chance de travailler avec lui.

Merci aux rapporteurs Ghislaine Gayraud et Franck Picard d'avoir accepté d'évaluer ma thèse.

Merci également aux membres du Jury, Sylvain Arlot, Ghislaine Gayraud, Christophe Giraud, Emilie Lebarbier, Franck Picard et Nicolas Verzelen.

Je remercie l'Ecole Doctorale de Mathématiques Hadamard pour m'avoir fournie un cadre plus qu'idéal pour effectuer mon doctorat. Merci au directeur Frédéric Paulin pour sa bienveillance.

J'ai eu l'immense privilège de partager un bureau avec Jeanne et Julien et un étage du Laboratoire de mathématiques d'Orsay avec Luc et Thomas. Je les remercie, ainsi que tous les autres doctorants pour leur bonne humeur, leur aide et leur soutien. En restant dans le LMO, merci à Christine Keribin et Luc Joseph avec qui j'ai effectué un grand nombre d'heures pour mon monitorat. Assurer des TDs a été une expérience plus qu'enrichissante et je n'aurais pas pu espérer meilleurs encadrants.
Je remercie aussi Théo, qui a commencé sa thèse en écologie en même temps que moi et avec qui j'ai travaillé sur l'un des chapitres de ma thèse.

Puisqu'une thèse ne peut pas se faire sans être bien entouré, je remercie infiniment ma mère et Jean-Baptiste pour leur soutien et leur aide quel que soit $x$. C'est le parcours de Kine qui m'a poussé à vouloir faire un doctorat. Je remercie aussi tous ceux sans qui je n'aurais jamais pu aller jusqu'au bout et dont certains se sont improvisés relecteurs : mon frère Colin, Camille, Amel, Brice, Florian, pour l'ensemble de leur oeuvre, Isabelle, pour sa gentillesse, Tontzy pour ses GIFs, Nukka et Balou pour leur présence et soutien sans faille. Merci à eux pour les fous rires, les soirées, les régalades et tout le reste.

Merci également à ma famille qui m'a soutenue et inspirée depuis le début de mes études : mon père, Corinne, Grand Pierre, Claire, Florence, Stéphane, Philippe. Merci à mes cousins : Lili, Micha, Matthieu et Sébastien pour leur presque-calme qui m'a permis de finir la rédaction de mon introduction dans un petit village près d'Avignon. Ayant fait relire le manuscrit par un nombre conséquent de personnes, je les remercie toutes pour le temps passé à relever mes fautes d'orthographe innommables.

Merci à toutes les personnes nommées dans ces deux derniers paragraphes d'avoir cru en moi. Certaines m'ont montré mille fois que j'étais capable de faire bien plus que ce que je ne pensais.

À Jacqueline, Gisèle, Michel et Jean Pierre.

# CONTENTS

# CHAPTER 1

# INTRODUCTION

## 1.1 Motivations générales et contexte

Les travaux réalisés au cours de ma thèse ont tous comme thème central la synchronie, notion issue de l'écologie qui sera définie dans cette introduction. Afin d'expliquer les motivations et applications liées à mon domaine de recherche, je prendrai soin d'introduire aussi bien le contexte écologique et les notions qui lui sont liées que le contexte mathématique.

**Stabilité des écosystèmes et synchronie** Un écosystème est une unité écologique de base formée par le milieu, le biotope, et les organismes qui y vivent, la biocénose et dans lequel il existe des interactions entre les êtres vivants. Un lac, une forêt ou un champs sont considérés comme des écosystèmes. A plus grande échelle, un pays ou un continent sont aussi des écosystèmes. En écologie, une communauté est un ensemble d'organismes appartenant à des populations d'espèces différentes constituant un réseau de relations. La définition d'une communauté est assez proche de celle de la biocénose d'un écosystème. Cependant, les écologues utilisent le terme communauté afin de désigner des individus représentant un sous-ensemble de la biocénose (https://fr.wikipedia.org/wiki/Communauté). Par exemple, dans un jardin, les différentes espèces de papillons sont considérées comme une communauté au sein de l'écosystème formé par le jardin. On appelle fonctions d'un écosystème l'ensemble des processus qui contrôlent les flux de matière et d'énergie au sein d'un écosystème, comme la production et le recyclage de biomasse, la pollinisation, etc. [11, 19, 100, 110]. Les groupes fonctionnels d'espèces sont des groupes formés par des espèces partageant les mêmes fonctions au sein d'un écosystème ou d'une communauté [110].

L'être humain a un impact énorme sur les écosystèmes à cause de l'exploitation intensive de ces derniers. L'activité de l'espèce humaine ne cessant d'augmenter, elle est responsable d'une partie importante des changements que l'on peut remarquer à une échelle globale. Les changements globaux perturbent les écosystèmes et leur fonctionnement. il devient nécessaire de comprendre l'impact et les mécanismes de la dégradation des habitats et du changement climatique sur les êtres vivants et sur les communautés. Comprendre comment les perturbations environnementales affectent la stabilité au sein des écosystèmes est un enjeu majeur en écologie. La stabilité d'un écosystème se caractérise comme la capacité de ce dernier à rester dans un état de référence au cours du temps, et ce malgré les variations des conditions environmentales [46]. Entres autres, un écosystème est caractérisé par les êtres vivants qui le peuplent. Il semble donc naturel que l'un des aspects de l'étude d'un écosystème soit l'étude de sa biocénose. Le sujet est vaste et il existe de nombreuses façons de l'aborder. Du point de vue des populations d'un écosystème, la stabilité d'une communauté peut également se définir comme la résilience des espèces aux changements globaux. C'est-à-dire

la capacité des espèces au sein d'une communauté à retrouver les fonctions d'un état de référence après une ou plusieurs perturbations. Cette définition de la stabilité nous pousse à vouloir établir et quantifier la résilience des espèces aux changements globaux afin de comprendre certains mécanisme responsables de la stabilité des communautés. Il existe de nombreux articles en écologie sur le sujet de la stabilité dans les communautés, voir [24, 81, 96] par exemple pour un aperçu.

Dans le cadre de ma thèse, nous nous limiterons à l'étude des variations inter-annuelles des abondances d'espèces présentes au sein d'une communauté, où l'abondance d'une espèce à un instant $t$ est définie comme le nombre d'individus présents ou observés à cet instant. Ainsi, pour étudier la stabilité des communauté, nous voulons définir et quantifier le fait que les abondances de certaines espèces varient de façon similaire. C'est à ce moment qu'intervient la synchronie. Il existe plusieurs types de synchronie. Je m'intéresse à seulement deux d'entre elles, la synchronie inter-espèce et la synchronie spatiale d'une espèce. Ces deux notions sont les plus importantes afin d'appréhender ma problématique.
La synchronie inter-espèces se définie comme la similarité des variations d'abondance entre espèces. Deux espèces sont dites synchrones si les variations de leurs abondances suivent un modèle similaire. En termes mathématiques, des espèces sont synchrones si les séries temporelles de leurs abondances sont fortement corrélées. La définition de la synchronie spatiale est semblable à la définition précédente sauf qu'au lieu de comparer les séries temporelles de deux espèces différentes, on compare les séries temporelles d'abondances d'une seule espèce, en différents points géographiques. Dans le cadre de la synchronie interspécifique, chaque série temporelle représente une espèce, alors que dans le cadre de la synchronie spatiale, chaque série temporelle représente un site, c'est-à-dire un lieu géographique. S'intéresser à la synchronie inter-espèces permet de regarder directement le lien diversité-stabilité dans un écosystème. La synchronie spatiale, elle, est utile pour essayer de comprendre les facteurs qui influencent les dynamiques compensatoires entre espèces et donc la stabilité des communautés.

Les différents chapitres de ma thèse apportent des outils et des analyses qui peuvent permettre de mieux comprendre les dynamiques de population ainsi que le lien entre cette dynamique et l'état général de l'écosystème. Le lien entre la synchronie et la stabilité peut être illustré de la façon suivante. Considérons un écosystème peuplé de différentes espèces, alors si toutes les espèces sont sensibles de la même façon aux facteurs environnementaux, on s'attend à ce que la population de chaque espèce varie de la même façon. C'est-à-dire, que si un facteur évolue de façon favorable aux espèces, alors le nombre d'individus de chaque espèce augmente tandis que lorsqu'un facteur évolue défavorablement, le nombre d'individus diminue. Cela signifie que si un événement très défavorable aux espèces apparaît, il est fortement probable que cela provoque une extinction de l'écosystème. A l'inverse, s'il existe une vraie diversité au niveau des sensibilités de chaque espèce, une évolution des conditions environnementales sera favorable à certaines espèces et défavorable à d'autres. Au final, même si un événement provoque l'extinction de certaines espèces, la probabilité d'extinction de l'écosystème est quasi-nulle.

Nos travaux sur la synchronie s'inscrivent dans la thématique fondamentale du lien entre la diversité et la stabilité des communautés ou plus généralement du lien diversité-stabilité dans les écosystèmes. Il existe de nombreux travaux qui traitent ce sujet en écologie [43]. Cependant, la plupart des articles s'intéressent à des espèces de plantes dans des milieux artificiels et fermés. Le lien diversité-stabilité pour les espèces animales n'a pas été beaucoup exploré, il en est de même pour l'étude de la diversité-stabilité en milieu naturel. Ceci est compréhensible car les données en milieu naturel ont tendance à être plus difficiles à collecter et leur qualité n'est pas toujours satisfaisante. S'il est facile d'observer une communauté de plantes, il est beaucoup plus compliqué d'effectuer des mesures d'abondances pour des espèces mobiles surtout dans un milieu ouvert. Pourtant, comprendre l'impact des changements globaux sur la stabilité des communautés animales en milieu naturel devient crucial afin de prévoir, et dans certains cas, prévenir les conséquences de l'intensification de

l'activité humaine ou d'autres événements environnementaux.

Afin de détecter et de quantifier la synchronie entre deux séries temporelles, nous définissons une mesure de synchronie. La définition mathématique de la synchronie nous pousse à définir une mesure basée sur la corrélation entre les séries temporelles d'abondances. Proposée par Karl Pearson en 1896 [98], la corrélation de Pearson $\rho^P \in [-1, 1]$ de deux variables $X \in \mathbb{R}$ et $Y \in \mathbb{R}$ de variance $\sigma_X^2$ et $\sigma_Y^2$ se définie comme :

$$\rho^P(X, Y) = \frac{\mathrm{Cov}(X, Y)}{\sigma_X \sigma_Y}. \tag{1.1}$$

Si $X$ et $Y$ sont des variables aléatoires dans $\mathbb{R}$ comme c'est le cas ici, alors la covariance est égale à $\mathbb{E}[XY] - \mathbb{E}[X]\,\mathbb{E}[Y]$ où $\mathbb{E}[.]$ est la fonction espérance.
Si maintenant $\mathbf{X} \in \mathbb{R}^p$ et $\mathbf{Y} \in \mathbb{R}^p$ sont deux vecteurs aléatoires de taille $p$, alors la covariance entre $\mathbf{X}$ et $\mathbf{Y}$ s'écrit

$$\mathrm{Cov}(\mathbf{X}, \mathbf{Y}) = \frac{1}{p} \sum_{i=1}^{p} (\mathbf{X}_i - \mathbb{E}[\mathbf{X}])\,(\mathbf{Y}_i - \mathbb{E}[\mathbf{Y}])\,. \tag{1.2}$$

Les vecteurs $\mathbf{X}$ et $\mathbf{Y}$ peuvent représenter, par exemple, deux séries temporelles d'abondances de taille $p$. Contrairement à la corrélation de Pearson, la corrélation de Spearman $\rho^S \in [-1, 1]$ mesure des relations non affines entre deux variables. La définition est similaire à celle de $\rho^P$ sauf qu'au lieu de considérer les vecteurs de variables aléatoires $\mathbf{X}$ et $\mathbf{Y}$, on considère le rang des valeurs de $\mathbf{X}$ et $\mathbf{Y}$. Un exemple simple, si $\mathbf{X} = (1, 4, 2, 10)$ alors le rang de $\mathbf{X}$ noté $rg_{\mathbf{X}}$ est égal à $(1, 3, 2, 4)$.

$$\rho^S = \frac{\mathrm{Cov}(rg_{\mathbf{X}}, rg_{\mathbf{Y}})}{\sigma_{rg_{\mathbf{X}}} \sigma_{rg_Y}}, \tag{1.3}$$

où $\sigma^2_{rg_{\mathbf{X}}}$ et $\sigma^2_{rg_{\mathbf{Y}}}$ sont les variances des variables $rg_{\mathbf{X}}$ et $rg_{\mathbf{Y}}$.
Une comparaison des différentes méthodes pour calculer la corrélation entre deux variables ou vecteurs aléatoires est proposée dans [14].

Puisque ce sont les variations des abondances et non leurs amplitudes qui permettent d'estimer la synchronie entre deux séries temporelles, l'idéal est d'utiliser une corrélation basée sur le rang des observations. Si la corrélation de Pearson est souvent utilisée comme mesure de corrélation en écologie [35, 84], la corrélation de Spearman est beaucoup plus adaptée à nos objectifs car elle estime à quel point deux variables sont corrélées de façon monotone, sans que la dépendance ne soit linéaire. Que ce soit pour la synchronie inter-espèce ou la synchronie spatiale, il est possible de créer une matrice de corrélation entre les observations dont les coefficients sont les corrélations de Spearman entre deux séries temporelles.
Si le coefficient de Spearman de deux espèces est égal à 1, alors ces deux espèces sont parfaitement synchrones, leurs séries temporelles suivent exactement les mêmes variations. Si ce coefficient est arbitrairement proche de 1, alors on dira que les deux espèces sont fortement synchrones. Une valeur de 0 indique qu'il n'existe aucune corrélation entre les séries d'abondances des deux espèces. Cela peut être dû au fait qu'elle ne sont pas du tout sensibles aux mêmes facteurs environnementaux. Ces deux espèces sont dites asynchrones. Si le coefficient est égal à $-1$ ou est proche de $-1$, cela signifie que les séries temporelles sont corrélées mais négativement.

Un des objectifs écologiques principaux est de faire apparaître au sein d'une communauté des groupes d'espèces synchrones appelés groupes de synchronie. Une fois les groupes formés, nous voulons, avec l'aide et l'expertise d'écologues, étudier l'impact des groupes de synchronie sur la stabilité de la communauté ou de l'écosystème ainsi que les moteurs de la synchronie. L'idée de former des groupes d'espèces selon leurs fonctions a été formulée par Schulze et Mooney [110]. Notre objectif n'est pas forcément de créer des groupes fonctionnels d'espèces mais des groupes d'espèces synchrones, même s'il est possible que certaines espèces partageant les mêmes fonctions soient synchrones en notre sens.

Afin de créer les groupes de synchronie, s'il est possible d'utiliser des méthodes statistiques d'apprentissage non supervisé appelées clustering, nous verrons dans la suite que d'autres approches sont possibles.

**Clustering et écologie** Le clustering, aussi appelé apprentissage non supervisé, est une méthode d'analyse de données permettant de diviser un jeu de données en plusieurs groupes homogènes d'observations. Pour chaque observation sont données les valeurs d'une ou plusieurs variables explicatives qui permettent de caractériser l'observation selon des attributs. Une distance, ou dissimilarité, est définie sur le jeu de données, ce qui permet de quantifier la distance entre deux observations. Un algorithme de clustering se base ensuite sur les distances entre les observations pour créer des groupes dans lesquels les observations sont proches au sens de la distance choisie. Ces algorithmes cherchent à minimiser des distances entre les observations, afin que la distance intra-groupe, soit la moyenne des distances entre les observations d'un même groupe, soit minimale et la distance inter-groupe, soit la moyenne des distances entre les observations de différents groupes, soit maximale. Le livre [62] est une excellente introduction au clustering et aux différents algorithmes d'apprentissage non supervisés. Dans notre sujet d'étude, la mesure de synchronie définie plus haut à l'aide de la corrélation de Spearman peut servir de mesure de similarité entre deux séries temporelles.

Soit un groupe d'observation obtenu grâce à une méthode de clustering, il est possible de définir le centre du groupe, en prenant par exemple le centre de gravité. Alors, dans certains cas idéaux, les observations d'un même groupe partagent exactement la même moyenne et le centre du groupe est égal à cette moyenne. Dans ce cas, l'écart des observations par rapport au centre s'explique par la variance des données plus ou moins faible. On parle alors de clustering parfait. A l'inverse, il arrive que les moyennes des observations d'un même groupe ne soient qu'approximativement les mêmes. Dans ce cas, les observations peuvent paraître beaucoup plus dispersées autour du centre de leur groupe. On parle alors de clustering imparfait ou approximatif. Dans des cas critiques de clustering imparfait, il peut être difficile de déterminer l'appartenance à un groupe d'une observation ou de décider si un groupe de diamètre très large ne serait pas en fait deux groupes distincts. Il est alors beaucoup plus compliqué pour un algorithme de clustering de trouver des groupes homogènes car la distance entre les observations d'un même groupe peut être relativement grande.

Parmi les algorithmes de clustering les plus populaires, on trouve le K-means et le clustering hiérarchique. Le K-means est une méthode de partitionnement dont l'idée originale date des années 1950. C'est désormais l'un des algorithmes de clustering les plus utilisés [56] et il existe de nombreuses améliorations de cet algorithme, comme le X-means [99] qui propose une méthode d'initialisation plus efficace que dans le K-means classique, ou d'autres variantes [76, 125].
Le clustering hiérarchique est un algorithme qui a pour but de construire ce qu'on appelle un dendogramme. Un dendogramme est un arbre dont les feuilles sont les observations et les noeuds internes des groupes d'observations formés en fonction de la distance $d(.,.)$ choisie. Il existe deux types de stratégie :

- La méthode agglomérative : au début, chaque observation est seule dans son propre groupe. Puis, à chaque étape, les deux observations ou groupes d'observations qui sont les plus proches en termes de distance $d(.,.)$ sont fusionnées pour ne former qu'un seul groupe. A la fin de cette procédure, toutes les observations font partie d'un unique groupe. L'algorithme stocke les distances auxquelles les observations ou groupes d'observations ont été fusionnés et les partitions créées à chaque étape dans le dendogramme.

- La méthode divisive : au début, toutes les observations font partie d'un unique groupe puis, à chaque étape, on divise un des groupes actuels en deux en minimisant un critère, comme la somme des distances à l'intérieur des groupes, ou en maximisant la distance entre les groupes.

De même, les distances auxquelles les groupes sont scindés ainsi que les partitions créées sont stockées dans un dendogramme.

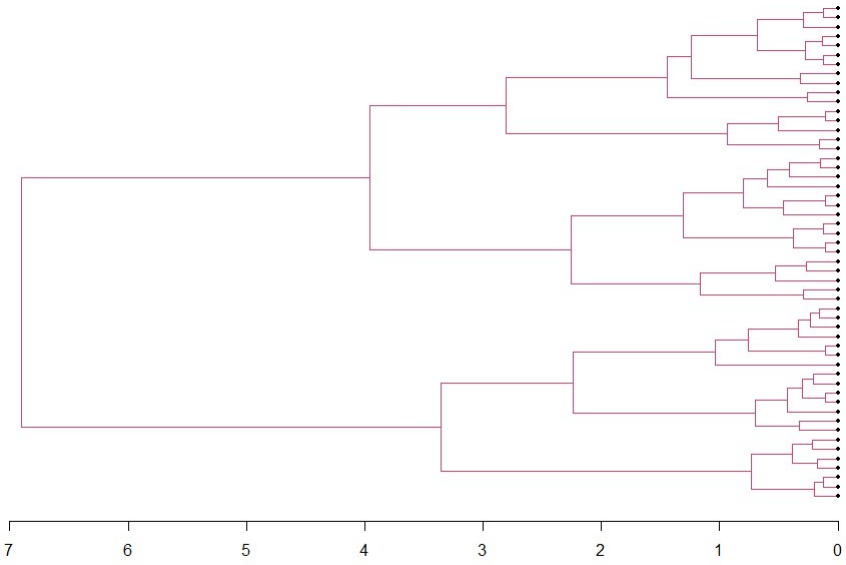



Figure 1.1 – Un exemple de dendogramme. Les feuilles (à droite) représentent les observations. Les noeuds représentent les groupes formés au fur et à mesure de l'algorithme de clustering hiérarchique. Les observations appartenant à un groupe sont celles telles qu'il existe un chemin entre la feuille correspondante et le noeud représentant le groupe. En bas figure une échelle de distance. Si un noeud se trouve à une distance $d$, alors c'est que la distance entre les deux groupes qui ont formés ce noeud sont à une distance $d$ l'un de l'autre.

Une fois le dendogramme créé, plusieurs méthodes existent afin de déterminer le nombre de groupes optimal pour le jeu de données. Ces methodes sont présentées dans [62]. Une fois le nombre $k$ de groupes choisi, la partition optimale proposée par l'algorithme de clustering hiérarchique peut être lue dans le dendogramme. Je renvoie encore une fois au livre de Kauffman pour de plus amples informations sur le clustering hiérarchique [62].

Deux problèmes principaux se posent avec les algorithmes de clustering. Le premier provient du fait que l'on ne connaît pas les vrais groupes des observations. Il est possible de tester les algorithmes sur des jeux de données dont on connaît la vraie partition des observations. Cependant, il peut être difficile, lorque l'on s'intéresse à de nouveaux jeux de données, d'estimer si la partition trouvée est bien la vraie partition de l'échantillon. De plus, si la fonction à minimiser par l'algorithme n'est pas convexe, il est possible que l'algorithme produise une partition sous-optimale. Le deuxième problème problème est lié au premier car il concerne le choix du nombre de groupes. La plupart des algorithmes de clustering ont besoin en entrée du nombre de groupes de la partition qu'il doit construire. S'il est parfois possible d'obtenir des informations grâce à la théorie du domaine d'où proviennent les données, le plus souvent il est nécessaire de faire appel à d'autres méthodes dont les plus populaires sont, par exemple, la méthode du coude ou la méthode silhouette et sont discutées dans [62].

Les algorithmes de clustering trouvent des applications dans divers domaines. Par exemple, la plateforme de diffusion Netflix utilise du clustering afin de construire des groupes d'utilisateurs et des groupes de contenu afin d'adapter ses recommandations à chaque client [27]. En finance, le clustering peut être utilisé pour identifier des changements de régimes dans un signal [80]. C'est

également un outil permettant de repérer les SPAM parmi les messages d'une boîte mail. Et dans un thème plus proche du nôtre, dans [33], le Fuzzy clustering, qui consiste à assigner des probabilités d'appartenance aux différents groupes plutôt qu'un groupe, est utilisé comme outil de description des communautés écologiques.

**Différentes stratégies** Comme la plupart des problématiques en écologie, la synchronie au sein d'une communauté est un sujet qui peut être étudié de plusieurs façons. Au lieu de choisir un unique angle d'attaque, ma thèse se divisera en trois parties indépendantes présentant trois facettes différentes de la synchronie. Ce choix permet d'adopter différentes stratégies en vu d'apporter des analyses et outils mathématiques pour répondre aux questions sur le lien entre diversité et synchronie, deux notions très liées à la stabilité des communautés, et sur les conséquences des changements environnementaux à l'échelle des communautés. Même si ces trois parties sont liées par leurs motivations d'étude de la synchronie au sein des communautés et par le clustering de façon plus ou moins directe, les domaines mathématiques abordés dans chacune des parties diffèrent grandement. Je prendrai soin dans la suite de cette introduction de justifier le lien entre chaque chapitre et la synchronie. Je préciserai également l'angle abordé ainsi que notre contribution.

## 1.2 Estimation des normes de Schatten et rang effectif

### 1.2.1 Motivations

Le premier chapitre est théorique et ne fera pas l'objet d'une application sur des données réelles dans ce manuscrit. Il est consacré à l'estimation de fonctionnelles de normes de Schatten de matrices. Avant d'introduire nos travaux, il est important de préciser le lien avec l'étude de la synchronie inter-espèces au sein d'une communauté.

Dans le cadre de l'étude de la synchronie, des séries temporelles d'abondances de taille $p$ de $n$ espèces différentes sont observées au sein d'une communauté. Pour chaque espèce $i$, nous avons accès à une série temporelle $\mathbf{y_i}$, telle que les moyennes des lois des coefficients de $\mathbf{y_i}$ sont stockées dans un $p$-vecteur $\mathbf{m_i}$ appelé vecteur moyenne. Soit une espèce $i$ de l'échantillon, son abondance pour l'année $j$ est alors notée $y_{i,j}$ et la moyenne de sa distribution est notée $\mu_{i,j}$. On note $\mathbf{Y} \in \mathbb{R}^{n\times p}$, appelée matrice d'abondances, la matrice des observations dont les lignes sont égales aux séries temporelles d'abondances des espèces. Chaque ligne correspond à une espèce et chaque colonne à une année d'observation.
Nous supposons de plus la matrice d'observation $\mathbf{Y}$ se décompose comme la somme d'une matrice déterministe $\mathbf{A} \in \mathbb{R}^{n\times p}$ et d'une matrice aléatoire $\mathbf{E} \in \mathbb{R}^{n\times p}$ représentant un bruit.

$$\mathbf{Y} = \mathbf{A} + \mathbf{E}, \tag{1.4}$$

où $\mathbf{A}$ est une matrice telle que $\mathbf{A}_{i,j} = \mu_{i,j}$. Ainsi, les lignes de $\mathbf{A}$ sont les vecteurs $\mathbf{m_i}$ dont les coefficients sont les moyennes des distributions des observations et $\mathbf{E}$ est une matrice dont les coefficients sont aléatoires et indépendants.

Si deux espèces sont synchrones au sens défini précédemment, les variations des séries temporelles de leurs abondances seront corrélées positivement. Cela se traduit par le fait que les variations des moyennes de leur loi seront corrélées positivement. Dans un monde parfait, il existe $K$ groupes de synchronie et les espèces appartenant à un même groupe suivent une loi ayant exactement les mêmes paramètres. Toutes les observations du groupe $k$ suivent une même loi et partagent toutes le même $p$-vecteur moyenne $\mathbf{m_k} \in \mathbb{R}^p$. Quel que soit $k$, toutes les lignes de la matrice des moyennes $\mathbf{A}$ dans (1.4) correspondant aux espèces du groupe $k$ sont égales à $\mathbf{m}_k$. Alors, le rang de la matrice

$\boldsymbol{A}$ est égal au nombre de groupes d'espèces synchrones.

Dans ce cas, et s'il est possible d'observer la matrice $\mathbf{A}$ au lieu de $\mathbf{Y}$, le taux de synchronisation est défini comme le ratio $\frac{K}{n}$. Si le taux de synchronisation est faible, cela signifie qu'il existe peu de groupes d'espèces synchrones, ce qui compromet la stabilité de l'écosystème. A l'inverse, si le taux de synchronisation est élevé, il existe un grand nombre de groupes d'espèces synchrones, et donc les espèces ont tendance à être asynchrones les unes des autres. Le taux de synchronisation est une information importante car avoir une estimation du nombre de groupes d'espèces synchrones est utile pour se faire une idée de la diversité au sein d'une communauté. Cela permet de faciliter l'utilisation d'algorithmes de clustering car, comme vu précédemment, le nombre de groupes est un paramètre qui peut poser problème lorsqu'il n'est pas connu.

Malheureusement, lorsque l'on observe de vraies abondances d'espèces dans une communauté, il est impossible de considérer que l'on est dans le cas idéal présenté précédemment. La première difficulté provient du fait que même si deux espèces sont fortement synchrones au sens où $\rho_S$ est proche de 1, alors les moyennes de leur loi ne seront pas égales, mais au mieux proportionnelles. Dans ce cas, et si les moyennes des espèces appartenant à un même groupe noté $G_k$ sont parfaitement proportionnelles, dans le sens où il existe un scalaire $\lambda > 0$ tel que $\mathbf{m_i} = \lambda \mathbf{m_{i'}}$, pour tout $i$ et $i'$ dans $G_k$, le rang de $\mathbf{A}$ sera toujours égal au nombre de groupes $K$. Si le but est uniquement d'estimer le taux de synchronisation au sein d'un échantillon, il est toujours possible de calculer la ratio $\frac{K}{n}$. Pour plus de clarté, lorsque nous ferons référence à ce modèle dans la suite, nous l'appellerons modèle (Ideal Proportionnel).

De nouveau, il est optimiste d'espérer une telle structuration des données. En effet, la nature des données implique un clustering imparfait. Il est en effet difficile d'imaginer des groupes bien délimités où les espèces d'un même groupe sont synchrones les unes avec les autres et totalement asynchrones avec le reste des espèces. Il est encore plus difficile d'imaginer que toutes les espèces d'un groupe soient parfaitement ou fortement synchrones entre elles au point que la variation de leurs abondances soient parfaitement proportionnelles. Toutefois, s'il existe des groupes d'espèces synchrones et que nous sommes capables de mesurer la synchronie entre les espèces, nous pouvons supposer que notre modèle s'approche du modèle décrit précédemment. Soit $\mathbf{A}$ la matrice des moyennes des observations dans le modèle réel, nous supposons que $\mathbf{A}$ se décompose comme la matrice $\mathbf{M} \in \mathbb{R}^{n \times p}$ des moyennes si les observations étaient tirées selon le modèle (Idéal Proportionnel), plus une matrice $\boldsymbol{\Delta} \in \mathbb{R}^{n \times p}$ dont les coefficients représentent les variations par rapport au modèle (Idéal Proportionnel).

$$\mathbf{A} = \mathbf{M} + \boldsymbol{\Delta}. \tag{1.5}$$

La matrice $\boldsymbol{\Delta}$ permet de modéliser les particularités des données réelles. Dans ce cas, il est possible de décomposer la matrice $\mathbf{Y}$ comme :

$$\mathbf{Y} = (\mathbf{M} + \boldsymbol{\Delta}) + \boldsymbol{\epsilon}. \tag{1.6}$$

La matrice $\boldsymbol{\Delta}$ de taille $n \times p$ est inconnue et correspond aux variations qui existent entre les moyennes réelles des lois des observations et les moyennes idéales du modèle (Ideal Proportionnel).

Malgré la proximité du modèle réel avec un modèle où les moyennes des lois des espèces d'un même groupe sont propositionnelles, il ne possède pas les propriétés permettant de calculer le rang de la matrice $\mathbf{M}$ à partir de $\mathbf{Y}$. En effet, la matrice inconnue $\boldsymbol{\Delta}$ perturbe les coefficients de $\mathbf{M}$ de façon à ce que le rang de la matrice des moyennes $\mathbf{M} + \boldsymbol{\Delta}$ ne soit plus égal au nombre de groupes d'espèces synchrones $K$ comme c'est le cas dans (Ideal Proportionnel). Comme le rang est très sensible aux petites variations des coefficients d'une matrice, si $\boldsymbol{\Delta}$ est tirée selon une loi à densité nous pouvons même dire que $\mathbf{A}$ est presque surement de rang plein c'est à dire que son rang est presque surement égal au minimum entre $n$ et $p$.

### 1.2.2 Normes de Schatten et Rang effectif

D'après ce qui précède, dans le cas de données réelles, il est impossible d'estimer le rang de $\mathbf{M}$ à partir de la matrice d'observation $\mathbf{Y}$ à cause de $\mathbf{\Delta}$ et du bruit $\mathbf{E}$. Si l'on oublie pour l'instant la matrice $\mathbf{E}$, il est tout de même possible d'obtenir des informations sur le taux de synchronisation de $\mathbf{M}$ à partir de $\mathbf{A}$. En effet, il existe des quantités plus robustes que le rang, comme le spectre de la matrice, qui rendent compte du taux de synchronisation dans une matrice d'observations. En effet, les valeurs singulières donnent des informations sur les corrélations entre les lignes ou les colonnes d'une matrice. Les valeurs singulières de $\mathbf{A}$ sont les racines carrées des valeurs propres de la matrice carrée $\mathbf{A}^T\mathbf{A}$, où $\mathbf{A}^T$ désigne la transposée de $\mathbf{A}$. Au vu de sa dimension, $\mathbf{A}$ possède $\min(n,p)$ valeurs singulières notées $\sigma_1(\mathbf{A}) \geq \sigma_2(\mathbf{A}) \geq \ldots, \sigma_{n\wedge p}(\mathbf{A})$ pour $i$ entre 1 et $n \wedge p$. Supposons sans perte de généralité que $n$ est plus grand que $p$, alors la matrice $\mathbf{A}^T\mathbf{A}$ est une matrice de taille $p \times p$ qui possède $p$ valeurs propres notées $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$.

$$\sigma_i(\mathbf{A}) = \sqrt{\lambda_i}, \tag{1.7}$$

pour tout $i$ entre 1 et $p$.

La décomposition en valeurs singulières (SVD pour son nom anglais Singular Values Decomposition) est une représentation matricielle qui s'applique à n'importe quelle matrice réelle ou complexe. La matrice $\mathbf{A}$ introduite plus haut se décompose comme le produit de trois matrices $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Avec $\mathbf{\Sigma}$ une matrice de taille $n \times p$ dont les coefficients sont tous nuls sauf les coefficients diagonaux qui sont égaux à $\sigma_i(\mathbf{A})$, et $\mathbf{U}$ et $\mathbf{V}$ deux matrices unitaires de taille $n \times n$ et $p \times p$ respectivement. Les valeurs singulières de $\mathbf{A}$ sont liées au rang de $\mathbf{A}$ car celui ci est égal au nombre des valeurs singulières non nulles de $\mathbf{A}$. Les valeurs singulières sont plus robustes que le rang et ne sont pas fortement modifiées par des petites perturbations des coefficients de la matrice observée. Si le rang de $\mathbf{A}$ et celui de $\mathbf{M}$ ne sont pas du tout égaux, même pour des coefficients de $\mathbf{\Delta}$ très petits, il est possible que leurs valeurs singulières ne soient pas si différentes. En effet, le théorème de Weyl [133], assure que pour tout $i$ entre 1 et $p \wedge n$ :

$$|\sigma_i(\mathbf{A}) - \sigma_i(\mathbf{M})| \leq \|\mathbf{A} - \mathbf{M}\|_\infty = \|\mathbf{\Delta}\|_{op}, \tag{1.8}$$

où $\|.\|_\infty$ est la norme matricielle d'opérateur.

Si les coefficients de $\mathbf{\Delta}$ sont arbitrairement petits, alors les valeurs singulières de $\mathbf{A}$ seront arbitrairement proches de celles de $\mathbf{M}$. Si le rang de $\mathbf{A}$ est égal à $p$, alors ses valeurs singulières sont toutes non nulles. Cependant, si les coefficients de $\mathbf{\Delta}$ restent arbitrairement petits, il existe un certain nombre de valeurs singulières de $\mathbf{A}$ qui sont arbitrairement petites, c'est-à-dire arbitrairement proches de 0. D'autres propriétés et théorèmes utiles sur les valeurs singulières peuvent être trouvés dans [114].

Même s'il est difficile, voire impossible, de retrouver le rang de $\mathbf{M}$ à partir de la matrice $\mathbf{A}$, il est possible de calculer le nombre de valeurs singulières de $\mathbf{A}$ qui sont plus petites que $\delta > 0$ pour un $\delta$ arbitraire. Ce nombre s'appelle le $\delta$-rang numérique [44] car c'est une extension de la notion de rang. C'est à partir de cette notion de rang numérique que l'on va définir le rang effectif. Le rang effectif peut ensuite remplacer le rang dans la définition du taux de synchronisation.

Il existe plusieurs définitions du rang effectif, qui peut être interprété comme le nombre effectif de valeurs singulières non nulles, c'est-à-dire le nombre de valeurs singulières suffisamment grandes pour être interprétées comme des valeurs propres non nulles. La plupart de ces définitions donnent le rang effectif comme étant une fonctionnelle des normes de Schatten de $\mathbf{A}$. Pour tout $s \geq 1$, la $s$-norme de Schatten de $\mathbf{M}$ est égale à la norme $l_s$ de ses valeurs singulières, c'est à dire :

$$\|\mathbf{A}\|_s^s = \sum_{i=1}^{p} \sigma_i^s(\mathbf{A}) \tag{1.9}$$

Les $s$-normes de Schatten généralisent certaines normes de matrices. Le cas $s = 2$ correspond à la norme euclidienne, appelée norme de Frobenius, de matrices. Son expression est également égale à la somme du carré des coefficients de $\mathbf{A}$.

$$\|\mathbf{A}\|_2^2 = \sum_{i=1}^{p} \sigma_i^2(\mathbf{A}) = \sum_{i=1}^{n}\sum_{j=1}^{p} \mathbf{A}_{i,j}^2.$$

Le cas $s = 1$ correspond à la norme nucléaire, ou norme Trace, et le cas $s = \infty$ correspond à la norme opérateur notée $\|.\|_\infty$ qui est égale à la plus grande valeur singulière de $\mathbf{A}$.

$$\|\mathbf{A}\|_\infty = \sigma_1(\mathbf{A}).$$

Comme nous considérons le cas où la matrice $\mathbf{A}$ se décompose comme la matrice $\mathbf{M}$ qui est a priori de faible rang et la matrice $\mathbf{\Delta}$, il est très probable que $\mathbf{A}$ possède quelques grandes valeurs singulières et beaucoup de petites valeurs singulières. Ce phénomène est connu et plusieurs définitions du rang effectif ont été introduites afin de pouvoir estimer le nombre de grandes valeurs singulières de $\mathbf{A}$. Un premier exemple si $\mathbf{A}$ est une matrice symétrique non négative : Koltchinskii et Lounici [66] proposent de définir le rang effectif comme le ratio $\frac{\|\mathbf{A}\|_1}{\|\mathbf{A}\|_\infty}$. Ce ratio équivaut à diviser la somme des valeurs singulières par la plus grande valeur singulière. Alors, si $\mathbf{A}$ est de rang 1, le rang et le rang effectif auront la même valeur. En se plaçant dans le même contexte que nos données, si la matrice $\mathbf{A}$ est de rang plein mais possède un petit nombre de grandes valeurs singulières et un grand nombre de petites valeurs singulières, alors le rang effectif de $\mathbf{A}$ sera très proche du rang de $\mathbf{M}$.
En général, le ratio d'une $s$-norme de Schatten et de la plus grande valeur singulière est une bonne approximation du rang pour les raisons citées précédemment. Roy et Vertelli [107] donnent une définition un peu plus générale du rang effectif à partir de l'entropie de Shannon qui peut se calculer pour des matrices rectangulaires.

$$\mathrm{ER}_{1,1}(\mathbf{A}) = \exp\left[-\sum_{i=1}^{q} \frac{\sigma_i(\mathbf{A})}{\|\mathbf{A}\|_1} \log\left(\frac{\sigma_i(\mathbf{A})}{\|\mathbf{A}\|_1}\right)\right] , \tag{1.10}$$

qui correspond au nombre de Shannon effectif de la distribution induite par le vecteur de probabilité $(\sigma_i(\mathbf{A})/\|\mathbf{A}\|_1)$. De façon plus générale, on peut calculer n'importe quel nombre effectif en se basant sur le vecteur de probabilités associé aux valeurs singulières de $\mathbf{A}$ ou celles de la matrice carrée $\mathbf{A}^T\mathbf{A}$. Le vecteur de probabilités associés à $\mathbf{A}^T\mathbf{A}$ est égal à $\left(\sigma_i^2(\mathbf{A})/\|\mathbf{A}\|_2^2\right)$. La justification d'une telle définition pour le rang effectif est détaillée dans [107].

Dans [60], l'auteur considère plusieurs mesures de la diversité. Parmi celles-ci, se trouve le nombre effectif de Hill, qui est directement dérivé de l'entropie de Renyi. L'entropie de Renyi d'ordre $\alpha \geq 0$, avec $\alpha \neq 1$, notée $H_\alpha(.)$ généralise plusieurs entropies :

$$H_\alpha(X) = \frac{1}{1-\alpha} \log\left(\sum_{i=1}^{n} p_i^\alpha\right), \tag{1.11}$$

où $X$ est une variable aléatoire à valeurs dans $\{1, \ldots, n\}$ telle que $p_i = \mathbb{P}(X = i)$.
Par exemple, quand $\alpha$ tend vers 1, la limite de $H_\alpha(X)$ est l'entropie de Shannon. Certaines valeurs de $\alpha$ renvoient à d'autres entropies connues comme la min-entropie ou l'entropie de Hartley, voir [102] pour plus de détails. Le nombre effectif de Hill, noté $D^q$ permet de quantifier la diversité des espèces dans un écosystème.

$$D^q = \left(\sum_{i=1}^{n} p_i^q\right)^{\frac{1}{1-q}} , \tag{1.12}$$

où $n$ est le nombre d'espèces, et $p_i$ sont des poids qui permettent de pondérer chaque espèce par l'amplitude de leurs abondances. Le paramètre $q$ détermine la sensibilité du nombre de Hill à

l'amplitude des abondances. Si $q$ est égal à 0, alors on ne prend en compte que la présence des espèces dans l'écosystème, et non pas l'amplitude de leurs abondances, alors qu'un $q$ élevé donne une grande importance aux amplitudes.

A partir du nombre effectif de Hill, il est possible de définir le rang effectif de la façon suivante pour la matrice $\mathbf{A}$ ou pour sa matrice carrée $\mathbf{A}^T\mathbf{A}$, en remplaçant le paramètre $q$ par une fonction dépendant de la $s$-norme de Schatten considérée et les probabilités $p_i$ par un ratio de normes de Schatten.

$$\mathrm{ER}_{1,s}(\mathbf{A}) = \left(\frac{\|\mathbf{A}\|_s}{\|\mathbf{A}\|_1}\right)^{s/(1-s)} \quad ; \qquad \mathrm{ER}_{2,s}(\mathbf{A}) = \mathrm{ER}_{1,s}(\mathbf{A}^T\mathbf{A}) = \left(\frac{\|\mathbf{A}\|_{2s}}{\|\mathbf{A}\|_2}\right)^{2s/(1-s)} . \tag{1.13}$$

Deux cas particuliers du rang effectif (2.5) sont $\mathrm{ER}_{1,\infty}(\mathbf{A})$ et $\mathrm{ER}_{2,\infty}$ qui peuvent être vus comme des extensions du rang effectif de Koltchinskii et Louni à des matrices rectangulaires.
Plus de détails peuvent être trouvés dans [60].

### 1.2.3 Une autre application

Réussir à estimer le rang effectif d'une matrice $\mathbf{A}$ à partir d'une observation bruitée $\mathbf{Y}$ comme dans (1.4) possède d'autres applications.

Un modèle de mélange gaussien est un modèle statistique où la distribution des observations est exprimée comme une combinaison convexe de lois normales. Nous nous plaçons dans le cas où les observations $\mathbf{y}_i \in \mathbb{R}^p$ pour $i$ entre 1 et $n$ sont des vecteurs gaussiens de moyenne $\mathbf{m_i}$ et de matrice de covariance $\mathbf{\Gamma_i}$. Soit $f_{\mathcal{N}}(\mathbf{y_i}, \theta_i)$ la densité d'un vecteur gaussien de paramètres $\theta_i = (\mathbf{m_i}, \mathbf{\Gamma_i})$ et soit un entier $K \geq 1$, un ensemble de paramètres $\theta_i, \ldots, \theta_K$ et un ensemble de poids $\pi_1, \ldots, \pi_K$ tels que $\sum_{k=1}^K \pi_k = 1$, alors la densité de l'échantillon suit une densité de mélange $g$ telle que pour $\mathbf{y_i} \in \mathbb{R}^p$:

$$g(\mathbf{y_i}, \theta_1, \ldots, \theta_K) = \sum_{k=1}^{K} \pi_k f_{\mathcal{N}}(\mathbf{y_i}, \theta_k). \tag{1.14}$$

On observe un échantillon de $n$ individus dans $\mathbb{R}^p$ tel que quel que soit $i$ entre 1 et $n$, chaque observation $\mathbf{y_i}$ appartient à un des $K$ groupes $G_1, \ldots, G_K$ d'observations. Si $\mathbf{y_i}$ appartient à $G_k$ pour un entier $k$ entre 1 et $K$, alors $\mathbf{y_i}$ est de moyenne $\mathbf{m_k}$ et de matrice de covariance $\mathbf{\Gamma}_k$ propres au groupe $G_k$. Nous supposons dans la suite que toutes les observations sont indépendantes et que la variance est la même pour tout l'échantillon et est notée $\sigma^2$. Alors, la matrice $\mathbf{Y} \in \mathbb{R}^{n\times p}$ des observations admet la décomposition suivante :

$$\mathbf{Y} = \mathbf{A} + \mathbf{E} \tag{1.15}$$

où $\mathbf{A}$ est une matrice déterministe dont la $i$-ème ligne est égale à $\mathbf{m_i}$ et $\mathbf{E}$ est une matrice dont les coefficients sont i.i.d et telle que $\mathbf{E}_{i,j}$ suit une loi normale centrée de variance $\sigma^2$.

La décomposition (1.15) nous renvoie à la décomposition (1.4). Dans le modèle de mélange idéal toutes les observations d'un même groupe partage exactement la même moyenne. Dans ce cas le rang de $\mathbf{A}$ est égal au nombre de classes dans le modèle de mélange. Pour les mêmes raisons que pour les séries temporelles d'abondances, il en est de même si les moyennes des observations au sein d'un même groupe sont exactement proportionnelles. Cependant, cela n'est pas toujours le cas et il est possible que $\mathbf{A}$ se décompose comme (1.5). Alors, l'estimation du rang effectif de $\mathbf{A}$ à partir de $\mathbf{Y}$ permettrait de déterminer le nombre de classes du modèles de mélange de $\mathbf{A}$ même si celle ci est perturbée comme dans (1.5).

**Objectifs.** S'il est possible d'estimer le rang effectif de $\mathbf{A}$ à partir de la matrice d'observation bruitée $\mathbf{Y}$, alors le taux de synchronisation de $\mathbf{A}$ est ensuite calculé comme le ratio entre le rang effectif de $\mathbf{A}$ et le nombre d'espèces $n$. Les abondances d'espèces sont des entiers et sont souvent modélisées par des lois de Poisson ou des lois de Binomiale Négative [7]. Il est ensuite possible, sous certaines conditions d'approcher de telles lois par des gaussiennes. C'est un sujet délicat que l'on n'abordera pas dans ce chapitre.

Dans la suite, nous nous plaçons dans le modèle gaussien présenté précédemment. Nous supposons que nous observons une matrice $\mathbf{Y} \in \mathbb{R}^{n \times p}$ telle que les observations $\mathbf{y_i}$ sont des vecteurs gaussiens de moyenne $\mathbf{m_i}$ et de matrice de covariance $\sigma\,\mathbf{Id}$ avec $\sigma$ connue. Alors $\mathbf{Y}$ admet la décomposition (1.15). Nous nous attaquons donc au problème général d'estimation du rang effectif de $\mathbf{A}$ grâce à la matrice d'observations bruitée $\mathbf{Y}$ dans le cas où $\mathbf{Y}$ se décompose comme dans (1.6) et où il est possible que $\mathbf{A}$ se décompose comme dans (1.5). A l'exception du rang effectif basé sur l'entropie de Shannon (2.4), toutes les autres mesures du rang effectif sont définies à partir de ratios de normes de Schatten de $\mathbf{A}$. Ce qui fait qu'estimer ces dernières nous permet d'estimer le rang effectif.

### 1.2.4 Etat de l'art

Si la matrice $\mathbf{A}$ était de faible rang, ce qui est le cas si $\mathbf{\Delta} = 0$, il est possible de retrouver la matrice $\mathbf{A}$ à partir de la matrice $\mathbf{Y}$ par des méthodes de seuillage des valeurs singulières, comme dans [21, 41]. Nous précisons que dans ce cas, c'est uniquement la matrice $\mathbf{E}$ qui nous empêche d'estimer directement le rang de $\mathbf{A}$. Nous considérons l'estimateur de $\mathbf{A}$ obtenu en fixant à 0 toutes les valeurs singulières de $\mathbf{Y}$ qui sont plus petites qu'un certain seuil. Dans [31] de Donoho et Gavish, les auteurs ont évalué le risque asymptotique de tels estimateurs. Si $n$ et le rang de $\mathbf{A}$ sont proportionnels à $p$, on atteint un risque quasi optimal. D'autres articles comme [93, 111] proposent des estimateurs calculés grâce à des transformations non linéaires des valeurs singulières qui ont un risque légèrement meilleur dans un cadre asymptotique où $\frac{n}{p}$ tend vers une constante.
Une fois l'estimateur de $\mathbf{A}$ obtenu, les $s$-normes de Schatten de cet estimateur peuvent être calculées et utilisées pour en déduire le rang effectif de $\mathbf{A}$.

Lorsque la matrice $\mathbf{\Delta}$ n'est pas nulle, la matrice $\mathbf{A}$ n'est pas de faible rang. Nous voulons alors estimer les $s$-normes de Schatten de $\mathbf{A}$ afin de construire un estimateur d'un des rangs effectifs définis précédemment. Plusieurs méthodes sont envisageables. Il est possible d'estimer directement $\|\mathbf{A}\|_s$ à partir des $\|\mathbf{Y}\|_s$. On peut également trouver un estimateur $\widehat{\mathbf{A}}$ de $\mathbf{A}$, puis définir un estimateur de $\|\mathbf{A}\|_s$ comme $\|\widehat{\mathbf{A}}\|_s$. Enfin, nous pouvons estimer le spectre de $\mathbf{A}$ puis remplacer les $\sigma_i(\mathbf{A})$ par leurs estimateurs dans (1.9).

Quelle que soit la méthode utilisée, il est nécessaire de pouvoir calculer le risque des estimateurs afin de pouvoir évaluer leur précision et les comparer. Le risque est une fonction permettant d'évaluer à quel point un estimateur est proche de la quantité qu'il a pour but d'estimer. Soit $\widehat{\theta}$ un estimateur de $\theta$, alors le risque quadratique de $\widehat{\theta}$ est défini comme :

$$R_2 = \mathbb{E}\left[\|\widehat{\theta} - \theta\|_2^2\right]. \tag{1.16}$$

La fonction $\|\widehat{\theta} - \theta\|_2^2$ peut être remplacée par une autre fonction de perte $l(.,.)$, par exemple, $l(\widehat{\theta}, \theta) = \|\widehat{\theta} - \theta\|_1$.
La définition générale du risque est la suivante. Soit $\theta$ une quantité que l'on veut estimer, $\widehat{\theta}$ un estimateur de $\theta$ et soit une fonction de perte $l(\widehat{\theta}, \theta)$ qui permet d'évaluer la différence entre $\theta$ et son estimateur, comme par exemple les moindres carrés ou la norme $L_1$ vus précédemment, alors le risque est défini comme $\mathbb{E}\left[l(\widehat{\theta}, \theta)\right]$.

Une fois le risque d'un estimateur calculé, nous voulons pouvoir vérifier si le risque est optimal ou s'il est possible de mieux faire. Parfois, il n'est pas possible d'estimer une certaine quantité avec un risque plus petit qu'un certain ordre de grandeur ou qu'une certaine constante. Pour formaliser cela, nous utilisons le risque minimax. Soit $\theta$ une quantité que l'on cherche à estimer, on suppose que $\theta$ appartient à un ensemble $\Theta$. Alors, le risque minimax $R_{\text{minimax}}(\Theta)$ donne le risque du meilleur estimateur possible dans le pire des cas pour $\theta \in \Theta$. C'est-à-dire :

$$R_{\text{minimax}}(\Theta) = \inf_{\widehat{\theta}} \sup_{\theta \in \Theta} \mathbb{E}_\theta \left[ l(\widehat{\theta}, \theta) \right] \tag{1.17}$$

pour la fonction de perte $l(.,.)$ utilisée et où $\mathbb{E}_\theta[\cdots]$ est la fonction espérance sous la loi de paramètre $\theta$. D'autres définitions et notions sur les estimateurs peuvent être trouvées dans [42].

La plupart des travaux autour du rang effectif considèrent des modèles sans bruit, comme par exemple Roy et Veterlli dans [107] qui estime le rang de la matrice $\mathbf{A}$ dans le cas où cette matrice n'est que partiellement observée. Dans un autre registre, en informatique il existe un domaine de recherche concernant le Matrix Sketching. Le sketching est une procédure qui permet de traiter des données en grande dimension en n'interrogant qu'une faible partie de ces données. Dans [75], un sketch linéaire est décrit comme une distribution sur des matrices $\mathbf{S}$ de taille $n \times k$ telles que quel que soit un vecteur $v$, il est possible d'approximer une fonction $f(v)$ à partir de l'information $\mathbf{A}v$, l'objectif étant d'avoir $k$ le plus petit possible. Dans le cas des matrices, on veut pouvoir retrouver $\|\mathbf{A}\|_s$ en ayant accès à $\mathbf{TAS}$ pour des matrices $\mathbf{T}$ et $\mathbf{S}$ ou en ayant accès à $L(\mathbf{A})$, où $L(.)$ est une fonction linéaire. Une méthode de sketching peut être utilisée pour retrouver des fonctionnelles de $\mathbf{A}$, comme sa $s$-norme de Schatten ou son rang. De telles méthodes sont décrites dans plusieurs articles comme [6, 63, 75] mais elles sont très éloignées de notre procédure.

Sachant que les $s$-normes de Schatten de $\mathbf{A}$ s'expriment comme la $l_s$ norme des valeurs singulières de $\mathbf{A}$, si nous trouvons un estimateur consistant des valeurs singulières de $\mathbf{A}$, il est possible de construire un estimateur de $\|\mathbf{A}\|_s^s$ en insérant les estimateurs des $\sigma_i^s(\mathbf{A})$ dans (1.9). Il existe dans la littérature un nombre certain d'articles ayant pour but d'estimer asymptotiquement le spectre de matrices. Dans la plupart de ces articles, les auteurs se placent dans le cas où $\frac{n}{p}$ tend vers une constante. Dans [25], notre modèle (1.15) correspond au modèle information plus bruit. Soit la matrice symétrique $\frac{1}{p}\mathbf{Y}^T\mathbf{Y}$ dont les valeurs propres ordonnées sont notées $\lambda_1, \ldots, \lambda_p$, la mesure spectrale est définie comme $\mu_{\mathbf{Y}^T\mathbf{Y}} := \frac{1}{p}\sum_{i=1}^p \delta_{\lambda_i}$. Si $\mathbf{A} = 0$, il a été montré dans [2] que $\mu_{\mathbf{Y}^T\mathbf{Y}}$ converge faiblement vers la distribution de Marchenko-Pastur qui décrit le comportement asymptotique du spectre de matrices rectangulaires. Voir [86] pour plus d'informations sur la distribution des valeurs singulières de matrices aléatoires.
Plus généralement, quand la mesure spectrale $\frac{1}{p}\mathbf{A}^T\mathbf{A}$ converge vers une mesure de probabilité, [123] a caractérisé la limite de $\mu_{\mathbf{Y}^T\mathbf{Y}/p}$ à travers sa transformée de Stieljes. A partir de là, il peut être judicieux d'inverser la fonction qui associe la limite de $\mu_{\mathbf{Y}^T\mathbf{Y}/p}$ à la limite de $\mu_{\mathbf{A}^T\mathbf{A}/p}$ pour estimer des fonctionnelles du spectre de la matrice carrée $\mathbf{A}^T\mathbf{A}$ à partir de $\mathbf{Y}$. Comme il est dit dans [25], si $\mathbf{A}$ a asymptotiquement un nombre fini $k$ de valeurs singulières, Il est possible en déduire la consistance de ces valeurs singulières.
Cependant, les conditions nécessaires pour avoir des estimateurs consistants des valeurs singulières ne sont pas remplies dans notre cas de figure. Il n'est donc pas possible de suivre la même méthodologie.

Les $s$-normes de Schatten étant des fonctionnelles non linéaires des coefficients de $\mathbf{A}$, il est aussi possible de se référer et de s'inspirer d'articles cherchant à estimer des fonctionnelles non linéaires. Par exemple, en vectorisant les matrices $\mathbf{Y}$, $\mathbf{A}$ et $\mathbf{E}$, il est possible de réécrire le modèle (1.15) comme un modèle de suites gaussiennes dont la définition peut être trouvée dans l'introduction de [59]. Estimer la norme de Frobenius est équivalent à estimer la norme $l_2$ dans le modèle des suites gaussiennes. Les articles [23, 32] et leurs références présentent des résultats sur l'estimation de fonctionnelles quadratiques. Plus généralement, l'estimation de normes $l_r$ de vecteurs est un sujet

largement étudié, par exemple dans [18, 48, 72]. Cependant, hormis dans le cas de la norme euclidienne qui correspond à $s = 2$, la $s$-norme de Schatten n'est pas une $l_r$ norme des coefficients de $\mathbf{A}$ et il n'est pas possible d'adapter l'analogie entre le modèle (1.15) et le modèle de suites gaussiennes à d'autres $s$-normes de Schatten.

Parmi les travaux les plus proches de notre modèle et de la structure de nos données, Kong et Valiant dans [67] considèrent l'estimation des normes de Schatten paires de la matrice de covariance $\Sigma$ dans le cas où l'on observe un $n$-échantillon de moyenne nulle et de matrice de covariance $\Sigma$. Ils proposent des estimateurs non biaisés et peu coûteux à implémenter mais ils n'évaluent pas l'optimalité de leurs estimateurs en termes de risque.

### 1.2.5 Notre contribution - résultats

L'article qui constitue le chapitre sur les normes de Schatten présente dans un premier temps des estimateurs de la norme de Frobenius de $\mathbf{A}$ à partir de $\mathbf{Y}$. L'estimateur $(\|\mathbf{Y}\|_2^2 - np)_+^{1/2}$, avec $x_+ = \max(x, 0)$ a un risque optimal de l'ordre de $(np)^{1/4}$. Cet ordre de grandeur du risque ne peut être amélioré par d'autres estimateurs, même si le rang de $\mathbf{A}$ est inférieur ou égal à 1. Cette minoration du risque minimax est généralisée à toutes les $s$-normes de Schatten. Ensuite, nous montrons que l'on peut estimer $\|\mathbf{A}\|_\infty = \sigma_1(\mathbf{A})$ avec une transformation non linéaire de $\sigma_1(\mathbf{Y})$ et que le risque de cet estimateur est également de l'ordre de $(np)^{1/4}$.

Dans le cas plus général des normes de Schatten paires $\|\mathbf{A}\|_{2k}$, $\|\mathbf{A}\|_{2k}^{2k} = \mathrm{tr}[(\mathbf{A}^T\mathbf{A})^{2k}]$ est un polynôme en les entrées de $\mathbf{A}$. Cela nous permet de construire un estimateur non biaisé $U_k$ de $\|\mathbf{A}\|_{2k}^{2k}$ grâce aux polynômes de Hermite. L'invariance de $\|\mathbf{A}\|_{2k}^{2k}$ par des transformations orthogonales droite ou gauche nous permet d'établir que $U_k$ peut s'exprimer simplement comme une combinaison algébrique de $\mathrm{tr}[(\mathbf{YY})^s]$. Enfin, nous montrons que l'estimateur $(U_k)_+^{1/2k}$ atteint un risque optimal de l'ordre de $(np)^{1/4}$ pour n'importe quelle matrice $\mathbf{A}$.

Finalement, nous donnons quelques résultats partiels pour les normes de Schatten qui ne sont pas paires et qui sont beaucoup plus compliquées à estimer. L'estimateur proposé atteint un risque de l'ordre de $p(np)^{1/4}$. Nous prouvons ensuite que dans le cas de la norme Trace $\|\mathbf{A}\|_1$, il n'est pas possible de construire un estimateur dont le risque est de l'ordre de $(np)^{1/4}$ et qu'aucun estimateur n'atteint un risque plus petit que $p/\sqrt{\log(p)}$.

Comme application des travaux présentés dans l'article, nous construisons un estimateur du rang effectif $\mathrm{ER}_{2,\infty}$ et calculons son erreur en probabilité. De plus, nous évoquons les raisons qui font que les mesures du rang effectif $\mathrm{ER}_{2,s}$ où $s \geq 2$ est un entier, sont plus faciles à estimer que les autres mesures (1.10,1.13).

## 1.3 Tree Segmentation

### 1.3.1 Motivations

Le deuxième chapitre de ma thèse comprend des éléments théoriques et des éléments algorithmiques. Deux sections seront dédiées à des applications des travaux réalisés sur des données numériques simulées et sur des données réelles de pêche sur le Bassin de la Loire.

Afin d'introduire les motivations des travaux réalisés dans ce chapitre, je vais d'abord rappeler des notions de base sur les graphes et les arbres.
Un graphe $\mathcal{G}$ est constitué d'un ensemble de noeuds $\mathcal{V}(\mathcal{G})$ et d'un ensemble d'arêtes $\mathcal{E}(\mathcal{G})$ qui relient certains noeuds à d'autres, $\mathcal{G} = (\mathcal{V}(\mathcal{G}), \mathcal{E}(\mathcal{G}))$. Par convention, les noeuds sont souvent indexés par

des entiers allant de 1 à $n$ la taille du graphe, c'est à dire le nombre de noeuds dans $\mathcal{V}$, $n = |\mathcal{V}|$. Les arêtes peuvent être orientées ou non et sont souvent représentées par un vecteur de taille 2 qui contient les deux noeuds reliés.

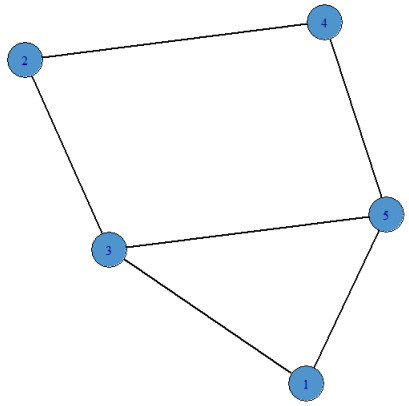



Figure 1.2 – Un exemple de graphe quelconque de taille 5. Les noeuds sont reliés par des arêtes. Ici, les noeuds sont numérotés avec des entiers de 1 à 5.

Un cycle est un chemin tel que son origine et son extrémité sont les mêmes. Il existe une arête entre chaque noeud consécutif et entre le noeud $n$ et le noeud 1. Un graphe est acyclique s'il ne comporte pas de cycle.
Un arbre T est un graphe acyclique et est connexe, c'est-à-dire que quels que soient deux noeuds dans $\mathcal{V}$(T), il existe un chemin d'arêtes de $\mathcal{E}$(T) qui relie ces deux noeuds.

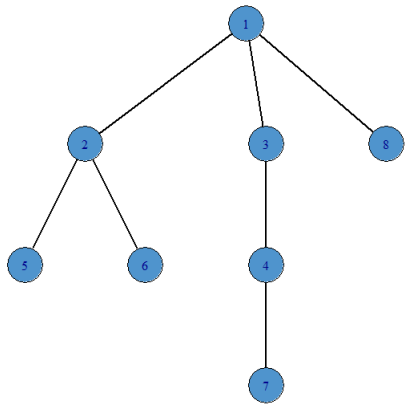



Figure 1.3 – Un exemple d'arbre de taille 8. Les noeuds sont reliés par des arêtes. Les noeuds sont numérotés avec des entiers de 1 à 8. Le graphe est connexe et il n'y a pas de cycle.

Le graphe chaîne est un autre cas particulier des arbres, l'ensemble de ses noeuds est $\{1, \ldots, n\}$

et l'ensemble de ses arêtes $\mathcal{E} = \{\{i, i+1\}, i = 1, \ldots, n-1\}$.

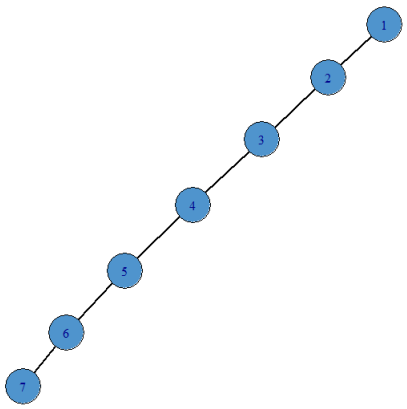



Figure 1.4 – Un exemple de chaîne de taille 7. Les noeuds sont reliés par des arêtes. Les noeuds sont numérotés avec des entiers de 1 à 7.

Soit T un arbre, un sous-arbre $S$ de T est un arbre tel que $\mathcal{V}(S)$ est inclus dans $\mathcal{V}(\mathrm{T})$ et tel que $\mathcal{E}(S) = \{e = (s,t) | e \in \mathcal{E}(\mathrm{T}); s, t \in \mathcal{V}(S)\}$. Comme $S$ est un arbre, les ensembles $\mathcal{V}(S)$ et $\mathcal{E}(S)$ sont tels que $S$ est connexe.

Soit T un arbre enraciné, nous définissons un ordre $\preceq$ sur les sommets de T de telle façon que les enfants d'un noeud $t$ sont les noeuds $s \in \mathcal{V}(\mathrm{T})$ tels que $(t, s)$ appartient à $\mathcal{E}(\mathrm{T})$ et tels que $t \preceq s$. Par exemple dans la Figure 1.3, les enfants de 1 sont 2,3 et 8.
Les descendants d'un noeud $t$ sont tous les noeuds $s$ tels que $t \preceq s$ et il existe un chemin d'arête entre $s$ et $t$ 1.5

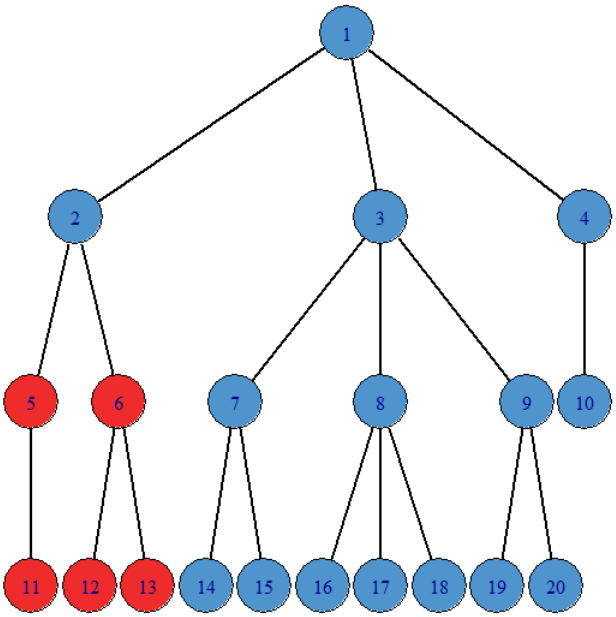



Figure 1.5 – Représentation d'un arbre de racine 1. Les descendants du noeud 2 sont représentés en rouge, les autres noeuds sont représentés en bleu.

Le sous-arbre de T engendré par $t$ est noté $\mathrm{T}_t$. L'ensemble de ses noeuds est constitué de $t$ et de tous ses descendants. l'ensemble de ces arêtes sont les arêtes de T qui relient deux noeuds appartenant $\mathcal{V}(\mathrm{T}_t)$, voir la Figure 1.6.

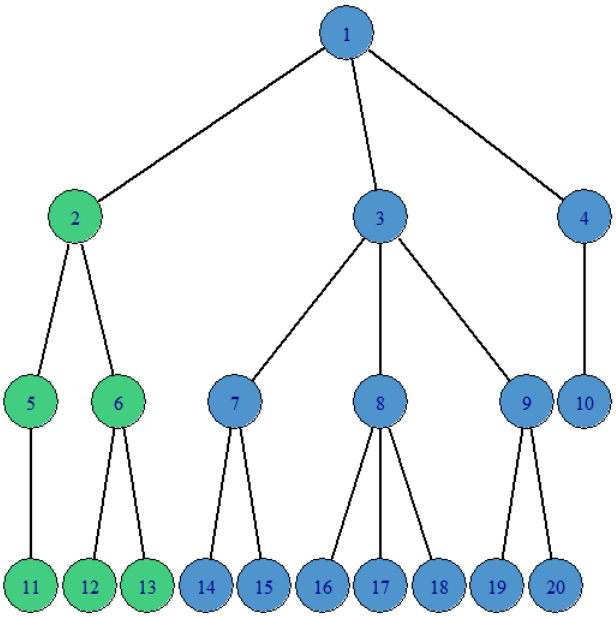
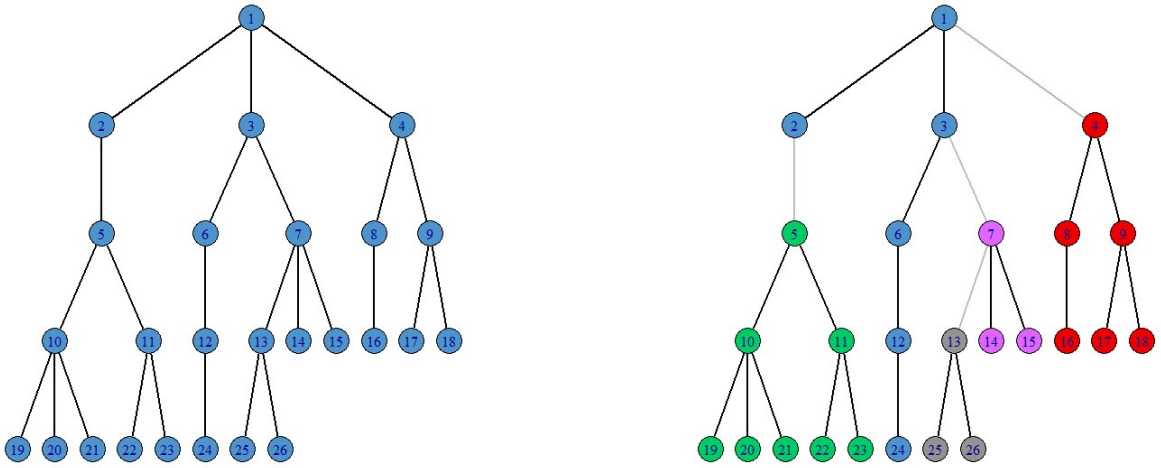



Figure 1.6 – Représentation d'un arbre de racine 1. Les noeuds du sous-arbre engendré par 2 sont représenté en vert et les autres noeuds en bleu. Le sous-arbre engendré par 2 se définit comme $\mathcal{V}(T_2) = \{2, 5, 6, 11, 12, 13\}$ et $\mathcal{E}(\mathrm{T}_2) = \{(2, 5), (2, 6), (5, 11), (6, 12), (6, 13)\}$.

Une partition $\mathbf{P} = \{S_1, \ldots, S_K\}$ de T est appelée une segmentation de T si tous les $S_j$ sont des sous-arbres de T. Dans l'ensemble du document, toutes les partitions considérées sont des segmentations. On note $\mathcal{P}(\mathrm{T})$ la collection des partitions de T. Soit une partition $\mathbf{P} \in \mathcal{P}(\mathrm{T})$, sa taille $|\mathbf{P}|$ est égale à son nombre de sous-arbres. Il existe une équivalence entre une partition $\mathbf{P} = \{S_1, \ldots, S_K\}$ de T en $K$ sous-arbres et l'ensemble des arêtes $(s, t)$ telles que $s$ et $t$ n'appartiennent pas au même sous-arbre $S_i$. Cet ensemble d'arêtes associé à la partition $\mathbf{P}$ est noté $\mathcal{J}_{\mathbf{P}}$. On montre dans le chapitre sur la segmentation d'arbre que comme T est un arbre, le nombre de ruptures $|\mathcal{J}_{\mathbf{P}}|$ est égal à $K - 1$. Un exemple de partition est illustré dans la Figure 1.7.



(a) (b)

Figure 1.7 – La figure (a) représente un arbre de racine 1. Tous les noeuds sont représentés en bleu. La figure (b) représente une partition $\mathbf{P}$ du même arbre. L'ensemble des noeuds d'une même couleur et des arêtes reliant ces noeuds forment les sous-arbres de $\mathbf{P}$. Les arêtes grisées représentent les ruptures associée à $\mathbf{P}$. Ce sont les arêtes de l'ensemble $\mathcal{J}_{\mathbf{P}}$.

Nous nous plaçons dans un contexte où l'on observe sur les noeuds d'un arbre T de taille $n$ des observations $y_s$ dont la distribution dépend totalement ou en partie du noeud $s$ sur lequel elle est observée. On suppose qu'il existe une partition $\underline{\mathbf{P}}$ de T telle que les observations d'un même sous-arbre $S_i \in \underline{\mathbf{P}(\mathrm{T})}$ suivent la même distribution.
L'objectif du chapitre 2 est d'implémenter un algorithme capable de retrouver la vraie partition $\underline{\mathbf{P}}$ de l'arbre T à partir de la structure de T et des observations $y_s$ pour $s \in \{1, \ldots, n\}$.

Je vais exposer les objectifs écologiques pour mettre en lumière le lien entre l'étude de la synchronie spatiale et la segmentation d'arbre. On s'intéresse au cas où l'on observe un signal réparti sur un arbre T. Dans le cas des abondances d'espèces, on suppose qu'il existe un certain nombre de sites d'observations répartis sur un territoire. Sur chaque site d'observation, on a accès à la série temporelle des abondances d'une espèce sur $p$ années. Dans certains cas, il est possible, voire nécessaire, de représenter les sites d'observations sous la forme d'un arbre T où chaque noeud est un site. C'est le cas, par exemple, pour un réseau de rivières. L'objectif, d'un point de vue écologique, consiste à étudier la synchronie spatiale d'une espèce sur l'arbre T. Pour cela, nous voulons trouver dans l'arbre T des sous-arbres sur lesquels les observations sont homogènes dans le sens où les séries temporelles sur deux sites du sous-arbre sont synchrones.
Au premier abord, on peut envisager d'utiliser des algorithmes de clustering classiques sur T afin de diviser les sites en groupes homogènes. Pour cela, il faudrait définir une matrice de similarité, ce qui peut être fait avec la corrélation de Spearman, comme évoqué dans la première partie de l'introduction. Cependant, nous ne voulons pas considérer les sites comme des observations sans disposition particulière. Nous avons structuré les données sous la forme d'un arbre, ce qui complique la tâche remplie par des algorithmes de clustering qui minimisent des distances entre les observations. Il faut ajouter des contraintes et interpréter la structure de l'arbre de façon à implémenter un algorithme capable de retrouver des groupes homogènes tout en respectant la structure des données. Il existe des articles qui s'intéressent au clustering sur les arbres, le problème est abordé comme l'optimisation d'une fonction analogue à une fonction de coût [85]. Dans [85], l'algorithme atteint un coût polynomial pour des arbres simples mais est NP dans des cas critiques comme le graphe étoile. Cependant, les auteurs n'utilisent pas vraiment de modèle sur les arbres et les fonctions de coût diffèrent des nôtres.

Nous modélisons les séries temporelles des abondances sur les sites, par exemple par des lois de Poisson. Sur le site $s$, les abondances de l'espèce suivent une loi de Poisson dont la moyenne pour chaque année $j$ est stockée dans un $p$-vecteur moyenne $\mathbf{m}_s$.

$$y_{s,j} \sim \mathcal{P}\left(\mathbf{m}_s^{(j)}\right), \tag{1.18}$$

où $\mathbf{m}_s^{(j)}$ est le $j$-ième coefficient de $\mathbf{m}_s$.

la matrice de taille $n \times p$ dont les lignes sont les vecteurs $\mathbf{m}_s$ pour tout $s \in \mathcal{V}(\mathrm{T})$ est notée $\mathbf{m}$. Nous supposons que si l'espèce considérée est synchrone sur deux sites de l'arbre T, alors les variations des paramètres des lois seront arbitrairement proches sur ces deux sites. Dans un cas idéal, les variations des abondances de l'espèce sur deux sites synchrones sont proportionnelles. Même en se plaçant dans le cas idéal, nous savons que la variance des lois de Poisson est élevée dans le sens où elle est égale à la moyenne de la loi. Il n'est donc pas toujours évident de déterminer si les variations des moyennes des lois de deux observations est la même. De plus, dans le cas des données réelles, il faut ajouter un bruit qui est dû au protocole de collecte des données, mais également prendre en compte les particularités environnementales de chaque site, ce qui rend la comparaison des séries temporelles sur deux sites différents délicate. Nous n'allons pas modéliser le bruit lié au protocole de collecte des données. Mais, pour rendre compte d'un "effet site" qui représente les variations des abondances d'espèces qui sont dues aux particularités de chaque site $s$, nous adaptons la forme de la moyenne de la distribution des observations.

$$\mathbf{m}_s^{(j)} = \mathrm{e}^{\alpha_s + \mathbf{b}_{s,j}} \tag{1.19}$$

où $\alpha_s$ est un effet site qui ne dépend que de $s$, et $\mathbf{b}_{s,j}$ est un paramètre qui dépend de l'année $j$ et du site $s$ et représente les variations d'abondances.
Notre objectif est de trouver une partition des $n$ noeuds de T en $K$ sous-ensembles sur lesquels les séries temporelles d'abondances sont synchrones. Pour nous, cela revient à trouver une partition de T telle que, quelle que soit l'année $j$, le paramètre $\mathbf{b}_{s,j}$ est le même pour tous les sites du sous-arbre. Soit $\underline{\mathbf{P}} = \left(\underline{S_1}, \ldots, \underline{S_K}\right)$ la vraie partition des observations de T en $K$ groupes de sites sur lesquels les séries temporelles d'abondances sont synchrones, alors les sites $s$ faisant partie d'un même sous-arbre $\underline{S_i}$ partagent le même paramètre $\mathbf{b}_{s,j}$. Le paramètre $\mathbf{b}_{s,j}$ ne dépend plus du site, mais du sous-arbre $\underline{S_i}$ de la vraie partition de T auquel $s$ appartient. Soit $\mathbf{P} = \{S_1, \ldots, S_K\}$ une partition de taille $K$ de T alors la log-vraisemblance $l(\mathbf{y}, \mathbf{m}, \mathbf{P})$ s'écrit :

$$l(\mathbf{y}, \mathbf{m}, \mathbf{P}) = \sum_{S_i \in \mathbf{P}} \sum_{s \in S_i} \sum_{j=1}^{p} -\log(y_{s,j}!) - \mathrm{e}^{\alpha_s + \mathbf{b}_{S_i,j}} + y_{s,j}\left(\alpha_s + \mathbf{b}_{S_i,j}\right) \tag{1.20}$$

En dérivant selon les paramètres, nous trouvons que les $\widehat{\alpha}_s$ et les $\widehat{\mathbf{b}}_{S_i,j}$ qui minimisent la vraisemblance (1.20) admettent les expressions suivantes :

$$\begin{aligned} \widehat{\alpha}_s &= \frac{\sum_{y=1}^{p} y_{s,j}}{\sum_{y=1}^{p} \mathrm{e}^{\widehat{\mathbf{b}}_{S_i,j}}} \\ \widehat{\mathbf{b}}_{S_i,j} &= \frac{\sum_{s \in S_i} y_{s,j}}{\sum_{s \in S_i} \mathrm{e}^{\alpha_s}} \end{aligned} \tag{1.21}$$

Comme le problème est sur-paramétré, il faut ajouter une contrainte sur les paramètres, sinon il existe une infinité de solutions. La contrainte prend la forme $\sum_{s \in S_i} \mathrm{e}^{\mathbf{b}_{S_i,j}} = 1$ pour tout $j$ entre 1 et $p$. Cela donne finalement une formule explicite pour la log-vraisemblance (1.20). On remarque que celle-ci dépend principalement de la partition $\underline{\mathbf{P}}$ et il en est de même pour les paramètres $\widehat{\alpha}_s$ et $\widehat{\mathbf{b}}_{S_i,j}$. Comme les paramètres dépendent de la partition, la matrice contenant les paramètres obtenus avec la partition $\mathbf{P}$ est notée $\mathbf{m}(\mathbf{P})$. Nous montrons dans le chapitre dédié qu'en choisissant

judicieusement une fonction de pénalité pen($\mathbf{P}$) qui pénalise la taille des partitions, la partition optimale $\underline{\mathbf{P}}$ est la partition qui minimise moins la log-vraisemblance pénalisée.

$$\underline{\mathbf{P}} = \min_{\mathbf{P}\in\mathcal{P}(\mathrm{T})} -l\left(\mathbf{y}, \mathbf{m}, \mathbf{P}\right) + \mathrm{pen}(\mathbf{P}). \tag{1.22}$$

J'ai pris la décision d'illustrer le processus en choisissant le modèle (1.18) avec $\mathbf{m}_s$ selon (1.19), mais des résultats similaires sont obtenus avec d'autres modèles, comme une loi Binomiale négative, ou même des lois continues comme des lois gaussiennes. Dans le cas gaussien, ce sont les moindres carrés pénalisés qu'il faut minimiser sur les partitions possibles de T.

Il est possible de caractériser une partition de T comme l'ensemble de ses sous-arbres ou comme un ensemble d'arêtes qui représentent les ruptures de distributions entre les sites. Il est donc équivalent de chercher la partition $\underline{\mathbf{P}}$ de l'arbre T qui minimise la log-vraisemblance (1.20) ou de chercher les ruptures dans la distribution de l'échantillon correspondant à la même partition.
Notre problème se ramène alors à un problème de détection de ruptures, sujet que l'on trouve abondamment dans la littérature. Cependant, si la détection de ruptures sur les chaînes est un problème largement étudié, il n'existe quasiment aucun article sur la détection de ruptures dans le cas plus général des arbres. Pourtant, certaines données ne peuvent pas être représentées linéairement et développer un algorithme capable de détecter les ruptures dans la distribution d'observations sur les arbres est utile pour traiter des données comme les arbres phylogénétiques ou pour l'analyse d'images et de vidéos [65, 91].

Pour revenir à l'étude de la synchronie spatiale, la détection de ruptures sur les arbres nous permet d'étudier sur un territoire les zones de synchronie d'une espèce et, éventuellement, grâce à l'expertise d'écologues, de déterminer les facteurs géographiques responsables de la synchronie spatiale, ou des ruptures dans la distribution d'un échantillon.

### 1.3.2 Etat de l'art

Comme évoqué plus haut, il existe une très large littérature dans le cas de la détection de rupture sur les chaînes. Commençons par formaliser le problème et ses notations.
On se place dans le cas où T est une chaîne. On observe un processus $y_s \in \mathcal{Y}$ qui est indexé par les noeuds de T. L'objectif est de trouver la partition $\underline{\mathbf{P}}$ de T telle que les observations d'un même sous-arbre de $\underline{\mathbf{P}}$ ont la même distribution. Le modèle que l'on a introduit pour la synchronie spatiale est une variante d'un problème spécifique au sein de la détection de rupture qui cherche à détecter des ruptures dans la moyenne des observations. Dans le cas gaussien, le modèle correspondant à ce problème est le suivant :

$$y_s = \mu_s + \epsilon_s, \tag{1.23}$$

où $\mu_s$ est la moyenne de la loi de $y_s$ et $\epsilon_s$ une variable aléatoire normale centrée et de variance $\sigma$ connue.

Dans le cas des chaînes, les sous-arbres de T sont des segments de la forme $[(\tau_s + 1); \tau_{s+1}]$ avec $\tau_0 = 0$ et $\tau_K = n$ et les ruptures sont les arêtes $\{\tau_s, \tau_s + 1\}$ for $s = 1, \ldots, K-1$. Donc $\mathcal{P}$ note la collection de partitions de $\{1, \ldots, n\}$ en plusieurs segments.

Au fil du temps et des articles, de nombreuses méthodes visant à résoudre le problème de la détection de ruptures sur les chaînes ont vu le jour. Deux difficultés principales se posent :

- Le nombre de ruptures dans la vraie partition de T est le plus souvent inconnu.
- Il existe un grand nombre de possibilités pour les emplacements des ruptures dans T.

Pour un arbre T de taille $n$, il existe $\binom{n-1}{K-1}$ possibilités pour placer les $K$ ruptures parmi les arêtes de T. Si en plus la valeur de $K$ est inconnue, alors il faut tester $2^{n-1}$ partitions différentes et les comparer. Cela donne une complexité exponentielle pour un algorithme naïf qui testerait toutes les partitions possibles et choisirait la meilleure d'entre elles selon un critère choisi. La plupart des méthodes développées pour pallier ce problème font face à un dilemme entre une complexité algorithmique relativement faible, jusqu'à $n \log n$ pour la Binary Segmentation [39], et l'assurance de trouver la partition optimale de la chaîne T.

Je vais désormais introduire des méthodes classiques de résolution du problème de détection de ruptures sur les chaînes.

La segmentation Binaire, ou Binary Segmentation est une procédure de dichotomie gloutonne qui a été introduite par [124]. L'algorithme coupe récursivement l'arbre en deux en minimisant un certain critère, comme CUSUM [39]. La plupart du temps, cette méthode est utilisée dans le cas de la détection de ruptures de la moyenne (1.23) mais il est également possible d'étendre son utilisation à la détection de ruptures dans la covariance des observations comme dans [127]. Un critère d'arrêt est sélectionné afin d'arrêter l'algorithme quand il ne détecte plus de rupture significative dans la distribution des données. Comme une fois qu'une rupture a été détectée par l'algorithme, elle ne peut plus être enlevée de la partition qui est en cours de construction, il est possible que l'algorithme ne produise qu'une partition sous-optimale de l'arbre. Fryzlewicz propose une modification de l'algorithme dans [39] appelé WBS pour Wild Binary Segmentation qui s'avère être consistent.

Toujours dans le cas de la détection de ruptures dans la moyenne des observations, Tibshirani et al. [118] proposent une procédure de Lasso fusionné, Fused Lasso en anglais, qui a pour but de minimiser le critère pénalisé :

$$\widehat{\theta}_\lambda = \arg\min_{\theta \in \mathbb{R}^n} \sum_{i=1}^{n} (y_i - \theta_i)^2 + \lambda \sum_{i=1}^{n-1} |\theta_{i+1} - \theta_i| \ , \tag{1.24}$$

où $\lambda$ est un paramètre de pénalisation. La quantité $\widehat{\theta}_\lambda$ obtenue donne les ruptures de la partition optimale. Le critère (1.24) est convexe et il est prouvé qu'on peut le calculer en un temps quasi-linéaire. La pénalité $\lambda \sum_{i=1}^{n-1} |\theta_{i+1} - \theta_i|$ oblige le vecteur $\widehat{\theta}$ à être constant par morceaux. Si les ruptures de la vraie partition sont réparties régulièrement le long de la chaîne T, il est montré dans [47] que l'on retrouve la vraie partition des observations à une vitesse proche de l'optimale. Cependant, lorsque cette contrainte sur la répartition des ruptures n'est pas respectée, la procédure de Fused Lasso est moins performante que d'autres algorithmes qui seront introduits dans la suite. Il existe d'autres méthodes qui sont référencées pour la plupart dans un papier récent [121] de Truong. Nous n'avons présenté ici que deux méthodes parmi les plus populaires mais nous pouvons également mentionner le kernel change-points algorithm [4, 40].

La méthode qui nous intéresse particulièrement et que l'on souhaite adapter au cas des arbres s'appelle procédure de minimisation de coût, Minimization Cost Procedure en anglais. Comme l'objectif du chapitre est d'implémenter un algorithme de segmentation sur les arbres basé sur la minimisation de coût, je vais introduire les notations et les définitions qui lui sont associées dans le cas des arbres. Je préciserai en plus certains changements de notations dans le cas des chaînes, dans le but d'être cohérente avec les notations qui existent dans la littérature.

L'objectif principal de la procédure de minimisation de coût est de minimiser sur toutes les partitions possibles $\mathbf{P}$ de T, une fonction de coût notée $\mathrm{Cost}_\mathrm{T}(\mathbf{P})$ et qui est linéaire en les sous-arbres de la partition $\mathbf{P}$.

$$\mathrm{Cost}_\mathrm{T}(\mathbf{P}) = \sum_{S_i \in \mathbf{P}} \mathrm{C}_{S_i} \, . \tag{1.25}$$

où $\mathrm{C}_{S_i}$ est le coût d'un sous-arbre $S_i \in \mathbf{P}$. On note $K_{\mathbf{P}}+1$ la taille de la partition $\mathbf{P}$. La fonction de coût doit être déterminée à l'avance. Dans le cas des séries temporelles d'abondances, il est possible de choisir $\mathrm{Cost}_{\mathrm{T}}(\mathbf{P})$ égal à moins la log-vraisemblance de l'échantillon observé. Dans ce cas, le coût d'un segment $S_i$ peut être égal à moins la log-vraisemblance des observations sur $S_i$. Les moindres carrés sont aussi souvent utilisés dans les procédures de minimisation de coût.
En général, on est libre de choisir la fonction de coût qui correspond le mieux aux données, mais cette fonction doit respecter (1.25) ainsi que la propriété suivante.
Soit T un arbre et $S_1$ et $S_2$ deux sous-arbres d'intersection nulle de T tels que leur union est égale à T, alors la fonction de coût doit respecter

$$\mathrm{C}_{\mathrm{T}} \geq \mathrm{C}_{S_1} + \mathrm{C}_{S_2}\,. \tag{1.26}$$

En théorie, il faut choisir une fonction de coût qui est minimisée par la vraie partition des observations, par exemple les moindres carrés ou moins la log-vraisemblance. Cependant, si la fonction de coût respecte la propriété (1.26), alors si une rupture est ajoutée à une partition $\mathbf{P}$ de T, le coût de la nouvelle partition est toujours inférieur à l'ancienne. Cela implique que la partition qui minimise le coût $\mathrm{Cost}_{\mathrm{T}}(\mathbf{P})$ est la partition où chaque arête est une rupture.
Pour pallier ce problème, deux solutions existent. La première consiste à trouver la partition $\mathbf{P}$ de taille $K$ qui minimise la fonction $\mathrm{Cost}_{\mathrm{T}}(\mathbf{P})$, pour $K$ entre 1 et un entier $K_{\max} \leq n$. Ensuite, la partition qui semble la plus proche de la vraie partition des observations est choisie parmi toute celles qui ont été trouvées. Ce problème est appelé le cas contraint. On note $\mathbf{P}^*_K(\mathrm{T})$ la partition optimale de T en $K$ sous arbre.

$$\mathbf{P}^*_K(\mathrm{T}) = \min_{\mathbf{P}\in\mathcal{P}||\mathbf{P}|=K} \mathrm{Cost}_{\mathrm{T}}(\mathbf{P}). \tag{1.27}$$

La deuxième solution consiste à ajouter une pénalité $\mathrm{pen}(\mathbf{P})$ au coût que l'on cherche à minimiser afin de contrôler la taille de la partition optimale créée. Ce problème est appelé cas pénalisé. On note $\mathbf{P}^*(\mathrm{T})$ la partition optimale de T obtenue.

$$\mathbf{P}^*(\mathrm{T}) = \min_{\mathbf{P}\in\mathcal{P}} \mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) + \mathrm{pen}(\mathbf{P}). \tag{1.28}$$

Cela nous renvoie à notre problème sur la synchronie spatiale où l'objectif est de trouver la partition qui minimise la log-vraisemblance pénalisée ou contrainte des observations sur T. Dans le cas des chaînes, les sous-arbres sont des segments $[\tau_i + 1 : \tau_{i+1}]$, on peut donc remplacer les sous-arbres $S_i$ par des segments dans (1.25) et (1.26). De plus, l'union de deux segments est un segment. Soit $s < t < v$, alors $[s : t] \cup [t + 1 : v] = [s : v]$. L'arbre T de taille $n$ qui est une chaîne peut aussi être noté $[1 : n]$. Enfin, dans le cas des arbres, les ruptures sont des arêtes. Dans le cas des chaînes, comme il est équivalent de considérer une arête et l'un des deux noeuds reliés par l'arêtes, il est fréquent dans la littérature sur la détection de ruptures sur les arbres de placer les ruptures au niveau des noeuds plutôt que sur les arêtes.

Il a déjà été statué qu'implémenter un algorithme qui résoudrait (1.27) ou (1.28) en testant toutes les partitions possibles aurait une complexité algorithmique exponentielle. Il est parfois possible, grâce à la théorie du domaine d'où sont issues les données, de réduire l'ensemble des partitions possibles à un espace plus petit. Cependant, même quand cela reste rare et n'est en général pas suffisant pour réduire la complexité algorithmique à une complexité polynomiale. Le problème de la génération de toutes les partitions possibles trouve sa solution dans la Programmation Dynamique. Dans un article datant de 1954, Bellman [10] introduit les équations du même nom, qui permettent de réduire le coût des procédures d'optimisation grâce à une récurrence. La contribution de Bellman est à la source de la création d'algorithmes de programmation dynamique.

Dans le cas de la minimisation du coût constraint (1.27), cela permet de calculer la partition optimale de T en $K$ segments notée $\mathbf{P}^*_K$ en un temps quadratique comme c'est le cas dans l'algorithme

de Auger et Lawrence [5]. Le principe général dans notre cas repose sur le fait que la partition optimale $\mathbf{P}_K^*$ de taille $K$ du graphe chaîne $[1:n]$ est l'union de la partition optimale du segment $[1:\tau]$ pour un certain $\tau < n$ en $K-1$ segments et du segment $[\tau+1:n]$. Il est donc possible de trouver de façon itérative la partition optimale de $[1:t]$ en $k$ segments, dont le coût est noté $\mathrm{C}_{[1:t]}^k$, sachant le coût de la partition optimale de $[1:s]$ pour $s$ plus petit que $t$ en $k-1$ segments.

$$\mathrm{C}_{[1:t]}^k = \min_{s<t} \mathrm{C}_{[1:s]}^{k-1} + \mathrm{C}_{[s+1:t]} \tag{1.29}$$

où $\mathrm{C}_{[s+1:t]}$ est le coût du segment $[s+1:t]$.

Auger et Lawrence se sont basés sur les travaux de Bellman dans [5] pour implémenter un algorithme appelé Segment Neighborhood, abrégé en SN, qui trouve la partition optimale d'une chaîne de taille $n$ en $K$ segments, pour tout $K$ entre 1 et $n$, avec une complexité de $O(Kn^2)$ opération. Cette complexité est atteinte si le temps de calcul de chaque segment de $[1:n]$ est une constante. Dans l'algorithme SN, il faut produire les partitions optimales de la chaîne en $K$ segments pour $K$ variant entre 1 et un certain entier $K_{\max}$ puis sélectionner la meilleure ce qui constitue à la fois un avantage et un inconvénient. S'il peut être pratique d'avoir accès à des partitions de $[1:n]$ de différentes tailles, il est également coûteux de devoir construire plusieurs partitions.

Plus récemment, Jackson et al. [55] ont développé un algorithme appelé Optimal Partitioning, ou OP, capable de retrouver la partition optimale de $[1:n]$ en résolvant la minimisation de coût pénalisé (1.28). Dans ce cas, l'équation de récurrence utilisée dans l'algorithme pour construire la partition optimale $\mathbf{P}_{[1:t]}^*$ de $[1:t]$ s'écrit :

$$\mathbf{P}_{[1:t]}^* = \min_{s<t} \mathbf{P}_{[1:s]}^* + \mathrm{C}_{[s+1:t]} + \beta. \tag{1.30}$$

Dans l'article de Jackson et al [55], les auteurs utilisent une pénalisation linéaire où la taille de la partition donc, de façon équivalente, soit le nombre de ruptures soit le nombre de segments sont pénalisés.

$$\mathrm{pen}(\mathbf{P}) = \beta\,(K_{\mathbf{P}} - 1) \tag{1.31}$$

où $\beta$ est un paramètre de pénalité à choisir judicieusement. Avec cette forme de pénalité, chaque rupture d'une partition ajoute une constante $\beta$ au coût de la partition. Si le paramètre $\beta$ est bien choisi, la partition obtenue est la vraie partition de la chaîne. Un $\beta$ trop petit ou trop grand donnera une partition trop grande, respectivement trop petite en termes de nombre de segments.
Si la pénalité est linéaire, elle est peut être directement incluse dans le coût de chaque sous-arbres d'une partition. Le choix de la pénalité et dans ce cas du paramètre $\beta$ est crucial pour permettre à l'algorithme de retrouver la vraie partition de $[1\ldots,n]$.

En général, il existe un large choix de pénalité pour la fonction $\mathrm{pen}(\mathbf{P})$ dans (1.28).
Supposons que l'on se place dans le modèle gaussien univarié avec une variance constante $\sigma^2$ connue. Il est montré dans Lebarbier [71] que si l'on pénalise les moindres carrés par une pénalité concave de la forme $\mathrm{pen}(\mathbf{P}) = \sigma^2\,[c_1|\mathbf{P}| + c_2|\mathbf{P}|\log(n/|\mathbf{P}|)]$ avec des constantes $c_1$ et $c_2$ judicieusement choisies, il est possible d'estimer le vecteur des moyennes des observations à une distance presque optimale et donc obtenir un bon estimateur de la vraie partition des observations. Les travaux de Lebarbier sont eux même inspirés de la sélection de modèle de Massart [87]. La pénalité introduite dans ce cas est concave, il n'est donc pas possible de l'utiliser dans l'algorithme OP de Jackson et al. qui requiert une pénalité linéaire. Pour pallier ceci, il est possible d'utiliser une pénalité linéaire dans l'algorithme qui correspond à la pénalité concave de Lebarbier. Pour cela, il faut montrer le paramètre $\beta$ de la pénalité linéaire est lié à la dérivée de la pénalité concave et il est possible de mettre au point une stratégie de mise à jour du paramètre $\beta$. Après avoir choisi une valeur initiale pour $\beta$, l'algorithme OP, ou un autre algorithme similaire, est itéré. Supposons que la partition construite est de taille $k$. Le $\beta$ est mis à jour grâce à la dérivée de la pénalité concave en $k$ et on recommence jusqu'à

convergence. Cette méthode proche d'une descente de gradient est entièrement inspirée de l'article de Killick et al. [64].

Si la variance n'est pas connue, comme avec certains jeux de données réelles, la sélection de modèle pour un modèle gaussien devient plus compliquée. Quand la variance est connue, moins la log-vraisemblance de l'échantillon est égal au coût des moindres carrés. Si ce n'est pas le cas, l'expression de la log-vraisemblance est beaucoup moins agréable à appréhender et cela génère des difficultés à la fois pour minimiser le coût pénalisé et pour trouver la pénalité qui permet d'obtenir la partition optimale. Afin de pallier ce problème annexe, l'article de Baraud, Giraud et Huet [8] propose des pénalités adaptées pour différentes applications de la sélection de modèle sur des observations gaussiennes de variance inconnue.

Pour revenir au choix de la pénalité dans OP, les cas où la variance $\sigma^2$ n'est pas connue ou d'une distribution de familles exponentielles sont abordés dans [71] et [22]. Des résultats similaires sont montrés dans [4, 40] pour le kernel change-points algorithm.

Dans les algorithmes SN et OP, un coût est minimisé sur la dernière rupture de la partition de $[1:t]$ pour $t$ de 1 à $n$. L'algorithme donne naturellement en sortie la meilleure partition de la chaîne $[1:n]$. La complexité de ces algorithmes est de l'ordre de $Kn^2$ ou $n^2$ ce qui est une énorme amélioration par rapport à la complexité exponentielle de la méthode naïve mais qui reste élevée surtout lorsque l'on a affaire à de gros jeux de données. Pour des gros jeux de données, seulement les algorithmes capables de détecter les ruptures en un temps quasi-linéaire son envisageables. Heureusement, de nombreux articles traitent le sujet en proposant des méthodes d'élagages afin de réduire considérablement la complexité des algorithmes existants. L'élagage repose sur un principe simple, à l'étape $t$, plutôt que de minimiser le coût ou le coût pénalisé sur un ensemble de taille $t-1$, qui correspond à tous les noeuds plus petits que $t$, on va minimiser sur un ensemble plus petit, en se débarrassant de certains noeuds qui ne peuvent pas être des ruptures dans la partition optimale. Pour déterminer les noeuds à retirer de l'ensemble que l'on cherche à minimiser, l'algorithme vérifie qu'ils respectent certains critères.

Ce sont Killick et al. [64] qui proposent un premier article sur l'élagage pour l'algorithme OP dans [64], créant l'algorithme PELT. L'idée générale de PELT est de se débarrasser des partitions dont on sait qu'elles sont sous optimales dans les étapes de minimisation du coût. Si une partition de $[1:t]$ telle qu'il existe une rupture en $s$ possède un coût plus grand qu'un certain seuil, alors $s$ ne peut pas être une rupture dans la partition optimale de n'importe quel segment de $[1:t']$ avec $t'$ plus grand que $t$. Il n'est donc pas nécessaire de considérer $s$ comme une rupture dans les futures étapes de minimisation de l'algorithme OP. L'algorithme PELT permet de trouver la partition optimale des observations sans contrainte sur la taille de celle-ci. Dans le pire des cas, la complexité reste quadratique car l'élagage n'est pas possible, mais dans certains cas particuliers, comme quand il y a beaucoup de ruptures dans la vraie partition, il est possible d'atteindre une complexité linéaire.
L'élagage de PELT se formalise de la façon suivante.
Le coût doit respecter une contrainte supplémentaire. Soit $s < \tau < t$, alors il existe une constante $\kappa$ telle que :

$$\mathrm{C}_{[s:\tau]} + \mathrm{C}_{[\tau:t]} + \kappa \leq \mathrm{C}_{[s:t]}\,. \tag{1.32}$$

Dans ce cas, l'algorithme d'élagage de Killick et al. [64] assure que si pour un certain $\tau$ plus petit que $t$ l'inégalité :

$$\mathrm{Cost}_{[1:\tau]}(\mathbf{P}^*_{[1:\tau]}) + \mathrm{C}_{[\tau+1:t]} + \kappa > \mathrm{Cost}_{[1:t]}(\mathbf{P}^*_{[1:t]}), \tag{1.33}$$

est vérifiée, cela implique que $\tau$ ne peut pas minimiser l'équation (1.30) pour tout $s$ plus grand que $t$. En d'autres mots, $\tau$ ne peut pas être une rupture dans la partition optimale de tout segment $[1:s]$ pour $s$ plus grand que $t$ et à fortiori ne peut pas être une rupture dans la partition optimale de $[1:n]$. Il devient alors possible de ne plus considérer $\tau$ dans l'ensemble des dernières ruptures

sur lequel on minimise dans (1.30).

Au prix de conditions plus fortes sur la fonction de coût, il existe un autre type d'élagage qui s'appelle l'élagage fonctionnel. Celui-ci est introduit dans un article de Rigaill [103] qui applique cette méthode d'élagage à l'algorithme SN. Plus tard, l'article de Maidstone et al. [83] généralise l'élagage fonctionnel à l'algorithme OP et montre que l'élagage fonctionnel est toujours plus efficace que l'élagage PELT.

Notre objectif est d'adapter l'algorithme PELT au cas des arbres afin de pouvoir l'utiliser sur des données d'abondances. En plus de l'étude de la synchronie spatiale, l'algorithme implémenté est utile dans d'autres contextes déjà évoqués en début de sous-section. En adaptant naïvement l'algorithme OP, sans ajouter l'élagage PELT, la complexité de l'algorithme sera exponentielle. En effet, si le nombre de partitions sur un arbre ou sur une chaîne sont égaux, ce n'est pas le cas pour le nombre de derniers segments. Lorsque la relation de récurrence qui nous permet de construire toutes les partitions grâce à la Programmation Dynamique est utilisée, le coût est minimisé sur les dernières ruptures possibles dans la partition, ce qui signifie le dernier segment de la partition. Dans le cas des arbres, nous voulons minimiser sur l'ensemble des derniers sous-arbres qui remplacent les derniers segments du cas chaîne. Si le nombre de derniers segments de la partition d'une chaîne est au plus $n$, le nombre de sous-arbres de la partition d'un arbre est exponentiel en la taille de $n$, même pour des arbres binaires. Plus généralement, tout algorithme de détection de ruptures sur les chaînes par minimisation de coût que l'on adapte au cas des arbres sans méthode d'élagage souffrira d'une complexité algorithmique exponentielle.

Par conséquent, un algorithme de détection de ruptures sur les arbres ne peut se passer d'une phase d'élagage, d'où l'intérêt de vérifier que l'élagage PELT est possible dans notre modèle. En plus de l'adaptation de la relation de récurrence nécessaire à la Programmation Dynamique et de l'élagage PELT, nous voulons pouvoir faire de la sélection de modèle pour des données multivariées. Ainsi, un de nos objectifs est de trouver une pénalité permettant d'obtenir la partition optimale des données dans le cas où celles-ci sont multivariées.

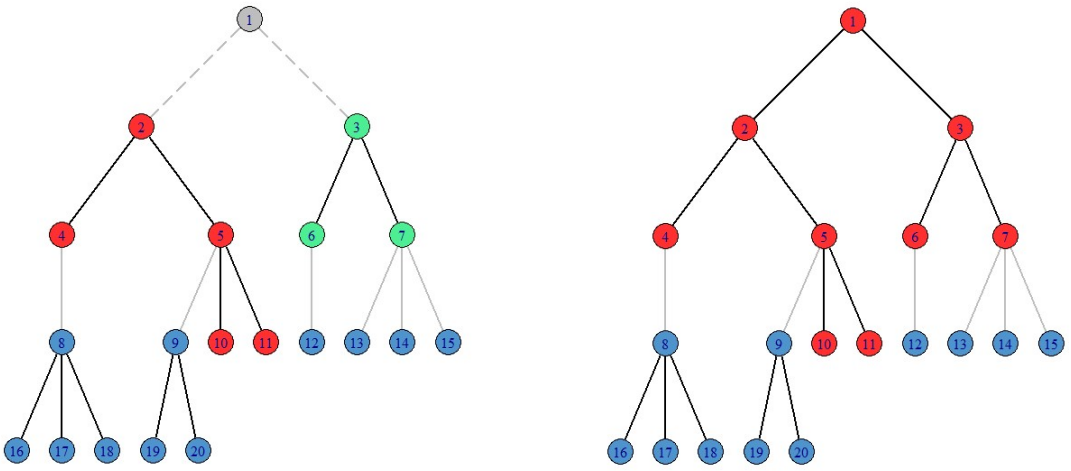
L'étude de la synchronie spatiale, en dehors de la structure d'arbre que l'on a choisi, est l'objet de plusieurs articles. Dans [43], les auteurs proposent une méthode statistique pour identifier des régions géographiques possédant des dynamiques de population synchrones à partir de données d'abondance basé sur la minimisation de la log-vraisemblance pénalisé des observations. Si certains aspects de leur procédure sont proches de la notre, leurs travaux n'exploitent pas la structure spécifique des arbres.

En sortant du contexte des procédures de minimisation de coût, et dans le cas de graphes quelconques, il est possible d'étendre l'estimateur obtenu grâce au Fused Lasso en utilisant la pénalité $\sum_{\{i,j\}\in\mathcal{E}} |\theta_i - \theta_j|$. Cette méthode est parfois appelée en anglais total variation denoising [53] ou trend filtering sur les graphes [131]. Le critère choisi est convexe, il est donc possible de l'optimiser. Dans le cas très spécifique de la détection de ruptures dans la moyenne (1.23) d'observations univariées, Fan et Guan [34] proposent une méthode d'approximation pour minimiser les moindres carrés pénalisés par le nombre de ruptures. Cependant, il est très compliqué d'étendre cette méthode à un modèle plus général. Le cas des graphes quelconques est un sujet beaucoup plus délicat que la segmentation sur les arbres abordée dans ce chapitre. C'est pour cela que nous nous concentrons sur le projet plus réaliste d'implémenter un algorithme capable de retrouver la partition optimale des observations réparties sur un arbre T, plutôt que de considérer le cadre plus général des graphes quelconques.

### 1.3.3 Notre contribution - résultats

Notre contribution se divise en plusieurs parties.
Dans un premier temps, nous nous intéressons à l'aspect algorithmique des procédures de minimisation du coût sur les arbres. S'il existe une littérature très riche sur le sujet dans le cas de la détection de rupture sur les chaînes, nous n'avons pas connaissance de résultats sur les arbres. Il n'est malheureusement pas envisageable d'étendre les algorithmes de Programmation Dynamique [10] au cas des arbres à cause de la différence de structure entre les chaînes et les arbres. Si des algorithmes comme ceux de Auger et Lawrence[5] ou de Jackson et al. [55] atteignent une complexité quadratique, leur adaptation aux arbres aurait un coût exponentiel. Ainsi, Réalisant qu'une stratégie d'élagage sera nécessaire pour implémenter un algorithme de segmentation sur les arbres, nous nous sommes inspirés des travaux de Killick [64] et de son algorithme PELT afin de diminuer la complexité algorithmique.

Nous avons formalisé toutes les notations des chaînes au cas des arbres. La récurrence de Programmation Dynamique (1.28) doit être repensée pour le cas des arbres. Nous enracinons dans un premier temps l'arbre T. Toutes les partitions de sous-arbres engendrés par des noeuds de T sont définies par leur dernier sous-arbre. Plus précisément, soit $\mathrm{T}_s$ le sous-arbre engendré par $s \in \mathcal{V}(\mathrm{T})$, alors la racine de $\mathrm{T}_s$ est $s$. Une partition $\mathbf{P}$ de $\mathrm{T}_s$ est définie par le sous-arbre de racine $s$. Dans le cas des chaînes, une partition était définie par son dernier segment, ou sa dernière rupture de manière équivalente.
L'algorithme que nous proposons est capable de construire tous les derniers sous-arbres de $\mathrm{T}_s$, et donc toutes les partitions possible de $\mathrm{T}_s$, à partir des derniers sous-arbres des $\mathrm{T}_t$ pour $t$ enfant de $s$. La figure 1.8 illustre notre procédure.




(a) (b)

Figure 1.8 – Cette figure illustre comment nous construisons le dernier sous-arbre d'une partition de $T_1$ à partir des derniers sous-arbres de partition de $T_2$ et $T_3$. Sur la figure (a), nous considérons en rouge le dernier sous-arbre d'une partition $\mathbf{P}^{(2)}$ de $T_2$ et en vert le dernier sous-arbre d'une partition $\mathbf{P}^{(3)}$ de $T_3$. Le noeud 1 est représenté en gris, nous ne le prenons pas en compte dans cette figure, c'est pour cela que les arêtes entre 1 et ses enfants sont pointillées. Les autres noeuds sont représentés en bleu, ils font partie d'autres sous-arbres des partitions $\mathbf{P}^{(2)}$ et $\mathbf{P}^{(3)}$. La figure (b) représente en rouge le dernier sous-arbre de la partition $\mathbf{P}^{(1)}$ créée à partir des partitions $\mathbf{P}^{(2)}$ et $\mathbf{P}^{(3)}$. Le sous-arbre rouge de la figure (b) est définie par l'union du noeud 1 et des noeuds rouge et vert de la figure (a) ainsi que des arêtes qui relient ces noeuds. Dans les deux figures, les arêtes pleines et grises représentent les ruptures des différentes partitions tandis que les arêtes pleines et noires représentent les arêtes qui relient deux noeuds d'un même sous-arbre.

Ensuite, nous avons réécrit le théorème d'élagage de PELT au cas des arbres. Le principe de l'élagage PELT dans les arbres est similaires au cas des chaînes. La différence réside dans le fait que les derniers segments, ou dernières ruptures, sont remplacés par les derniers sous-arbres. Afin d'élaguer certaines partitions, le théorème d'élagage de Killick et al. [64] s'adapte aux arbres de la façon suivante. Soit $T_s$ le sous-arbre de T engendré par $s$, si une partition $\mathbf{P}(T_s)$ est telle que son coût est supérieur au coût de la partition optimale de $T_s$ plus le paramètre de pénalisation $\beta$, alors le dernier sous-arbre de $\mathbf{P}(T_s)$ ne peut pas être un sous-arbre de la partition optimale de T. Ce sous-arbre ne sera plus utilisé afin de construire des partitions des noeuds dont $s$ est le descendant. Nous pouvons alors enlever ce dernier sous-arbre de l'ensemble sur lequel le coût pénalisé est minimisé à chaque future étape de l'algorithme. Dans la figure 1.8, nous avons représenté la construction des derniers sous-arbres des partitions des sous-arbres de T engendrés par les noeuds $s \in \mathcal{V}(T)$. Pour illustrer l'élagage PELT adapté au cas des arbres, supposons que le sous-arbre rouge, que l'on note $S$, dans la figure 1.8a engendre une partition non optimale de $T_2$. Alors nous prétendons que le sous-arbre rouge de la figure 1.8b ne peut pas être associé à une partition optimale de $T_1$. Il est donc possible de retirer le sous-arbre $S$ lorsque nous construisons les derniers sous-arbres de $T_1$ à partir de ceux de $T_2$ et $T_3$. La procédure est expliquée plus en détails et de façon plus formelle dans le chapitre sur la segmentation d'arbres.

Dans un second temps, nous nous plaçons dans le modèle Gaussien multivarié et dans le cas de la détection de ruptures dans la moyenne des distributions (1.23) sur des arbres. Nous nous basons sur les travaux de Lebarbier [71] pour le modèle univarié afin d'introduire une pénalité concave spécialement adaptée à la fonction de coût des moindres carrés pénalisés et au modèle étudié, et telle que l'estimateur $\widehat{\mathbf{P}}$ associé atteigne des performances statistiques proches de l'optimum.

Notre algorithme est utilisé sur des données de pêche donnant les séries d'abondances de différentes espèces de poissons sur le réseau de rivières du Bassin de la Loire en France.

Nous avons implémenté un algorithme de détection de ruptures sur les arbres avec Rcpp. Ce dernier peut être utilisé avec deux fonctions de coût différentes : les moindres carrés, et la log-vraisemblance pour un échantillon i.i.d. issus d'un modèle de Poisson multivariée. Le code est disponible à l'url suivante `https://github.com/Sosotitili/Tree-Segmentation.git`.

## 1.4 Synchronie interspécifique

### 1.4.1 Motivations

Le troisième chapitre de ma thèse concerne des travaux exploratoires sur des données d'abondances de papillons. Dans ce chapitre, nous développons une méthodologie afin de modéliser des données issues des sciences participatives et d'analyser la synchronie inter-espèces au sein de la communauté formée par les observations. Le cadre est très proche de ce qui a été présenté au tout début de cette introduction, il est donc possible que certaines informations se répètent, mais elles sont nécessaires à la bonne compréhension des travaux menés dans ce chapitre.

J'ai déjà évoqué au début de l'introduction la popularité de l'étude de la stabilité des communautés en écologie. La stabilité des communauté est très liée aux notions de diversité et de synchronie. Beaucoup de travaux s'intéressent aux liens entre la stabilité et la synchronie, particulièrement pour des communautés de plantes. Cependant, la relation diversité-synchronie a été beaucoup moins étudiée dans les communautés animales. Cela peut être en partie dû à la difficulté d'avoir des données d'abondances exploitables lorsque l'on s'intéresse à des espèces mobiles qui se cachent des êtres humains. Cependant, au vu du contexte actuel de changements globaux et de la grande contribution des espèces animales dans les fonctions des écosystèmes (la pollinisation, la dispersion des graines, ou la propagation de maladies, etc.), il est très important de comprendre les éléments moteurs de la stabilité d'un écosystème et l'impact de la dégradation de l'habitat ou des changements climatiques sur les communautés.

Dans ce chapitre, nous nous intéressons plus précisément à la synchronie au sein des fluctuations temporelles des abondances d'espèces animales. Les tendances temporelles à long terme intéressent particulièrement les écologues comme en témoignent les nombreux programmes de surveillance dédiés à diverses espèces [20, 82]. Elles permettent d'étudier et de prévoir le déclin ou, au contraire, la prospération des communautés et d'agir en conséquence. Nous supposons que si les espèces qui partagent la même tendance temporelle à long terme sont dans le même groupe de synchronie, alors il est attendu que certains groupes de synchronie s'éteignent, impactant significativement les dynamiques de population et la stabilité au sein de l'écosystème. Les raisons des fluctuations des séries temporelles d'abondances sont fortement liées aux changements de l'environnement et à la compétition entre les espèces [50, 57]. Il est possible que la sensibilité plus ou moins forte de chaque espèce aux différents facteurs environnementaux s'explique en grande partie par les traits qui leurs sont propres. Dans ce cas, les espèces d'un même groupe de synchronie partageraient des traits similaires. Une fois les groupes de synchronie établis, il est intéressant de voir si les traits liés aux fonctions de l'écosystème sont présents dans différents groupes de synchronie mais aussi si certains traits peuvent expliquer la synchronie inter-espèces.

Nos travaux sont alors motivés par deux questions :

- Existe-t-il un lien entre la synchronie au sein d'une communauté et les tendances temporelles à long terme ?
- Existe-t-il un lien entre la synchronie au sein d'une communauté et les traits des espèces ?

Nous nous sommes penchés sur ces deux questions à une échelle régionale, car la synchronie régionale est supposée être l'un des principaux moteurs de la stabilité régionale. Cette échelle correspond également à l'échelle la plus adaptée en écologie lorsque l'on s'intéresse aux stratégies de gestion et de conservation des écosystèmes.

### 1.4.2 Données et Modèle

Nous répondons à ces questions en analysant des données d'abondances de papillons réunies dans un jeu de données couvrant le territoire de la Grande-Bretagne. Le jeu de données consiste en des séries temporelles d'abondances entre 2006 et 2015, avec des observations journalières entre Mars et Octobre sur plusieurs centaines de sites. Il est extrait du programme de sciences participatives UK Butterfly Monitoring Scheme dont la procédure peut être trouvée à l'url (UKBMS, http://www.ukbms.org/).
C'est la tendance temporelle inter-annuelle qui nous intéresse, et non pas les variations intra-annuelles des espèces. Pour étudier les variations inter-annuelles au sein de la communauté et construire des groupes de synchronie basés sur les séries temporelles d'abondances, nous voulons avoir accès à la série temporelle des abondances relatives annuelles des espèces. L'abondance relative annuelle d'une espèce représente la proportion d'une espèce par rapport aux abondances des autres espèces de la communauté.

Malheureusement, la procédure de collecte des données engendre un manque de régularité dans les observations. En effet, les espèces ne sont pas forcément observées tous les jours de l'année et les jours de collecte ne sont pas forcément les mêmes selon les sites. A cause de cela, il n'est pas possible d'extraire directement les abondances relatives inter-annuelles. Afin de pallier ce problème, nous prenons en compte les phénologies des espèces qui représentent les événements périodiques qui régissent la vie d'un animal ou d'une plante. La phénologie est propre à chaque espèce et les phénologies des papillons sont liées aux séries temporelles des éclosions des cocons au cours de l'année.
Pour estimer les phénologies et l'abondance relative annuelle, nous nous basons sur les travaux de Schmucki et al. [109]. Leur approche est la suivante. Ils estiment dans un premier temps la forme des phénologies annuelles pour chaque espèce en utilisant les abondances de l'espèce agrégées sur tous les sites de la région. Puis la forme de la phénologie est utilisée pour prédire les données manquantes et corriger certaines observations pour chaque espèce. L'aire sous la courbe de la phénologie donne un indice d'abondances annuelles de l'espèce considérée. Lorsque l'on utilise cette approche, il est considéré que la phénologie annuelle d'une espèce est constante sur tous les sites de la région. Cependant, il est connu que les conditions météorologiques ont un impact considérable sur la phénologie [29, 97, 106, 112, 113, 126]. Nous devons donc partitionner les sites en régions bioclimatiques issues du UK Météorology Office et estimer une phénologie par espèce, région et année.

Le modèle statistique proposé par [109] pour les données d'abondances est un modèle linéaire généralisé semi-paramétrique. Soit $N_{syd}^{(i)}$ l'abondance de l'espèce $i$, sur le site $s$, le jour $d$ de l'année $y$, alors :

$$N_{syd}^{(i)} \sim \mathcal{P}\left(\mathrm{e}^{\left(\gamma_{sy}^{(i)} + f^{(i)}(d)\right)}\right), \tag{1.34}$$

où la fonction $f^{(i)}(d)$ représente la phénologie. La phénologie respecte la standardisation

$$\sum_{d} \exp(f^{(i)}(d)) = 1$$

pour chaque espèce $i$. Le paramètre $\gamma_{sy}^{(i)}$ se décompose comme la somme d'un effet site $\alpha_s^{(i)}$ ne dépendant pas du temps et $\beta_{sy}^{(i)}$ qui représente les variations inter-annuelles des abondances. C'est ce dernier paramètre qui permet de construire les groupes de synchronie. Le modèle est ajusté en maximant la vraisemblance des données. Une base de spline cubique est utilisée pour la phénologie.

Dans notre cas, le modèle est légèrement modifié en remplaçant la loi de Poisson par une Binomiale Négative car il a été montré dans la thèse de T. Olivier [95] que la variance des observations était très forte et que l'on obtient de bien meilleures adéquations avec une Binomiale Négative.

Une fois les paramètres $\gamma_{sy}^{(i)}$ estimés, le paramètre $\beta_{sy}^{(i)} = \gamma_{sy}^{(i)} - \alpha_s^{(i)}$ est récupéré et nous formons une série temporelle des $\beta$ par espèce et par site. Comme c'est la synchronie régionale qui nous intéresse, nous avons besoin de créer à partir des séries temporelles inter-annuelles des espèces, une seule série temporelle par espèce. Pour cela, nous agrégeons les variations d'abondance inter-annuelles $\beta_{sy}^{(i)}$ en une variation d'abondance inter-annuelle régionale en prenant la médiane sur tous les sites de la région.

$$\beta_y^{(i)} = \text{median}\{\beta_{sy}^{(i)} : s \in \text{UK}\}. \tag{1.35}$$

Ici, régional correspond à l'ensemble du territoire britannique et non pas aux régions bioclimatiques évoquées précédemment.

D'après [36], il est possible que les données soient sujettes à une tendance temporelle sur le long terme. Ce qui se traduit par le fait que les séries temporelles sont en déclin ou en expansion. Si la tendance est plus grande que les variations des abondances, les coefficients de synchronie que l'on va calculer entre les espèces risquent de ne refléter que la tendance, et non pas les variations qui nous intéressent. Pour éviter ce phénomène, nous calculons la tendance des séries temporelles des $\beta_y^{(i)}$ grâce à une régression linéaire et les $\beta_y^{(i)}$ sont corrigés en retranchant la tendance. Il reste alors les séries temporelles des résidus $\tilde{\beta}_y^{(i)}$ que nous allons analyser dans la suite de nos travaux.

Dans les deux prochaines sous-sections, je vais présenter succinctement les méthodes et les outils statistiques utilisés pour apporter des réponses aux deux problématiques de ce chapitre.

#### Etude empirique des groupes d'espèces synchrones

Notre but est de mettre en lumière des groupes d'espèces synchrones au sein de la communauté des papillons en Grande-Bretagne, puis d'étudier la répartition des tendances temporelles parmi les groupes de synchronie. Si les tendances temporelles sur le long terme sont réparties équitablement, cela impliquerait une certaine stabilité de l'écosystème, alors que la concentration des tendances temporelles dans des groupes distincts peut compromettre cette stabilité.

Dans un premier temps, nous calculons, à partir des séries temporelles des résidus calculés précédemment, des indices de synchronie entre les espèces. Pour ce faire, nous utilisons les coefficients de Spearman introduits au début de ce document. Ces derniers nous permettent de prendre en compte uniquement les variations des séries temporelles et non pas leurs amplitudes. Nous obtenons alors une matrice de corrélation dont les lignes et colonnes représentent les espèces et les coefficients $\rho_{i,j}^S$ sont les coefficients de Spearman entre l'espèce $i$ et l'espèce $j$.

Parmi les algorithmes de clustering cités précédemment, le clustering hiérarchique semble prometteur. Le livre de Kauffman [62] introduit parfaitement le clustering hiérarchique ainsi que d'autres méthodes de clustering. La création d'un dendogramme permet de visualiser les distances entre les espèces et les groupes d'espèces et ainsi d'ajuster le choix du nombre des groupes à l'analyse des écologues. La matrice de dissimilarité, ou distance, utilisée est 1 moins la matrice des corrélations de Spearman. Ses coefficients sont notés $d_{i,j}$ avec $d_{i,j} = 1 - \rho_{i,j}^S$.
Il aurait été possible d'utiliser une distance euclidienne entre les observations, mais la mesure de synchronie définit déjà une distance entre les observations. Il est alors plus judicieux d'utiliser la matrice des corrélations comme matrice de distance.

**Etablir des liens entre groupes synchrones et traits des espèces**

Une fois les groupes d'espèces synchrones estimés grâce aux travaux précédents, nous cherchons à établir un lien entre les traits des espèces et les groupes de synchronie.

Nous avons accès à un jeu de données contenant les traits des espèces et aux labels des observations qui correspondent aux groupes de synchronie. L'analyse que nous menons dans cette partie se découpe en deux axes.

- A quel point est-il possible de prédire les groupes de synchronie à partir des traits ?
- Parmi tous les traits, quels sont ceux qui ont la plus grande importance pour prédire les groupes ?

Afin de répondre à la première question, nous allons utiliser des algorithmes d'apprentissage supervisés. Dans les algorithmes d'apprentissage supervisé comme les arbres de décision, Support Vector Machine (SVM), Random Forest (RF), modèles de régressions, etc. [16, 68] les observations du jeu de données possèdent toutes un label qui est la variable à prédire à partir du reste des variables dites explicatives. L'objectif est de construire, à partir du jeu de données, un modèle capable de prédire le label de nouvelles observations. La procédure classique consiste à séparer le jeu de données en un jeu de données d'apprentissage, sur lequel on entraîne le modèle grâce aux labels, et un jeu de données de test, qui permet de tester le modèle sur de nouvelles observations et d'évaluer ses performances. Pour entraîner le modèle, un algorithme d'apprentissage supervisé va chercher à optimiser les paramètres d'un modèle de façon à minimiser une fonction de perte qui dépend du vrai label et du label prédit par le modèle.

Parmi toutes les méthodes de classification qui existent et dont on peut trouver les détails dans [68], nous avons sélectionné deux algorithmes. Le premier est l'algorithme Random Forest [16], basé sur les arbres de décision. Un des avantages de cet algorithme est qu'il est adapté à tous types de jeux de données. Notre jeu de données de traits comporte aussi bien des variables numériques que des variables catégorielles. Cela n'est pas un problème lorsque l'on utilise Random Forest car la méthode repose sur la construction d'arbres de décision dans lesquels chaque noeud comporte un test sur l'une des variables du jeu de données. L'algorithme construit, à partir d'un jeu de données d'entraînement, un certain nombre d'arbres de décision dont les feuilles donnent le groupe estimé de l'observation. La classification d'une nouvelle observation est votée à la majorité des arbres de décision.

La deuxième méthode de classification a été choisie pour son efficacité. Les réseaux de neurones artificiels s'inspirent du fonctionnement et de la structure du cortex cérébral humain. L'utilisation massive des réseaux de neurones dans des problèmes d'apprentissage supervisé est relativement récente alors que l'idée à l'origine des algorithmes connus aujourd'hui remonte aux années 1950 dans un article de biologie [74]. Une description plus complète du fonctionnement des réseaux de neurones est donnée dans les annexes du chapitre trois de ma thèse. Le livre [45] est une bonne introduction aux réseaux de neurones et au deep learning. J'explique ici brièvement ses mécanismes. L'idée est de construire un réseau qui se compose de plusieurs couches de neurones. Le signal va parcourir le réseau, neurone par neurone. A chaque neurone sont associés un poids, un biais et une fonction dite d'activation qui permettent de transformer le signal selon les objectifs du modèle. En entrée, on donne les valeurs des variable explicatives d'une observation. En classification, les neurones de sortie peuvent donner le groupe auquel l'observation est assignée ou une probabilité d'appartenance à chaque groupe. Grâce à une méthode d'optimisation, l'algorithme utilise un jeu de données d'entraînement pour choisir les poids et les biais qui permettent de prévoir au mieux le groupe de l'observation. Il est ensuite capable de prédire le label de nouvelles observations.

Les réseaux de neurones sont plus capricieux que l'algorithme Random Forest et un certain nombre de modifications doivent être faites sur les données avant de pouvoir les utiliser. Par exemple, les données doivent obligatoirement être renormalisées, ce qui n'est pas le cas avec les Random Forest, et un réseau de neurones artificiel ne prend pas de variables catégorielles en entrée. Même si les réseaux de neurones sont utilisés dans un très grand nombre de domaines comme la reconstruction d'image [136], la prise de décision en finance [120], etc. il est rare de les voir appliqués à des données d'abondances en écologie. Pour cette raison, nous avons utilisé des réseaux simples et de petites tailles afin d'avoir un premier aperçu des résultats qu'il est possible d'obtenir sur ce type de jeu de données.

Dans un second temps, nous nous intéressons aux variables les plus importantes pour prédire les groupes de synchronie. Pour cela, nous utilisons des fonctions R déjà implémentées provenant des package `caret` [38] et `NeuralNetTools` [9]. Pour les Random Forest, nous utilisons les méthodes Decreasing Gini ou Decreasing Accuracy. La première méthode consiste à calculer l'importance des variables en se basant sur l'indice Gini, un indice calculé pendant la construction des arbres de décision qui permet de choisir quelle variable est sélectionnée comme variable test à chaque noeud de l'arbre. Nous regardons alors à quel point la somme des indices Gini de l'ensemble de l'arbre décroit si l'on enlève une des variables du jeu de données. Plus la valeur de la somme décroît, plus la variable en question a un impact sur la décision finale du modèle. Dans le cas de Decreasing Accuracy, le principe est le même que Decreasing Gini mais en se basant sur la précision du modèle. Dans le cas des Réseaux de Neurones, nous privilégions la méthode Olden [94] qui utilise des ratios des poids du réseaux pour attribuer à chaque variable une valeur d'importance.

Les valeurs calculées reflètent à quel point les différentes variables ont un impact sur le modèle destiné à prédire le groupe de synchronie des espèces. Souvent, un petit groupe de variables se détache du reste avec une grande valeur d'importance. Parfois, les importances décroissent lentement entre les différentes variables. Dans le dernier cas, il faut s'interroger sur la fiabilité des résultats. Dans un jeu de données comme le nôtre, comprenant un grand nombre de variables explicatives, nous aimerions réussir à isoler les variables qui sont moteurs de la synchronie.

### 1.4.3 Notre contribution - résultats

Malgré l'efficacité reconnue des méthodes utilisées, nos différents jeux de données sont typiques des données en écologie. Les observations sont très bruitées, avec un nombre conséquent de données manquantes et souvent beaucoup de variables pour peu d'observations. Même s'il est possible d'obtenir des résultats, il peut être difficile d'adapter les méthodes existantes sans un effort supplémentaire de pré-traitement des données. La particularité des travaux présentés dans ce chapitre réside à la fois dans les motivations écologiques sur les variations temporelles des abondances d'animaux mais aussi dans la façon d'appliquer des méthodes classiques de clustering et d'apprentissage supervisé à des jeux de données particuliers.

Dans le cas de l'analyse du lien entre la synchronie et les tendances temporelles sur le long terme, nos travaux permettent de mettre en lumière la synchronie inter-spécifique régionale des variations des séries temporelles d'abondances parmi les espèces de papillons présentes en Grande-Bretagne. Les écologues du Museum d'Histoire Naturelle en déduisent l'existence possible d'une dynamique de compensation servant à stabiliser les abondances des communautés régionales de papillons. De plus, après analyse des tendances temporelles à long terme, nous n'avons pas trouvé de distribution particulière des tendances autre que la distribution aléatoire attendue parmi les groupes de synchronie. D'après les écologues, cela suggère que la synchronie et les tendances temporelles sur le long terme sont induites par des facteurs environnementaux différents.

Dans le cas de l'étude du lien entre la synchronie et les traits des espèces, les résultats sont moins

probants. Même si les Réseaux de Neurones donnent des résultats encourageants, nous n'avons pas réussi à construire un modèle fiable capable de prédire les groupes avec une précision satisfaisante. Les résultats sur les importances des variables ne sont pas assez consistants pour pouvoir en déduire un lien évident entre un ou plusieurs traits et les groupes de synchronie. Cependant, notre méthodologie ouvre la porte à d'autres analyses similaires.

## 1.5 Informations sur les chapitres

Le premier chapitre sur l'estimation des normes de Schatten et du rang effectif est basé sur des travaux communs effectués avec Nicolas Verzelen. Ces travaux donneront lieu à un article qui sera publié prochainement.

Les travaux présentés dans le deuxième chapitre sur la détection de rupture dans les arbres sont issus du travail commun de Guillem Rigaill, Nicolas Verzelen, Christophe Giraud et moi-même. Les algorithmes implémentés l'ont été avec l'aide de Guillem Rigaill. Ces algorithmes seront inclus dans un package R qui sera créé et mis en ligne. Les différentes parties du chapitre seront inclus dans un article qui verra le jour et sera publié prochainement.

Le troisième et dernier chapitre de cette thèse se base sur des travaux communs effectués avec Théophile Olivier et Christophe Giraud. La deuxième section de ce chapitre contient un article publié en écologie sur le lien entre la synchronie et les tendances temporelles à long terme.

# CHAPTER 2

# ESTIMATION OF EFFECTIVE RANK

## 2.1 Introduction

In many modern problems, scientists are faced with a large data matrix $\mathbf{Y}$, which is often assumed to be sum of a signal matrix $\mathbf{A}$ and a normally distributed noise. Under some structural assumptions on the signal such as small rank, it is possible to recover precisely the signal matrix $\mathbf{A}$, for instance with singular value thresholding methods (see e.g. [21, 41]). In this work, we focus our attention on estimating specific functionals of $\mathbf{A}$ related to its rank. On the one hand, the rank or the effective rank of the signal can help assessing the relevance of low-rank based procedures. On the other hand, evaluating the rank (or the effective rank) of $\mathbf{A}$ may also be an objective per se as a characterization of the complexity of the signal $\mathbf{A}$. We argue in Section 2.1.1 below that the problem of the effective rank of a matrix $\mathbf{A}$ mostly boils down to estimating Schatten norms of $\mathbf{A}$. This manuscript is dedicated to the latter problem.

To be more specific, we consider the model

$$\mathbf{Y} = \mathbf{A} + \mathbf{E} \,, \tag{2.1}$$

where $\mathbf{A}$ is the unobserved signal, $\mathbf{E}$ is a $p \times q$ noise matrix with independent entries following a standard normal distribution. Without loss of generality the noise variance is set to one. Also, without loss of generality, we assume throughout the manuscript that $p \geq q$.

Given a $p \times q$ matrix $\mathbf{A}$, we write $\sigma_1(\mathbf{A}) \geq \sigma_2(\mathbf{A}) \ldots \geq \sigma_q(\mathbf{A}) \geq 0$ for its ordered sequence of singular values. The rank of a matrix $\mathbf{A}$ corresponds to its number of positive singular values. In this manuscript, we will be interested in the Schatten norms of $s$. For any $s \geq 1$, the $s$-Schatten norm of $\mathbf{A}$ is defined as the $l_s$ norm of its sequence of singular values, that is

$$\|\mathbf{A}\|_s^s = \sum_{i=1}^{q} \sigma_i^s(\mathbf{A}) \,. \tag{2.2}$$

This manuscript is dedicated to the problem of estimating the Schatten norms of $\mathbf{A}$ from a single noisy observation $\mathbf{Y}$. Before describing our contribution, we explain how Schatten norms are related to effective rank measures.

### 2.1.1 Effective Rank of a matrix

Since the rank of a matrix is very sensitive to small perturbations, it is difficult to estimate it from $\mathbf{Y}$. Furthermore, a large rank matrix $\mathbf{A}$ may have only few large singular values together with many small singular values. For such a matrix, the rank is poorly informative on the structure of $A$. As an alternative, various notions of effective ranks have been introduced, which may be interpreted as an

effective number of non-zero singular values. Besides, many high-dimensional or infinite-dimensional probabilist results naturally depend on effective ranks (e.g. [66]). In [66], Koltchinskii and Lounici consider for a non-negative symmetric matrix $\mathbf{\Sigma}$ the indicator $\mathrm{tr}[\mathbf{\Sigma}]/\|\mathbf{\Sigma}\|_\infty = \|\mathbf{\Sigma}\|_1/\|\mathbf{\Sigma}\|_\infty$. For rectangular matrices $\mathbf{A}$, one may think of two extensions of this indicator, depending on whether we work with the singular values of $\mathbf{A}$ or that of the square matrix $\mathbf{A}^T\mathbf{A}$.

$$\mathrm{ER}_{1,\infty}(\mathbf{A}) = \frac{\|\mathbf{A}\|_1}{\|\mathbf{A}\|_\infty} \quad ; \qquad \mathrm{ER}_{2,\infty}(\mathbf{A}) = \mathrm{ER}_{1,\infty}(\mathbf{A}^T\mathbf{A}) = \frac{\|\mathbf{A}\|_2^2}{\|\mathbf{A}\|_\infty^2} \ . \tag{2.3}$$

Other notions of effective ranks are based on entropy measures. For instance Roy and Veterlli [107] introduce the Shannon effective rank:

$$\mathrm{ER}_{1,1}(\mathbf{A}) = \exp\left[ -\sum_{i=1}^{q} \frac{\sigma_i(\mathbf{A})}{\|\mathbf{A}\|_1} \log\left( \frac{\sigma_i(\mathbf{A})}{\|\mathbf{A}\|_1} \right) \right] \ , \tag{2.4}$$

which corresponds to the Shannon effective number of the distribution induced by the probability vector $(\sigma_i(\mathbf{A})/\|\mathbf{A}\|_1)$. More generally, one may extend any effective number based either on probability vector associated to the singular values of $\mathbf{A}$ $(\sigma_i(\mathbf{A})/\|\mathbf{A}\|_1)$ or to the square matrix $\mathbf{A}^T\mathbf{A}$ $(\sigma_i^2(\mathbf{A})/\|\mathbf{A}\|_2^2)$. For the purpose of biodiversity analysis, a large class of diversity measures are considered [60]. In particular, for any positive $s > 0$ and different from 1, the Hill's effective number [60] which is derived from Renyi entropy may straightforwardly extend to rank effective number

$$\mathrm{ER}_{1,s}(\mathbf{A}) = \left( \frac{\|\mathbf{A}\|_s}{\|\mathbf{A}\|_1} \right)^{s/(1-s)} \ ; \qquad \mathrm{ER}_{2,s}(\mathbf{A}) = \mathrm{ER}_{1,s}(\mathbf{A}^T\mathbf{A}) = \left( \frac{\|\mathbf{A}\|_{2s}}{\|\mathbf{A}\|_2} \right)^{2s/(1-s)} \ . \tag{2.5}$$

When all the non-zero singular values of $\mathbf{A}$ are equal, all these effective rank measures correspond to the rank of $\mathbf{A}$. However, these measures differ in the way they treat heterogeneous values for the singular values. In short, smaller $s$ values in $\mathrm{ER}_{1,s}(\mathbf{A})$ and $\mathrm{ER}_{2,s}(\mathbf{A})$ are more prone to take into account smaller singular values in the effective rank. See [60] and references therein for further discussions.

Most work around effective ranks consider noiseless setting [107] and the matrix $\mathbf{A}$ is sometimes allowed to be partially observed (e.g [6]). In this work, we tackle the general problem of estimating the effective rank of $\mathbf{A}$ relying on the noisy observation $\mathbf{Y}$. To the exception of the Shannon effective rank (2.4), all the other effective rank measures are ratio of Schatten norms of $\mathbf{A}$, so that evaluating the former amounts to estimating well the latter. Besides, if some Schatten norms $\|\mathbf{A}\|_s$ are provably much more difficult to estimate than other ones, this may lead the statistician to favor some specific effective rank measures.

### 2.1.2 Our Contribution

In this work, we first consider the case of the Frobenius norm and prove that the simple estimator $(\|\mathbf{Y}\|_2^2 - pq)_+^{1/2}$, where $x_+ = \max(x, 0)$ achieves the optimal risk $(pq)^{1/4}$. Interestingly, this risk $(pq)^{1/4}$ cannot be improved by any estimator even if one assumes that the matrix $\mathbf{A}$ has a rank at most one. Then, we establish that a non-linear transformation of $\sigma_1(\mathbf{Y})$ estimates $\|\mathbf{A}\|_\infty = \sigma_1(\mathbf{A})$ with the same optimal risk $(pq)^{1/4}$.

Regarding general even norms $\|\mathbf{A}\|_{2k}$, we first remark that $\|\mathbf{A}\|_{2k}^{2k} = tr[(\mathbf{A}^T\mathbf{A})^{2k}]$ is a polynomial with respect to the entries of $\mathbf{A}$. This allows us to build an unbiased estimator $U_k$ of $\|\mathbf{A}\|_{2k}^{2k}$ based on Hermite polynomials. Relying on the invariance of $\|\mathbf{A}\|_{2k}^{2k}$ by left and right orthogonal transformations, we establish that this estimator has a simple expression as an algebraic combination of $tr[(\mathbf{YY})^s]$. Then, we prove that the plug-in estimator $(U_k)_+^{1/2k}$ achieves again the optimal risk $(pq)^{1/4}$ for all matrices $\mathbf{A}$.

Finally, we have some partial results for non-even Schatten norms $\|\mathbf{A}\|_s$. First, we propose an estimator achieving the risk $q(pq)^{1/4}$. Second, we prove in the specific case of the nuclear norm $\|\mathbf{A}\|_1$, the risk $(pq)^{1/4}$ is not achievable and that no estimator exhibits an error smaller than $q/\sqrt{\log(q)}$.

As an applications of our findings, we build an estimator of the effective rank $\mathrm{ER}_{2,\infty}$ and control its error in probability. Besides, we argue why Effective ranks measures $\mathrm{ER}_{2,s}$ where $s \geq 2$ is an integer are easier to estimate than other definition (2.3,2.4,2.5).

### 2.1.3 Related Literature

**Low rank Matrix Estimation and Detection.** There is a long line of work regarding the estimation of the matrix $\mathbf{A}$ when $\mathbf{A}$ is assumed to be low-rank or approximately low-rank. Procedures based on singular value thresholding procedures [21, 41], that is estimating $\mathbf{A}$ by setting small singular values of $\mathbf{Y}$ to 0, have been proved to achieve near optimal performances. Donoho and Gavish [31] have assessed the asymptotic risk of singular value soft-thresholding procedures in an asymptotic framework where the rank is proportional to $q$ and $p$ is proportionnal to $q$. [93, 111] have introduced other non-linear singular values shrinkage estimators that turn out to achieve smaller risks (the constant in front of the rates is slightly better), at least in an asymptotic framework where $p/q \to c \in (0, \infty)$ and $\mathbf{A}$ has finite rank low-rank. As explained in Section 2.3, some of our Schatten norm estimators are reminiscent of those non linear shrinkage functions.

**Asymptotic estimation of the singular values in the asymptotic $p/q \to c$ .** In asymptotic matrix analysis, most work have been devoted to the setting where $p/q \to c \in (0, \infty)$. In that literature, the model (2.1) is coined as the information plus noise model [25]. Given the symmetric matrix $\frac{1}{p}\mathbf{Y}^T\mathbf{Y}$ with ordered eigenvalue $\lambda_1, \ldots, \lambda_q$, define the spectral measure as $\mu_{\mathbf{Y}^T\mathbf{Y}} := \frac{1}{q}\sum_{i=1}^{q} \delta_{\lambda_i}$. In the specific case where $\mathbf{A} = 0$, it has been established [2] that $\mu_{\mathbf{Y}^T\mathbf{Y}}$ converges weakly towards the Marchenko-Pastur distribution. More generally, when the spectral measure of $\frac{1}{p}\mathbf{A}^T\mathbf{A}$ converges to a probability measure, [123] have characterized the limit of $\mu_{\mathbf{Y}^T\mathbf{Y}/p}$ through its Stieljes transform. Given that characterization, one may try to invert the mapping from the limit of $\mu_{\mathbf{A}^T\mathbf{A}/p}$ to the limit of $\mu_{\mathbf{Y}^T\mathbf{Y}/p}$ to estimate some functional of the spectrum of $\mathbf{A}^T\mathbf{A}$ from $\mathbf{Y}$. In some specific case, where $\mathbf{A}$ has asymptotically a finite number $k$ of distinct singular values, consistency of these singular values has been derived. See [25, Ch.8] and references therein for more details. Given a consistent estimation of the spectral measure, on may plug it to estimate the Schatten norms. However, results in this line of work are not comparable to ours as they are restricted to very specific settings.

**Matrix Sketching.** In computer science, there is an active line of research around the problem of matrix sketching which can be defined as follows: the practitioner has access to some entries of the matrix $\mathbf{A}$ or to some linear combination of the entries of the matrix $\mathbf{A}$. In this noiseless problem, one aims at recovering functionals of $\mathbf{A}$ such as a Schatten norm or the rank using the smallest possible budget. See [6, 63, 75]. As in our noisy framework, [75] emphasizes that estimating even Schatten norms using sketches seems easier than estimating non-even Schatten norms. Apart from this, the techniques for both the lower and upper bounds highly differ from our settings.

**Estimation of non linear functionals.** Schatten norms are non-linear functionals of the entries of $\mathbf{A}$. Vectorizing the matrices $\mathbf{Y}$, $\mathbf{A}$, and $\mathbf{E}$, we may rewrite our problem in the Gaussian sequence model. In particular, estimating the 2-Schatten norm is equivalent to estimating the $l_2$ norm of a vector in the Gaussian sequence model. See [23, 32] and references therein for an account of work of quadratic functional estimation. More generally, the estimation of the $l_r$ norm of a vector has been addressed in [18, 48, 72]. However, apart from the specific case $s = 2$, $s$-Schatten norms do no correspond to $l_r$ norm and the methodology developed in the Gaussian sequence model does not extend to our setting. In discrete distribution estimation, optimal rates of convergence for Shannon entropy, Renyi entropy have been respectively established in [134] and [58] (see also [49]). Closer to our settings, Kong and Valiant [67] have considered the problem of estimating even Schatten norms of a covariance matrix $\mathbf{\Sigma}$ given a $n$-sample of a mean zero random vector with covariance matrix

$\mathbf{\Sigma}$. Interestingly, their estimator exhibit a low computational complexity and is unbiased for general noise distributions. However, they do not assess the optimality of their estimators.

### 2.1.4 Notation

In this work, $c$, $c_1$, $c'$ denote numerical positive constants that may vary from line to line.

Section 2.2 is devoted to the simple case of Frobenius norm estimation and provides both minimax upper and lower bounds. In Section 2.3, we consider the problem of estimating the operator norm ($\|\mathbf{A}\|_\infty$). General even norms are addressed in Section 2.4, whereas non-even norms are addressed in Section 2.5. Finally, we come back to the problem of effective estimation together with some open question in the discussion Section. Proofs are postponed to the appendix.

## 2.2 Frobenius Norm

### 2.2.1 Upper bound

We first estimate $\|\mathbf{A}\|_2^2 = \sum_{i,j} \mathbf{A}_{ij}^2$ and then we deduce an estimator of $\|\mathbf{A}\|_2$. Since $\|\mathbf{Y}\|_2^2$ follows a non-central $\chi^2$ distribution, we derive from standard computations for the normal random variables that $\mathbb{E}[\|\mathbf{Y}\|_2^2] = \|\mathbf{A}\|_2^2 + pq$. This leads us to the following unbiased estimator

$$U_1 = \|\mathbf{Y}\|_2^2 - pq \ , \tag{2.6}$$

whose risk is upper bounded in the next proposition.

**Proposition 1.** *For any $p \times q$ matrix $\mathbf{A}$ it holds that*

$$\mathbb{E}\left[|U_1 - \|\mathbf{A}\|_2^2|\right] \leq \sqrt{2pq} + 2\|\mathbf{A}\|_2 \ .$$

Regarding $\|\mathbf{A}\|_2$, we simply plug the estimator $U_1$ of $\|\mathbf{A}\|_2^2$. We take $(U_1)_+^{1/2}$.

**Proposition 2** (Risk of $(U_1)_+^{1/2}$)**.** *For any $p \times q$ matrix $\mathbf{A}$, it holds that*

$$\mathbb{E}\left[|(U_1)_+^{1/2} - \|\mathbf{A}\|_2|\right] \leq 3(pq)^{\frac{1}{4}}.$$

**Remark**: It turns out that $\|\mathbf{A}\|_2^2$ (resp. $\|\mathbf{A}\|_2$) corresponds to the square $l_2$ norm (resp. $l_2$ norm) of the vectorized version of $\mathbf{A}$. As as consequence, it is equivalent to estimating the square $l_2$ norm (resp. $l_2$ norm) of of a noisy vector. This problem has been thoroughly studied in the literature. See [23] and references therein. In particular, the counterparts of $U_1$ and $(U_1)_+^{1/2}$ in the vector setting were already analyzed in [23]. Still, we provide proofs of these two propositions for the sake of completeness.

### 2.2.2 Minimax lower bound

In this subsection, we asses the optimality of Proposition 2 by providing a minimax lower bound.

**Theorem 1.** *There exists a numerical constant $c > 0$ such that*

$$\inf_{\widehat{T}} \sup_{\mathbf{A}:\ \mathrm{rank}(\mathbf{A}) \leq 1} \mathbb{E}\left[|\widehat{T} - \|\mathbf{A}\|_2|\right] \geq c(pq)^{1/4} \ .$$

It turns out that one cannot estimate $\|\mathbf{A}\|_2$ uniformly over all matrices $\mathbf{A}$ at a rate faster than $(pq)^{1/4}$. More importantly, even when one restricts itself to the much simpler class of rank at most one matrices $\mathbf{A}$, it is impossible to achieve a faster rate. This theorem is proved using Le Cam's approach by constructing a discrete prior distribution on the space of rank 1 matrix (see the proof for more details).

Since for a rank 1 matrix $\mathbf{A}$, all Schatten norms are equal to $\sigma_1(\mathbf{A})$, we straightforwardly deduce a lower bound for all Schatten norms.

**Corollary 1.** *For any real number $r \geq 1$, we have*

$$\inf_{\widehat{T}} \sup_{\mathbf{A}\in\mathbb{R}^{p\times q}} \mathbb{E}\left[|\widehat{T} - \|\mathbf{A}\|_r|\right] \geq \inf_{\widehat{T}} \sup_{\mathbf{A}:\ \mathrm{rank}(\mathbf{A})\leq 1} \mathbb{E}\left[|\widehat{T} - \|\mathbf{A}\|_r|\right] \geq c(pq)^{1/4} ,$$

*where $c$ is as in Proposition 1.*

## 2.3 Operator Norm $\|\mathbf{A}\|_\infty$

In this subsection, we address the case $k = \infty$ which amounts to estimating $\|\mathbf{A}\|_\infty = \sigma_1(\mathbf{A})$. In view of Corollary 1, the optimal estimation risk is at least of the order of $(pq)^{1/4}$ for general matrices $\mathbf{A}$. Besides, it is of the order $(pq)^{1/4}$ for rank one matrices $\mathbf{A}$. Since $\mathbf{Y} = \mathbf{A} + \mathbf{E}$, we may be tempted to use plug-in estimators of the form $\|\mathbf{Y}\|_\infty$ or $\|\mathbf{Y}\|_\infty - \mathbb{E}\,[\|\mathbf{E}\|_\infty]$.

**Proposition 3.** *For any matrix $\mathbf{A}$, we have*

$$\mathbb{E}[|\|\mathbf{Y}\|_\infty - \|\mathbf{A}\|_\infty|] \leq \sqrt{p} + \sqrt{q} \qquad \textit{and} \qquad \mathbb{E}[|\|\mathbf{Y}\|_\infty - \mathbb{E}\,[\|\mathbf{E}\|_\infty] - \|\mathbf{A}\|_\infty|] \leq 2[\sqrt{p} + \sqrt{q}]$$

*Conversely, there exist two positive numerical constants $c$ and $c'$ such that, for $p$ large enough,*

$$\sup_{\mathbf{A},\,\mathrm{rank}(\mathbf{A})\leq 1} \mathbb{E}[|\|\mathbf{Y}\|_\infty - \|\mathbf{A}\|_\infty|] \geq c'\sqrt{p} \quad \textit{and} \quad \sup_{\mathbf{A},\,\mathrm{rank}(\mathbf{A})\leq 1} \mathbb{E}[|\|\mathbf{Y}\|_\infty - \mathbb{E}\,[\|\mathbf{E}\|_\infty] - \|\mathbf{A}\|_\infty|] \geq c\sqrt{p} .$$

When $p$ is of the same order as $q$ (nearly square matrices), $\sqrt{p}$ is of the same order as $(pq)^{1/4}$ and these simple plug-in estimators nearly match the minimax lower bound. However, for highly rectangular matrices, these estimators achieve a much slower rate than the minimax lower bound.

To improve the rate from $\sqrt{p}$ to $(pq)^{1/4}$, we start by estimating the operator norm $\|\mathbf{A}^T\mathbf{A}\|_\infty = \sigma_1^2(\mathbf{A})$. Define the $q \times q$ matrix $\mathbf{W} = \mathbf{Y}^T\mathbf{Y} - p\mathbf{I}$ where $\mathbf{I}$ is the identity matrix. Simple calculations lead to $\mathbb{E}[\mathbf{W}] = \mathbf{A}^T\mathbf{A}$. This motivates us to to estimate $\sigma_1^2(\mathbf{A})$ by $\sigma_1(\mathbf{W}) = \sigma_1^2(\mathbf{Y}) - p$.

**Proposition 4.** *For any matrix $\mathbf{A}$ it holds that*

$$\mathbb{E}\left[|\sigma_1^2(\mathbf{Y}) - p - \sigma_1^2(\mathbf{A})|\right] \leq 3\sqrt{pq} + 4\|\mathbf{A}\|_\infty\sqrt{q} + 8e^{-p/4} + \sqrt{64\pi p} ,$$

*and*

$$\mathbb{E}\left[|\left(\sigma_1^2(\mathbf{Y}) - p\right)_+^{1/2} - \|\mathbf{A}\|_\infty|\right] \leq c\,(pq)^{\frac{1}{4}} \ , \tag{2.7}$$

*for a positive numerical constant $c$.*

This new estimator $\left(\sigma_1^2(\mathbf{Y}) - p\right)_+^{1/2}$ achieves the optimal rate $(pq)^{1/4}$. As for the Frobenius norm, a low-rank assumption on $\mathbf{A}$ does not ease the operator norm estimation problem.

In an asymptotic framework where $p/n \to c' \in (0,\infty)$ and where $\mathbf{A}$ has finite rank $r$, Shabalin and Nobel [111] have introduced an estimator $\widehat{\mathbf{A}}$ of $\mathbf{A}$ such that $\sigma_i^2(\widehat{\mathbf{A}}) = \frac{1}{2}\left[\sigma_i^2(\mathbf{Y}) - p - q + \sqrt{[\sigma_i^2(\mathbf{Y}) - p - q]^2 - 4pq}\right]$ when $\sigma_i(\mathbf{Y})$ is large enough. When $q$ is small compared to $p$, then this $\sigma_1(\widehat{\mathbf{A}})$ does not much differ from our estimator $\sqrt{(\sigma_1(\mathbf{Y}) - p)_+}$. In fact, it is not hard to prove that $\sigma_1(\widehat{\mathbf{A}})$ estimates $\|\mathbf{A}\|_\infty$ at the optimal rate $(pq)^{1/4}$ in analogy to (2.7).

## 2.4 Estimation of General even norms

In this subsection, we first introduce an unbiased estimator of $\|\mathbf{A}\|_{2k}^{2k}$ and then plug it to estimate $\|\mathbf{A}\|_{2k}$. Recall that, when $k$ is an integer,

$$\|\mathbf{A}\|_{2k}^{2k} = \mathrm{tr}[(\mathbf{A}^T\mathbf{A})^k] = \sum_{i_1,\ldots,i_k=1}^{p} \sum_{j_1,\ldots,j_k=1}^{q} \prod_{t=1}^{k} \mathbf{A}_{i_t j_t}\mathbf{A}_{i_{t+1} j_t} ,$$

where by conventions $i_{k+1} = i_1$. Obviously, $\|\mathbf{A}\|_{2k}^{2k}$ is a polynomial of order $2k$ with respect to the entries of $\mathbf{A}$. Since $\mathbf{Y}_{ij} \sim \mathcal{N}(\mathbf{A}_{ij}, 1)$, one may unbiasly estimates each using polynomials in the entries of $Y$.

Write $\phi(y) = e^{-y^2/2}(2\pi)^{-1/2}$ for the density of the standard normal random variables. For positive integers $r$, we define the Hermite polynomial of degree $r$ by the equation

$$\frac{d^r}{dy^r}\phi(y) = (-1)^r H_r(y)\phi(y) \tag{2.8}$$

It is well known (e.g. [18]) that, for $Z \sim \mathcal{N}(x, 1)$, $\mathbb{E}[H_r(Z)] = x^r$. Given $i = (i_1, \ldots, i_k)$ and $j = (j_1, \ldots, j_k)$, we define $N_{rs}(ij)$ as the number of occurrences of $rs$ in $(i_t, j_t)$ and $(i_{t+1}, j_t)$ with $t = 1, \ldots, k$ and $i_{k+1} = i_1$. Then, the estimator

$$U_k = \sum_{i_1,\ldots,i_k=1}^{p} \sum_{j_1,\ldots,j_k=1}^{q} \prod_{r=1}^{p}\prod_{s=1}^{q} H_{N_{rs}(i,j)}(\mathbf{Y}_{rs}) \tag{2.9}$$

satisfies $\mathbb{E}[U_k] = \mathrm{tr}[(\mathbf{A}^T\mathbf{A})^k] = \|\mathbf{A}\|_{2k}^{2k}$.

Although it is easy to deduce the expectation $\mathbb{E}[U_k]$ from (2.9), this definition is not convenient for practical computations as it may suggest that $O(p^k q^k)$ operations are needed. It turns out in the next proposition that $U_k$ is an algebraic combination of Schatten norms of $Y^TY$. Thus, it can be computed in $O((p+k)q^2)$ operations.

Given a positive integer $l$, $\mathcal{S}[l]$ stands for the collections of nondecreasing positive integer values vectors $s$ whose sum equals $l$. In other words, $s$ satisfies $1 \le s_1 \le s_2 \le \ldots s_{|s|}$ and $\sum_i s_i = l$.

**Proposition 5.** *There exists coefficients $\alpha_s$ (depending only on $p$ and $q$) such that*

$$U_k = \mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})^k] + \alpha_0 + \sum_{l=1}^{k-1}\sum_{s\in\mathcal{S}[l]} \alpha_s \prod_{i=1}^{|s|} \mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})^{s_i}] \ . \tag{2.10}$$

The proof of this proposition relies on the orthogonal invariance of the Gaussian distribution and on the representation of symmetric polynomials by newton sums.

**Remark**: In the specific case $k = 1$, we have already used $U_1 = \mathrm{tr}[\mathbf{Y}^T\mathbf{Y}] - pq$. The coefficients in (2.10) are implicit. Still, by computing the expectation of $\mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})^s]$, one can recursively debias $\mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})^k]$ to derive the expression of $U_k$. As an example, the following proposition provides an explicit expression of $U_2$ and $U_3$.

**Proposition 6.** *For any $q \le p$, we have*

$$\begin{aligned} U_2 &= \mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})^2] - 2(p+q+1)\,\mathrm{tr}[\mathbf{Y}^T\mathbf{Y}] + pq(1+p+q) \ ; \\ U_3 &= \mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})^3] - 3(p+q+1)\,\mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})^2] + 3[p^2+q^2+pq+p+q-2]\,\mathrm{tr}[\mathbf{Y}^T\mathbf{Y}] + pq\left[-p^2-q^2+5\right] \ . \end{aligned}$$

As a consequence of Proposition 5 the distribution of $U_k$ is invariant by left and right orthogonal transformations of $\mathbf{Y}$. Hence, we may assume henceforth that $\mathbf{A}$ is null outside its diagonal and that $\mathbf{A}_{ii} = \sigma_i(\mathbf{A})$ for $i = 1, \ldots, q$. The next proposition bounds the risk of $U_k$. In the sequel, $c_{\exp}(k)$ is short for $c_1 k^{c_2 k}$ where $c_1$ and $c_2$ are numerical positive constants whose value may change from line to line.

**Proposition 7.** *For any integer $k \ge 2$, and any $p \ge q$, we have*

$$\mathrm{Var}\,(U_k) \ \le \ c_{\exp}(k)\left[(pq)^k + p\|\mathbf{A}\|_{4k-4}^{4k-4} + \|\mathbf{A}\|_{4k-2}^{4k-2}\right]$$

.

The proof is based on some combinatorial arguments: starting from the Hermite definition (2.9) of $U_k$, we first establish necessary conditions on sequences $(i_1, j_1), \ldots, (i_k, j_k)$ and $(i'_1, j'_1), \ldots, (i'_k, j'_k)$ so that the corresponding covariance is non-zero. Then, we count the remaining sequences building on earlier ideas on random matrix analysis [2]. From this proposition, we deduce a risk bound for the estimator $(U_k)_+^{1/2k}$ of $\|\mathbf{A}\|_{2k}$.

**Corollary 2.** *For any positive integer $k$, and any $p \times q$ matrix $\mathbf{A}$ one has*

$$\mathbb{E}\left[\left|(U_k)_+^{1/(2k)} - \|\mathbf{A}\|_{2k}\right|\right] \leq c_{\exp}(k)(pq)^{1/4} \ . \tag{2.11}$$

Interestingly, the estimator $(U_k)_+^{1/(2k)}$ achieves the optimal rate $(pq)^{1/4}$ for any even norm. This implies that, as for Frobenius norm estimation, estimating a small rank matrix $\mathbf{A}$ is as difficult as estimating a full rank matrix.

## 2.5 Estimation of General Norms

The purpose of this section is to estimate the Schatten norm $\|\mathbf{A}\|_s$ for a real number $s \geq 1$. In contrast to even Schatten norms, one cannot devise unbiased estimators of such norms as exemplified by the following lemma for $\|\mathbf{A}\|_1$.

**Lemma 1.** *For all square integrable estimators $\varphi(\mathbf{Y})$ and all $r > 0$, there exists a matrix $\mathbf{A}$ such that $\|\mathbf{A}\|_1 \leq r$ and $\mathbb{E}[\varphi(\mathbf{Y})] \neq \|\mathbf{A}\|_1$*

*Proof of Lemma 1.* For any $t \in \mathbb{R}$, consider the matrix $\mathbf{A}_t$ such that $(\mathbf{A}_t)_{1,1} = t$ and $(\mathbf{A}_t)_{i,j} = 0$ otherwise. Obviously $\|\mathbf{A}_t\|_1 = |t|$. The function $t \mapsto \|\mathbf{A}_t\|_1$ is not differentiable at 0. In contrast, the function $t \mapsto \mathbb{E}[\varphi(\mathbf{Y})] = (2\pi)^{-pq/2} \int \varphi(\mathbf{Y}) e^{-\|\mathbf{Y}\|_2^2/2 - t^2/2 + t\mathbf{Y}_{11}} d\mathbf{Y}$ is differentiable by Lebesgue's dominated convergence theorem. As a consequence, both functions cannot match in any neighborhood of 0. □

As a starting point and to assess the quality of our more refined procedure, we consider the simple plug-in estimator $\|\mathbf{Y}\|_s$ that simply evaluates the $s$-Schatten norm of the observed matrix $\mathbf{Y}$.

**Lemma 2.** *For any matrix $\mathbf{A} \in \mathbb{R}^{p \times q}$, we have $\mathbb{E}\left[\left|\|\mathbf{Y}\|_s - \|\mathbf{A}\|_s\right|\right] \leq 2q\sqrt{p}$.*

*Proof of Lemma 2.* From triangular inequality, we deduce that $\mathbb{E}\left[\left|\|\mathbf{Y}\|_s - \|\mathbf{A}\|_s\right|\right] \leq \mathbb{E}\left[\|\mathbf{E}\|_s\right]$. Since $\|\mathbf{E}\|_s \leq q\|\mathbf{E}\|_\infty$, we simply need to bound the expected operator norm of $\mathbf{E}$. By Lemma 9 in the appendix, we have $\mathbb{E}[\|\mathbf{E}\|_\infty] \leq \sqrt{p} + \sqrt{q}$, which concludes the proof. □

**Remark**: The risk bound achieved by $\|\mathbf{Y}\|_s$ is much larger than what we have achieved for even norms. Even for square matrices ($p = q$), Lemma 2 implies a rate of the order $p\sqrt{p}$ whereas even norms can be estimated at the rate $\sqrt{p}$.

**Remark**: As in Section 2.3, we argue the risk upper bound from Lemma 2 for the cannot be significantly improved. Indeed, consider the specific case, where $\mathbf{A} = 0$. For any $p \geq 4q$, we have $\mathbb{E}[\left|\|\mathbf{Y}\|_s - \|\mathbf{A}\|_s\right|] \geq cq\sqrt{p}$. Indeed, $\|\mathbf{Y}\|_s \geq q\sigma_q(\mathbf{Y})$, and we deduce from Lemma 10, that with probability higher than $1 - e^{-q/8}$, $\sigma_q(\mathbf{Y}) \geq \sqrt{p} - 3/2\sqrt{q} \geq \sqrt{p}/4$. The result follows.

The next estimator attempts to improve the $\sqrt{p}$ factor by correcting the singular values of $\mathbf{Y}$. Consider the matrix $\mathbf{W} = \mathbf{Y}^T\mathbf{Y} - p\mathbf{I}_q$. From simple computation, we deduce that

$$\mathbb{E}[\mathbf{W}] = \mathbf{A}^T\mathbf{A} + \mathbb{E}[\mathbf{E}^T\mathbf{A} + \mathbf{A}^T\mathbf{E}] + \mathbb{E}[\mathbf{E}^T\mathbf{E} - q\mathbf{I}_p] = \mathbf{A}^T\mathbf{A} \ .$$

As a consequence, $\mathbf{W}$ is an unbiased estimator $\mathbf{A}^T\mathbf{A}$, which could suggest that the eigenvalues of $\mathbf{W}$ are close to that of $\mathbf{A}^T\mathbf{A}$, which in turn are the square of the singular values of $\mathbf{A}$. Since the $i$-th eigenvalues of $\mathbf{W}$ equals $\sigma_i^2(\mathbf{Y}) - p$, this leads us to considering

$$T_s = [\sum_{i=1}^{q} [(\sigma_i^2(\mathbf{Y}) - p)_+]^{s/2}]^{1/s} \ . \tag{2.12}$$

**Proposition 8.** *There exists a numerical constant $c$ such that for any matrix $\mathbf{A} \in \mathbb{R}^{p\times q}$, we have*

$$\mathbb{E}[|T_s - \|\mathbf{A}\|_s|] \leq \mathbb{E}[\sum_{i=1}^{q} |(\sigma_i^2(\mathbf{Y}) - p)_+^{1/2} - \sigma_i(\mathbf{A})|] \leq cq(pq)^{1/4} \ . \tag{2.13}$$

The proof of this result relies on interlacing inequalities for eigenvalues (Corollary III. 1.5 in [12]). In comparison to the naive plug-in estimator, the $\sqrt{p}$ factor has been replaced by $(pq)^{1/4}$, which for highly rectangular matrices can be much smaller. In any case, the risk bound (2.13) is still $q$ times higher than the optimal risk for even norms.

Denote $\mathcal{D}_{p,q}$ the collection of "diagonal" rectangular matrices of size $p \times q$, that is the collections of matrices $\mathbf{A}$ such that $\mathbf{A}_{ij} = 0$ for all $i \neq j$. The following proposition provides a lower bound for the estimation risk of the nuclear norm.

**Proposition 9.** *There exists a positive constant $c$ such that the following holds*

$$\inf_{\widehat{T}} \sup_{\mathbf{A}} \mathbb{E}[|\widehat{T} - \|\mathbf{A}\|_1|] \geq \inf_{\widehat{T}} \sup_{\mathbf{A}\in\mathcal{D}_{p,q}} \mathbb{E}[|\widehat{T} - \|\mathbf{A}\|_1|] \geq c\frac{q}{\sqrt{\log(q)}} \ . \tag{2.14}$$

This proposition entails that, at least when $\mathbf{A}$ is not too rectangular, estimation of the nuclear norm is much harder than estimation of even norms. However, regarding the lower and upper bounds the bounds only match up to a factor $(pq)^{1/4}\sqrt{\log(q)}$.

**Future work and Conjecture**: Proposition 9 is based on a reduction of the nuclear norm estimation problem to the problem of estimating the $l_1$ norm of a vector $\theta$ in the Gaussian sequence model. For the latter problem, [18] and [48] devised optimal procedures based on polynomial approximation: the general idea is to approximate the $l_1$ norm $|\theta|_1$ by linear combination of even norm $|\theta|_{2k}^{2k}$, which in turn can be estimated unbiasly. The resulting estimator is then chosen to achieve a trade-off between the approximation error and the variance of estimation. In the future, we plan to apply this methodology to approximate $\|\mathbf{A}\|_1$ (or more generally $\|\mathbf{A}\|_s$) by a polynomial with respect to even norms $\|\mathbf{A}\|_{2k}^{2k}$. We expect that the risk of the corresponding estimator improves our $q(pq)^{1/4}$ bound by some polylogarithmic factor, that we conjecture to be optimal.

## 2.6 Discussion

### 2.6.1 Application to Effective rank estimation

Coming back to the problem of estimating the Effective rank of a noisy matrix, we observe from our work on Schatten norm that one should favor ratio of even Schatten norms as they are much easier to estimate. Let us further focus on this specific choice

$$\mathrm{ER}_{2,\infty}(\mathbf{A}) = \frac{\|\mathbf{A}\|_2^2}{\|\mathbf{A}\|_\infty} \ . \tag{2.15}$$

From Sections 2.2 and 2.3, we shall consider

$$\widehat{\mathrm{ER}}_{2,\infty}(\mathbf{A}) = \max\left[\frac{\mathrm{tr}[\mathbf{Y}^T\mathbf{Y} - q\mathbf{I}_p]}{\sigma_1[\mathbf{Y}^T\mathbf{Y} - q\mathbf{I}_p]}, 1\right] \ . \tag{2.16}$$

We cannot simply plug Propositions 1 and 4 to control this estimator as we only proved expectation bounds. The following result pushes the analysis slightly further to derive high probability deviation for the effective rank.

**Proposition 10.** *There exist numerical constants $c$–$c_3$ such that the following holds for any matrix $\mathbf{A}$ and for any $t > 0$. Provided that*

$$\|\mathbf{A}\|_2^2 \geq c(\sqrt{pqt} + t) \ , \ \text{and} \ \|\mathbf{A}\|_\infty \geq c\left[\sqrt{t} + (pq)^{1/4} + (pt)^{1/4}\right] \ ,$$

*then, with probability higher than* $1 - c_3 e^{-t}$*, it holds that*

$$\frac{\left|\widehat{\mathrm{ER}}_{2,\infty}(\mathbf{A}) - \mathrm{ER}_{2,\infty}(\mathbf{A})\right|}{\mathrm{ER}_{2,\infty}(\mathbf{A})} \leq c' \frac{\sqrt{pqt}}{\|\mathbf{A}\|_2^2} + \frac{\sqrt{pt} + \sqrt{pq}}{\|\mathbf{A}\|_\infty^2} + \frac{\sqrt{q} + \sqrt{t}}{\|\mathbf{A}\|_\infty} \ .$$

If $\sigma_1(\mathbf{A}) = \|\mathbf{A}\|_\infty$ is large compared to $(pq)^{1/4}$, the above proposition implies that the effective rank is well estimated. This condition is not surprising as we have shown in the proof of Theorem 1 that it is impossible to consistently decipher $\mathbf{A}$ from the null matrix when $\sigma_1(\mathbf{A})$ is of the order of $(pq)^{1/4}$.

### 2.6.2 Low-Rank matrices

The optimal risk of estimating even Schatten norms does not depend on the rank of $\mathbf{A}$. Estimating $\|\mathbf{A}\|_{2k}$ for a rank 1 matrix is as difficult (in terms of risk) than for a full rank matrix. The situation seems quite different for general norms. We have obtained slow convergence rates. As explained before, estimating the nuclear norm of a rank 1 matrix is possible at rate $(pq)^{1/4}$ since all Schatten norms are equal in this case, whereas nuclear norm estimation for full rank matrices can be much higher. An interesting direction for future research would be to explore more generally the impact of a small rank assumption on general Schatten norm estimation.

## 2.7 Proofs

### 2.7.1 Frobenius and operator norm

*Proof of Proposition 1.* The difference $U_1 - \|\mathbf{A}\|_2^2$ decomposes as

$$U_1 - \|\mathbf{A}\|_2^2 = \|\mathbf{Y}\|_2^2 - pq - \|\mathbf{A}\|_2^2 = \left(\|\mathbf{E}\|_2^2 - pq\right) + 2\langle \mathbf{A}, \mathbf{E}\rangle_2,$$

where $\langle \mathbf{B}, \mathbf{C}\rangle_2 = \mathrm{tr}(\mathbf{B}^T\mathbf{C})$. First, $\|\mathbf{E}\|_2^2$ follows a $\chi^2$ distribution with $pq$ degrees of freedom. Hence, we derive from Cauchy-Schwarz inequality that $\mathbb{E}\left[\left|\|\mathbf{E}\|_2^2 - pq\right|\right] \leq \mathrm{var}^{1/2}(\|\mathbf{E}\|_2^2) = \sqrt{2pq}$. Since $\mathbf{A}$ is deterministic, $\langle \mathbf{A}, \mathbf{E}\rangle_2 / \|\mathbf{A}\|_2$ follows a standard normal distribution and $\mathbb{E}\left[|\langle \mathbf{A}, \mathbf{E}\rangle_2|\right] \leq \|\mathbf{A}\|_2$. The result follows. □

*Proof of Proposition 2.* Since $|\sqrt{a} - \sqrt{b}| \leq \sqrt{|a-b|}$, we derive from Cauchy-Schwarz inequality and Proposition 1 that

$$\begin{aligned}
\mathbb{E}\left[\left|(U_1)_+^{1/2} - \|\mathbf{A}\|_2\right|\right] &\leq \mathbb{E}\left[\sqrt{\left|(U_1)_+^{1/2} - \|\mathbf{A}\|_2\right|}\right] \left[\mathbb{E}\left(\left|(U_1)_+ - \|\mathbf{A}\|_2\right|\right)\right]^{1/2} \\
&\leq \left[\mathbb{E}\left(\left|U_1 - \|\mathbf{A}\|_2\right|\right)\right]^{1/2} \leq \sqrt{(2pq)^{1/2} + 2\|\mathbf{A}\|_2} \ ,
\end{aligned}$$

which is smaller than $2(pq)^{1/4}$ as long as $\|\mathbf{A}\|_2 \leq \sqrt{pq}$. Now assume that $\|\mathbf{A}\|_2 \geq \sqrt{pq}$. Since $(x-y) = (x^2 - y^2)/(x+y)$, it follows again from Proposition 1 that

$$\mathbb{E}\left[\left|(U_1)_+^{1/2} - \|\mathbf{A}\|_2\right|\right] \leq \mathbb{E}\left[\frac{\left|U_1 - \|\mathbf{A}\|_2^2\right|}{\|\mathbf{A}\|_2}\right] \leq \frac{\sqrt{2pq} + 2\|\mathbf{A}\|_2}{\|\mathbf{A}\|_2} \leq 2 + \sqrt{2} \leq 3(pq)^{1/4} \ .$$

This concludes the proof. □

*Proof of Proposition 3.* By triangular inequality, we have $\mathbb{E}[|\|\mathbf{Y}\|_\infty - \|\mathbf{A}\|_\infty|] \leq \mathbb{E}[\|\mathbf{E}|_\infty]$ and $\mathbb{E}[|\|\mathbf{Y}\|_\infty - \mathbb{E}[\|\mathbf{E}|_\infty] - \|\mathbf{A}\|_\infty|] \leq 2\,\mathbb{E}[\|\mathbf{E}|_\infty]$. By Lemma 9, the expectation of the operator norm of $\mathbf{E}$ is less or equal to $\sqrt{p} + \sqrt{q}$. This concludes the first part of the proof.

Let us now lower bound the risk of $\|\mathbf{Y}\|_\infty$ and $\|\mathbf{Y}\|_\infty - \mathbb{E}\left[\|\mathbf{E}\|_\infty\right]$. Choose $\mathbf{A}$ to be the null matrix. The risk of $\|\mathbf{Y}\|_\infty$ is then $\mathbb{E}[\|\mathbf{E}\|_\infty]$. Consider the size $q$ vector $e_1 = (1, 0, \ldots, 0)$. Then,

$$\mathbb{E}[|\|\mathbf{Y}\|_\infty - \|\mathbf{A}\|_\infty|] \geq \mathbb{E}[|\mathbf{E}e_1|_2] \ .$$

Since $|\mathbf{E}e_1|_2$ is the norm of a standard Gaussian vector of dimension $p$, the expectation $\mathbb{E}[|\mathbf{E}e_1|_2]$ is that of the norm of a standard Gaussian vector. Relying on Deviation inequalities for $\chi^2$ random variables, we deduce that

$$\mathbb{E}[|\mathbf{E}e_1|_2] \geq \sqrt{\frac{p}{3}}\,\mathbb{P}[|\mathbf{E}e_1|_2^2 \geq p/3] \geq \frac{p}{4}\left(1 - e^{-p/32}\right) \geq cp\ ,$$

for a constant $c > 0$.

To bound the risk of $\|\mathbf{Y}\|_\infty - \mathbb{E}\left[\|\mathbf{E}\|_\infty\right]$, we choose a matrix $\mathbf{A}$ which is zero everywhere, except at its upper left entry which equals $a \geq 0$. Recall that the operator norm satisfies $\|\mathbf{A}+\mathbf{E}\|_\infty = \sup_{|u|_2=1,\ |v|_2=1} u^T(\mathbf{A}+\mathbf{E})v$. Denote the size $p-1$ and $q-1$ vectors defined by $w_i = \mathbf{E}_{i+1,1}$ and $w'_i = \mathbf{E}_{1,i+1}$. Besides, we denote $\mathbf{E}'$ the principal submatrix of $\mathbf{E}$ where we have removed the first row and the first column. Denote the size $p$ vector $e'_1 = (1,0,\ldots,0)$. Decomposing $u = \alpha e'_1 + u_1$ and $v = \beta e_1 + v_1$ where $u_1$ (resp. $v_1$ is orthogonal) to $v_1$, we obtain

$$\begin{aligned}
\|\mathbf{A}+\mathbf{E}\|_\infty &\leq \sup_{\alpha,\beta\in[-1,1]} |\alpha\beta\,(a+\mathbf{E}_{1,1}) + \alpha\sqrt{1-\beta^2}|w|_2 + \beta\sqrt{1-\alpha^2}|w'|_2 + \sqrt{1-\alpha^2}\sqrt{1-\beta^2}\|\mathbf{E}'\|_\infty| \\
&\leq \sup_{\alpha,\beta\in[0,1]} \left|\frac{\alpha^2+\beta^2}{2}|a+\mathbf{E}_{1,1}| + \sqrt{1-\beta^2}|w|_2 + \sqrt{1-\alpha^2}|w'|_2 + \left(1-\frac{\alpha^2+\beta^2}{2}\right)\|\mathbf{E}'\|_\infty\right| \\
&\leq \sup_{\alpha,\beta\in[0,1]} \left|\alpha^2|a+\mathbf{E}_{1,1}| + 2\sqrt{1-\alpha^2}(|w|_2 \vee |w'|_2) + \left(1-\alpha^2\right)\|\mathbf{E}'\|_\infty\right|\ ,
\end{aligned}$$

Write $\theta = |a+\mathbf{E}_{1,1}| - \|\mathbf{E}'\|_\infty$ and $\gamma = (|w|_2 \vee |w'|_2)$. Providing that $\theta > \gamma$, we may compute the above supremum by derivation, which leads us to $\|\mathbf{A}+\mathbf{E}\|_\infty \leq \theta + \frac{\gamma^2}{\theta}$. In other words, we have proved

$$\|\mathbf{A}+\mathbf{E}\|_\infty \leq a + |\mathbf{E}_{1,1}| + \frac{|w|_2^2 \vee |w'|_2^2}{|a+\mathbf{E}_{1,1}| - \|\mathbf{E}'\|_\infty} \text{ provided that } a > |\mathbf{E}|_{1,1} + \|\mathbf{E}'\|_\infty + (|w|_2 \vee |w'|_2)\ .$$

Since $a = \|\mathbf{A}\|_\infty$, this enforces

$$|\|\mathbf{Y}\|_\infty - \mathbb{E}[\|\mathbf{E}\|_\infty] - \|\mathbf{A}\|_\infty| \geq \mathbb{E}[\|\mathbf{E}\|_\infty] - |\mathbf{E}_{1,1}| - \frac{|w|_2^2 \vee |w'|_2^2}{a - |\mathbf{E}_{1,1}| - \|\mathbf{E}'\|_\infty}\ ,$$

provided that $a > |\mathbf{E}|_{1,1} + \|\mathbf{E}'\|_\infty + (|w|_2 \vee |w'|_2)$. Taking $a$ large enough, the condition holds with probability higher than $3/4$ and the ratio $\frac{|w|_2^2 \vee |w'|_2^2}{a - |\mathbf{E}_{1,1}| - \|\mathbf{E}'\|_\infty}$ is smaller or equal to one. Besides, with probability higher than $3/4$, $|\mathbf{E}_{1,1}|$ is smaller or equal to 1.2. By an union bound, we conclude that the loss of $\|\mathbf{Y}\|_\infty - \mathbb{E}[\|\mathbf{E}\|_\infty]$ is higher than $\mathbb{E}[\|\mathbf{E}\|_\infty] - 2.2$ with probability higher than $1/2$. Since we have proved above that $\mathbb{E}[\|\mathbf{E}\|_\infty] \geq c\sqrt{p}$, the result follows. □

*Proof of Proposition 4.* It follows from the triangular inequality that

$$\|\mathbf{Y}^T\mathbf{Y}\|_\infty - p - \|\mathbf{A}^T\mathbf{A}\|_\infty \leq 2\|\mathbf{A}^T\mathbf{E}\|_\infty + \|\mathbf{E}^T\mathbf{E}\|_\infty - p\ .$$

Conversely, we have

$$\begin{aligned}
\|\mathbf{Y}^T\mathbf{Y}\|_\infty &= \sup_{u,|u|_2=1} \left(u^T\mathbf{A}^T\mathbf{A}u + 2u^T\mathbf{A}^T\mathbf{E}u + u^T\mathbf{E}^T\mathbf{E}u\right) \geq \sup_{u,|u|_2=1} \left(u^T\mathbf{A}^T\mathbf{A}u\right) - 2\|\mathbf{A}^T\mathbf{E}\|_\infty + \lambda_q(\mathbf{E}^T\mathbf{E}) \\
&\geq \|\mathbf{A}^T\mathbf{A}\|_\infty - 2\|\mathbf{A}^T\mathbf{E}\|_\infty + \lambda_q(\mathbf{E}^T\mathbf{E})\ .
\end{aligned}$$

This allows us to bound the error

$$\begin{aligned}
\left|\|\mathbf{Y}^T\mathbf{Y}\|_\infty - p - \|\mathbf{A}^T\mathbf{A}\|_\infty\right| &\leq 2\|\mathbf{A}^T\mathbf{E}\|_\infty + [\lambda_1(\mathbf{E}^T\mathbf{E}) - p] \vee [\lambda_q(\mathbf{E}^T\mathbf{E}) - p] \\
&\leq 2\|\mathbf{A}^T\mathbf{E}\|_\infty + \|\mathbf{E}^T\mathbf{E} - p\mathbf{I}\|_\infty\ . \qquad (2.17)
\end{aligned}$$

Then, the first risk bound is a straightforward consequence of the two following lemmas whose proof is given below.

**Lemma 3.** *For all matrices* $\mathbf{A}$*, we have* $\mathbb{E}\left[\|\mathbf{A}^T\mathbf{E}\|_\infty\right] \leq 2\|\mathbf{A}\|_\infty\sqrt{q}$.

**Lemma 4.** $\mathbb{E}\left[\|\mathbf{E}^T\mathbf{E} - p\mathbf{I}\|_\infty\right] \leq 3\sqrt{pq} + 8e^{-p/4} + \sqrt{64\pi p}$.

As in the proof of Proposition 2, we deal differently with small and large values of $\|\mathbf{A}\|_\infty$. Since $\sqrt{x} - \sqrt{y} \leq \sqrt{|x-y|}$, we obtain by Cauchy-Schwarz inequality that

$$\begin{aligned}
\mathbb{E}\left[\left|(\|\mathbf{Y}^T\mathbf{Y}\|_\infty - p)_+^{1/2} - \|\mathbf{A}\|_\infty\right|\right] &= \mathbb{E}\left[\left|\sqrt{(\|\mathbf{Y}^T\mathbf{Y}\|_\infty - p)_+} - \sqrt{\|\mathbf{A}\|_\infty^2}\right|\right] \\
&\leq \mathbb{E}\left[\sqrt{\left|(\|\mathbf{Y}^T\mathbf{Y}\|_\infty - p)_+ - \|\mathbf{A}^T\mathbf{A}\|_\infty\right|}\right] \\
&\leq \left(\mathbb{E}\left[\left|\|\mathbf{Y}^T\mathbf{Y}\|_\infty - p - \|\mathbf{A}^T\mathbf{A}\|_\infty\right|\right]\right)^{\frac{1}{2}} \\
&\leq \left(3\sqrt{pq} + 4\|\mathbf{A}\|_\infty\sqrt{q} + c'(\sqrt{p}+1)\right)^{\frac{1}{2}}
\end{aligned}$$

If we assume that $\|\mathbf{A}\|_\infty \leq (pq)^{1/4}$, this leads us to

$$\mathbb{E}\left[\left|(\|\mathbf{Y}^T\mathbf{Y}\|_\infty - p)_+^{1/2} - \|\mathbf{A}\|_\infty\right|\right] \leq c(pq)^{1/4} \ .$$

Now assume that $\|\mathbf{A}\|_\infty \geq (pq)^{1/4}$, which is equivalent to $\|\mathbf{A}^T\mathbf{A}\|_\infty \geq \sqrt{pq}$. Since $|x-y| \leq |x^2-y^2|/|y|$, it follows that

$$\begin{aligned}
\mathbb{E}\left[\left|(\|\mathbf{Y}^T\mathbf{Y}\|_\infty - p)_+^{1/2} - \|\mathbf{A}\|_\infty\right|\right] &\leq \frac{1}{\|\mathbf{A}\|_\infty}\mathbb{E}\left[\left|(\|\mathbf{Y}^T\mathbf{Y}\|_\infty - p)_+ - \|\mathbf{A}\|_\infty^2\right|\right] \\
&\leq 3(pq)^{1/4} + 4\sqrt{q} + c'(p/q)^{1/4} \leq c(pq)^{1/4} \ ,
\end{aligned}$$

which concludes the proof. □

*Proof of lemma 3.* Let $\mathbf{A} = \mathbf{U}^T\mathbf{D}\mathbf{V}$ denote a singular value decomposition of $\mathbf{A}$. Since the distribution of $\mathbf{E}$ is invariant by left and right orthogonal transformation, it follows that $\|\mathbf{A}\mathbf{E}\|_\infty$ follows the same distribution as $\|\mathbf{V}\mathbf{D}^T\mathbf{E}\|_\infty = \|\mathbf{D}^T\mathbf{E}\|_\infty$. We shall bound the expectation of this last random variable. Since $\mathbf{D}^T$ is a $q \times p$ diagonal matrix, $\mathbf{D}^T\mathbf{E}$ does not depend on the entries $\mathbf{E}_{ij}$ with $i \geq q$. Write $\overline{\mathbf{D}}$ and $\overline{\mathbf{E}}$ for the restriction of $\mathbf{D}$ and $\mathbf{E}$ to their $q$ first rows. We obtain

$$\begin{aligned}
\mathbb{E}\left[\|\mathbf{A}^T\mathbf{E}\|_\infty\right] &= \mathbb{E}\left[\|\overline{\mathbf{D}}\overline{\mathbf{E}}\|_\infty\right] \leq \|\overline{\mathbf{D}}\|_\infty\,\mathbb{E}\left[\|\overline{\mathbf{E}}\|_\infty\right] \\
&\leq 2\|\mathbf{A}\|_\infty\sqrt{q} \ ,
\end{aligned}$$

by Lemma 9 and since $\|\overline{\mathbf{D}}\|_\infty = \|\mathbf{D}\|_\infty = \|\mathbf{A}\|_\infty$.

□

*Proof of lemma 4.* Observe that $\|\mathbf{E}^T\mathbf{E} - p\mathbf{I}\|_\infty = \max(\sigma_1^2(\mathbf{E}) - p, p - \sigma_q^2(\mathbf{E}))$. We deduce from Lemma 10 (taken from [28]) that, for any $t > 0$, with probability higher than $1 - 2e^{-t}$ we have

$$\sigma_1(\mathbf{E}) \leq \sqrt{p} + \sqrt{q} + \sqrt{2t} \ ; \sigma_q(\mathbf{E}) \leq \sqrt{p} - \sqrt{q} - \sqrt{2t} \ .$$

Hence, with probability higher than $1 - 2e^{-t}$, we have $\|\mathbf{E}^T\mathbf{E} - p\mathbf{I}\|_\infty \leq q + 2\sqrt{pq} + 2t + 4\sqrt{2pt}$, which implies that

$$\mathbb{P}\left[\|\mathbf{E}^T\mathbf{E} - p\mathbf{I}\|_\infty - q - 2\sqrt{pq} \geq x\right] \leq 2\exp\left[-\frac{1}{4}\left(x \wedge \frac{x^2}{16p}\right)\right] \ .$$

for any $x > 0$. Integrating this inequality we conclude that

$$\mathbb{E}\left[\|\mathbf{E}^T\mathbf{E} - p\mathbf{I}\|_\infty\right] \leq 3\sqrt{pq} + \int_{\mathbb{R}} e^{-x^2/(64p)} + 2\int_{16p}^{\infty} e^{-x/4} = 3\sqrt{pq} + 8e^{-4p} + \sqrt{64\pi p} \ .$$

□

### 2.7.2 Minimax lower bound

*Proof of Theorem 1.* We follow the general Le Cam's approach of fuzzy hypotheses as may be found in [122, Sec.2.7.4]. Given a matrix $\mathbf{A}$, we denote in the section of $\mathbb{P}_{\mathbf{A}}$ for the distribution of $\mathbf{Y}$. Consider a prior distribution $\mu$ on $\mathbf{A}$, such that, for some $t > 0$, $\|\mathbf{A}\|_2 \geq t$, $\mu$ almost surely. Let the set $\Theta$ stand for the union of the support of $\mu$ and the null matrix 0. Denoting $\mathbf{P} = \int_{\mathbf{A}} \mathbb{P}_{\mathbf{A}}\, \mu(d\mathbf{A})$ for the integrated distribution, it follows from Theorem 2.15 in [122] that

$$\inf_{\widehat{T}} \sup_{\mathbf{A}\in\Theta} \mathbb{P}_{\mathbf{A}}[|\widehat{T} - \|\mathbf{A}\|_2| \geq s/2] \geq \frac{1 - \sqrt{\chi^2(\mathbb{P}_0, \mathbf{P})/2}}{2} \ .$$

If we further assume that $\Theta$ only contains matrices of rank less or equal to one, this implies that

$$\inf_{\widehat{T}} \sup_{\mathbf{A},\ \mathrm{rank}(\mathbf{A})\leq 1} \mathbb{E}_{\mathbf{A}}[|\widehat{T} - \|\mathbf{A}\|_2|] \geq \frac{s}{4}\left[1 - \sqrt{\chi^2(\mathbb{P}_0, \mathbf{P})/2}\right] \ . \tag{2.18}$$

It remains to choose $\mu$ and bound the corresponding $\chi^2$ distance.

Let $s > 0$ be a positive quantity that will be fixed later. Let $\nu_1$ be the uniform distribution over the vectors of the form $p^{-1/2}(\eta_1, \ldots, \eta_p)$ where the $\eta_i$'s belong to $\{-1, 1\}$. Let $\nu_2$ be corresponding distribution where $q$ is replaced by $p$. Finally, let $\mu$ be the distribution of $\mathbf{A} = su^T v$ where $u$ and $v$ are sampled independently from $\nu_1$ and $\nu_2$. By construction, $\mu$ almost surely, $\mathbf{A}$ is a rank 1 matrix and $\|\mathbf{A}\|_2 = s$.

The main part of the proof amounts to upper bounding the $\chi^2$ discrepancy between $\mathbb{P}_0$ and $\mathbf{P}$. Writing $L = d\mathbf{P}/\mathbb{P}_0$ the likelihood ratio between $\mathbf{P}$ and $\mathbb{P}_0$, we have by definition [122] that

$$\chi^2(\mathbb{P}_0, \mathbf{P}) = \mathbb{E}_0[(L-1)^2] = \mathbb{E}_0[L^2] - 1 \ . \tag{2.19}$$

**Lemma 5.** *Taking $s = (pq/4)^{1/4}$, we have $\mathbb{E}_0\left[L^2\right] \leq \frac{5}{3}$.*

Equipped with this choice of $s$, we have $\chi^2(\mathbb{P}_0, \mathbf{P}) \leq 2/3$ and we conclude thanks to (2.18) that

$$\inf_{\widehat{T}} \sup_{\mathbf{A},\ \mathrm{rank}(\mathbf{A})\leq 1} \mathbb{E}_{\mathbf{A}}[|\widehat{T} - \|\mathbf{A}\|_2|] \geq \frac{(pq/4)^{1/4}}{4}(1 - \sqrt{1/3}) \geq \frac{(pq)^{1/4}}{20} \ .$$

□

*Proof of Lemma 5.* To alleviate the notation, we write $\|.\|$ (resp. $\langle . \rangle$) for the Frobenius norm $\|.\|_2$ (resp. inner product) in this proof. First, we work out the likelihood ratio $L$. Since the density of $\mathbb{P}_{\mathbf{A}}$ with respect to the Lebesgue measure is $(2\pi)^{-(pq)/2} e^{-\|\mathbf{Y}-\mathbf{A}\|_2^2/2}$, it follows that

$$\begin{aligned} L &= \int \exp\left[-\frac{1}{2}\|\mathbf{Y} - \mathbf{A}\|^2 + \|\mathbf{Y}\|^2\right] \mu(d\mathbf{A}) \\ &= \int \exp\left[-\frac{1}{2}\|\mathbf{A}\|^2 + \langle \mathbf{Y}, \mathbf{A}\rangle\right] \mu(d\mathbf{A}) \ . \end{aligned}$$

As a consequence, the second moment of the likelihood writes as

$$\begin{aligned} \mathbb{E}_0[L^2] &= \mathbb{E}_0\left[\int\int \exp\left[-\frac{1}{2}\|\mathbf{A}_1\|^2 - \frac{1}{2}\|\mathbf{A}_2\|^2 + \langle \mathbf{Y}, (\mathbf{A}_1 + \mathbf{A}_2)\rangle\right]\right] \\ &= \int\int \mathbb{E}_0\left[\exp\left[-\frac{1}{2}\|\mathbf{A}_1\|^2 - \frac{1}{2}\|\mathbf{A}_2\|^2 + \langle \mathbf{Y}, (\mathbf{A}_1 + \mathbf{A}_2)\rangle\right]\right] \mu(d\mathbf{A}_1)\mu(d\mathbf{A}_2) \\ &= \int\int \exp\left[-\frac{1}{2}\|\mathbf{A}_1\|^2 - \frac{1}{2}\|\mathbf{A}_2\|^2 + \|\mathbf{A}_1 + \mathbf{A}_2\|^2/2\right] \mu(d\mathbf{A}_1)\mu(d\mathbf{A}_2) \\ &= \int\int \exp\left[\langle \mathbf{A}_1, \mathbf{A}_2\rangle\right] \mu(d\mathbf{A}_1)\mu(d\mathbf{A}_2) \ , \end{aligned}$$

where we used Fubini's Theorem in the second line and the Laplace transform of the normal random variable in the third line. In view of the definition of $\mu$, this integral further decomposes as

$$\begin{aligned}\mathbb{E}_0[L^2] &= \int\int \exp\left[s^2 \operatorname{tr}[v_1^T u_1 u_2^T v_2^T]\right] \nu_1(du_1)\nu_1(du_2)\nu_2(dv_1)\nu_1(dv_2) \\ &= \int\int \exp\left[s^2 (v_1^T v_2)(u_1^T u_2)\right] \nu_1(du_1)\nu_1(du_2)\nu_2(dv_1)\nu_1(dv_2) \ .\end{aligned}$$

Since $Z_1 = p u_1^T u_2$ and $Z_2 = q v_1^T v_2$ are respectively distributed as sums of $p$ and $q$ independent Rademacher random variables, arrive at

$$\mathbb{E}_0[L^2] = \mathbb{E}\left[\exp\left(\frac{s^2 Z_1 Z_2}{pq}\right)\right]$$

For $a \in \mathbb{R}$ and $X$ a Rademacher random variable, we have $\mathbb{E}[e^{aX}] = \cosh(ax) \leq e^{a^2x^2/2}$ (compare the power series). We obtain by integration with respect to $Z_1$,

$$\mathbb{E}_0[L^2] = \mathbb{E}\left[\cosh\left(\frac{s^2 Z_2}{pq}\right)^p\right] \leq \mathbb{E}\left[\exp\left(\frac{s^4 Z_2^2}{2pq^2}\right)\right] = \mathbb{E}\left[\exp\left(\frac{1}{8q} Z_2^2\right)\right] \ ,$$

since we fixed $s = (pq/4)^{1/4}$. Since $Z_2$ is a sum of Rademacher random variables, we can apply Hoeffding's inequality [15], which leads us to $\mathbb{P}(|Z_2| \geq u) \leq 2e^{\frac{-u^2}{2q}}$. As a consequence,

$$\begin{aligned}\mathbb{E}_0[L^2] &\leq \mathbb{E}\left[\exp\left(\frac{1}{8q} Z_2^2\right)\right] = \int_0^\infty \mathbb{P}\left[\exp\left(\frac{Z_2^2}{8q}\right) \geq t\right] dt \\ &\leq 1 + 2\int_1^\infty \mathbb{P}\left[\exp\left(\frac{Z_2^2}{8q}\right) \geq t\right] dt \\ &\leq 1 + 2\int_1^\infty \frac{1}{t^4} dt = 5/3 \ .\end{aligned}$$

The result follows. □

### 2.7.3 Proof for general even norms $\|\mathbf{A}\|_{2k}$

We start with a lemma summarizing important properties of the Hermite polynomials (2.8).

**Lemma 6.** *[18, Lemma 3] Let $Z \sim \mathcal{N}(x, 1)$. Then, $\operatorname{Var}(H_r(Z)) \leq e^{x^2} r^r$. If $x^2 \geq r$, we have also $\operatorname{Var}(H_r(Z)) \leq (2x^2)^r$. For $Z \sim \mathcal{N}(0, 1)$, we have*

$$\mathbb{E}[H_r^2(Z)] = r! \ , \ \textit{and} \ \mathbb{E}[H_r(Z)H_l(Z)] = 0 \ , \ \textit{for} \ r \neq l \ .$$

*Proof of Proposition 7.* Let us upper bound $\operatorname{Var}(U_k) = \mathbb{E}\left[\left(U_k - \sum_i \sigma_i^{2k}(\mathbf{A})\right)^2\right]$. Recall that we assume without loss of generality that $\mathbf{A}_{rs} = \sigma_r(\mathbf{A})\mathbf{1}_{r=s}$. Given two sequences $i, j$ we write $W_{ij} = \prod_{r,s} H_{N_{rs}(i,j)}(\mathbf{Y}_{rs})$. Relying on the Hermite Decomposition of $U_k$, we observe that

$$\operatorname{Var}(U_k) = \sum_{i,j}\sum_{i',j'} \mathbb{E}\left[\left[W_{ij} - \mathbb{E}(W_{ij})\right]\left[W_{i'j'} - \mathbb{E}\left(W_{i'j'}\right)\right]\right] =: T_{iji'j'} \ .$$

In the remainder of this proof, we first use some simple combinatorial arguments to count the number of such non zero $T_{iji'j'}$. The expectation $\mathbb{E}(W_{ij})$ is non zero if and only if $N_{rs}(i,j) = 0$ for all $r \neq s$. This is possible only if $i_1 = i_2 = \ldots = i_k = j_1 = \ldots = j_k$. As a consequence, only the terms $H_{2k}(Y_{rr})$ have a non-zero expectation.

First, we bound the cross-covariance terms $T_{iji'j'}$. This bound is divided into two cases:
**Case 1**: $\mathbb{E}(W_{ij}) \neq 0$. Hence, this expectation is equal to $\sigma_r^{2k}(\mathbf{A})$ for some $1 \leq r \leq q$. Then, $T_{iji'j'}$ is non zero only if both the support of $i'$ and of $j'$ contain $r$. If $i' \neq i$ or if $j' \neq j$, this implies that

$(i',j')$ is not constant and $\mathbb{E}[W_{i'j'}]=0$. Besides, there are $s_1\neq s_2$ such that $N_{s_1s_2}(i'j')>0$. Since $\mathbf{A}_{s_1s_2}=0$ and $N_{s_1s_2}(ij)>0$ this implies that $T_{iji'j'}=0$ by independence of the noise components. If $(i'=i)$ and $(j'=j)$, then Lemma 6 ensures that $\mathbb{E}[T_{iji'j'}]=\operatorname{Var}H_{2k}(\mathbf{Y}_{rr})\leq c_{\exp}[k]\left[\sigma_r^{4k-2}\vee 1\right]$.
**Case 2**: $\mathbb{E}\left(W_{ij}\right)=\mathbb{E}\left(W_{i'j'}\right)=0$. In the sequel, the support of $(i,j)$ is defined as $\{(r,s):\ N_{rs}(ij)>0\}$. If the supports of $(i,j)$ and $(i',j')$ differ outsides the diagonal $(r,r)$, then $T_{iji'j'}=0$. Besides, from Lemma 6 we need $N_{rs}(ij)+N_{rs}(i'j')\equiv 0[2]$ for all $r\neq s$ otherwise $E[T_{iji'j'}]=0$. As explained in the proof of Proposition 5 below, we always have $\sum_{s=1}^q N_{rs}(ij)\equiv 0[2]$ and $\sum_{s=1}^p N_{sr}(ij)\equiv 0[2]$ because each indice in the sequence $i$ (or $j$) appears twice in the monomial $W_{ij}$. Hence, $E[T_{iji'j'}]\neq 0$ implies that, for all $r=1,\ldots,q$, $N_{rr}(ij)+N_{rr}(i'j')\equiv 0[2]$. Furthermore, if $N_{rr}(i'j')>0$, we need that either $N_{rs}(ij)>0$ or $N_{sr}(ij)>0$ for some $s$ to have $T_{iji'j'}\neq 0$ otherwise this contradicts $\mathbb{E}[W_{i'j'}]=0$ or the fact that the support of $(i,j)$ and $(i',j')$ are matching outside the diagonal. To summarize, we have proved that $T_{iji'j'}\neq 0$ only if:

(a) $N_{rs}(ij)+N_{rs}(i'j')\equiv 0[2]$, for all $r,s$.

(b) For any $r\neq s$, $N_{rs}(ij)=0\Leftrightarrow N_{rs}(i'j')=0$.

(c) The supports of $i'$ and $j'$ are included in the union of the supports of $i$ and $j$.

Then, Lemma 6 yields

$$\begin{aligned}
T_{iji'j'} &= \prod_{rs}\mathbb{E}\left[H_{N_{rs}(ij)}(\mathbf{Y}_{rs})H_{N_{rs}(ij)}(\mathbf{Y}_{rs})\right]\\
&\leq \prod_{r,s}c_{\exp}[N_{rs}(ij)N_{rs}(i'j')]\prod_{r=1}^{q}\left[\sigma_r^{N_{rr}(ij)+N_{rr}(i'j')}\vee 1\right]\\
&\leq c_{\exp}(k)\prod_{r=1}^{q}\left[(\sigma_r\vee 1)^{N_{rr}(ij)+N_{rr}(i'j')}\right]\ . \qquad (2.20)
\end{aligned}$$

Consider two sequences $\alpha^{(1)}=(\alpha_1^{(1)},\ldots,\alpha_q^{(1)})$ and $\alpha^{(2)}=(\alpha_1^{(2)},\ldots,\alpha_q^{(2)})$ of nonnegative integers such that $\sum_r\alpha_r^{(1)}\geq\sum_r\alpha_r^{(2)}$ . Let us count the numbers of $(i,j)$ and $(i',j')$ such that $N_{rr}(ij)=\alpha_r^{(1)}$, $N_{rr}(i'j')=\alpha_r^{(2)}$ and $T_{iji'j'}\neq 0$. From Case 1 above, we deduce that, if $\alpha_r^{(1)}=2k$ for some $r$, then we necessary have $\alpha_r^{(1)}=\alpha_r^{(2)}=2k$. Thus, there is only one such possibility. Now, assume that $\max_r\alpha_r^{(1)}<2k$. If $\alpha_r^{(1)}>0$, then both $i$ and $j$ must contain at least $\lceil\alpha_r^{(1)}/2\rceil$ times the indice $r$. We have $\sum_{s\neq r}N_{rs}(ij)+\sum_{s\neq r}N_{sr}(ij)\geq 2$ since the sequence $(i_li_{l+1})(i_{l+1}j_l)$ is not constant in $(r,r)$. As a consequence, either $i$ or $j$ must contain at least $\lceil(\alpha_r^{(1)}+1)/2\rceil$ occurrences of $r$. If $\alpha_r^{(1)}=0$ and $\alpha_r^{(2)}>0$, then $r$ occurs at least once in $i$ or $j$ by Property (c) above. Write $a:=\{r,\ \alpha_r^{(2)}>0 \text{ and } \alpha_r^{(1)}=0\}$. Since $p\geq q$, we conclude that there are less than

$$c_{\exp}(k)p^{k-\sum_r\lceil\alpha_r^{(1)}/2\rceil}q^{k-\sum_{r:\alpha_r^{(1)}>0}\lceil(\alpha_r^{(1)}+1)/2\rceil-a}$$

such sequences $(i,j)$ and given $(i,j)$ there are less than $c_{\exp}(k)$ possible sequences $(i',j')$. Writing $\overline{\sigma}_i(\mathbf{A})$ for $\sigma_i(\mathbf{A})\vee 1$, we obtain

$$\operatorname{Var}\left(U_k\right)\ \leq\ c_{\exp}(k)\left[\|\overline{\sigma}(\mathbf{A})\|_{4k-2}^{4k-2}+\sum_{\alpha^{(1)},\alpha^{(2)}}p^{k-\sum_r\lceil\frac{\alpha_r^{(1)}}{2}\rceil}q^{k-\sum_{r:\alpha_r^{(1)}>0}\lceil\frac{\alpha_r^{(1)}+1}{2}\rceil-a}\prod_{r=1}^{q}\overline{\sigma}_r^{\alpha_r^{(1)}+\alpha_r^{(2)}}(\mathbf{A})\right]\ ,$$

where the sum runs over sequences $\alpha^{(1)}$, $\alpha^{(2)}$ of integers such that $\max_r(\alpha_r^{(1)},\alpha_r^{(2)})\leq 2k-2$, $\alpha_r^{(1)}+\alpha_r^{(2)}\equiv 0[2]$, and $\sum_r\alpha_r^{(1)}\geq\sum_r\alpha_r^{(2)}$. Now fix a sequence $\alpha=(\alpha_1,\ldots,\alpha_q)$ of even nonnegative integers. Let us consider sequences $\alpha^{(1)}$ and $\alpha^{(2)}$ satisfying the above properties such that $\alpha^{(1)}+$

$\alpha^{(2)} = \alpha$ and let us maximize $p^{k-\sum_r \lceil \frac{\alpha_r^{(1)}}{2} \rceil} q^{k-\sum_r \lceil \frac{\alpha_r^{(1)}+1}{2} \rceil - a}$. Writing $d = \sum_r \alpha_r$ and $l = |\{r : \ \alpha_r \neq 0\}|$. First, we work out the exponent in $q$.

$$\sum_{r:\alpha_r^{(1)}>0} \lceil \frac{\alpha_r^{(1)}+1}{2} \rceil + a = \sum_{r:\alpha_r^{(1)}>0} (\lfloor \frac{\alpha_r^{(1)}}{2} \rfloor + 1) + a = \sum_{r:\alpha_r^{(1)}>0} \lfloor \frac{\alpha_r^{(1)}}{2} \rfloor + l$$

Since $\sum_r \lfloor \frac{\alpha_r^{(1)}}{2} \rfloor + \lceil \frac{\alpha_r^{(1)}}{2} \rceil = \sum_r \alpha_r^{(1)} \geq d/2$, we derive that

$$p^{k-\sum_r \lceil \frac{\alpha_r^{(1)}}{2} \rceil} q^{k-\sum_{r:\alpha_r^{(1)}>0} \lceil \frac{\alpha_r^{(1)}+1}{2} \rceil - a} \leq (pq)^k q^{-d/2-l} (q/p)^{\sum_r \lceil \frac{\alpha_r^{(1)}}{2} \rceil} \leq (pq)^k q^{-d/2-l} (q/p)^{\lceil d/4 \rceil} \ ,$$

since $p \geq q$. This leads us to

$$\mathrm{Var}\,(U_k) \leq c_{\exp}(k) \left[ \|\overline{\sigma}(\mathbf{A})\|_{4k-2}^{4k-2} + \sum_{\alpha} p^{k-\lceil \frac{d}{4} \rceil} q^{k-\lfloor \frac{d}{4} \rfloor - l} \prod_{r=1}^{q} \overline{\sigma}_r^{\alpha_r}(\mathbf{A}) \right] \ , \tag{2.21}$$

where the sum runs over sequences $\alpha$ of nonnegative integers such that $\sum_r \alpha_r \leq 4k-2$. Now consider a decreasing sequence $\beta_1 \geq \beta_2 \ldots \geq \beta_q$ of non-negative integers. We group all sequences $\alpha$ that are permutations of $2\beta$. For all such $\alpha$, the power of $p$ and $q$ in (2.21) is unchanged. We claim that the sum over such $\alpha$ of $\prod_{r=1}^q \overline{\sigma}_r^{\alpha_r}(\mathbf{A})$ is at most $\sum_{s \in \{1,\ldots,q\}^r} \prod_{r=1}^q \overline{\sigma}(\mathbf{A})_{s_r}^{\beta_r} = \prod_{r=1}^{|\{r:\ \beta_r \neq 0\}|} \|\overline{\sigma}(\mathbf{A})\|_{2\beta_r}^{2\beta_r}$. This leads us to

$$\begin{aligned}
\mathrm{Var}\,(U_k) &\leq c_{\exp}(k) \left[ \|\overline{\sigma}(\mathbf{A})\|_{4k-2}^{4k-2} + \sum_{\beta} p^{k-\lceil \frac{\sum_r \beta_r}{2} \rceil} q^{k-\lfloor \frac{\sum_r \beta_r}{2} \rfloor - |\{r:\ \beta_r \neq 0\}|} \prod_{r=1}^{|\{r:\ \beta_r \neq 0\}|} \|\overline{\sigma}(\mathbf{A})\|_{2\beta_r}^{2\beta_r} \right] \\
&\leq c_{\exp}(k) \left[ \|\overline{\sigma}(\mathbf{A})\|_{4k-2}^{4k-2} + (pq)^k + \sum_{l=1}^{2k-1} \sum_{\beta: \sum_r \beta_r = l} p^{k-\lceil \frac{l}{2} \rceil} q^{k-\lfloor \frac{l}{2} \rfloor - |\{r:\ \beta_r \neq 0\}|} \prod_{r=1}^{|\{r:\ \beta_r \neq 0\}|} \|\overline{\sigma}(\mathbf{A})\|_{2\beta_r}^{2\beta_r} \right] \\
&\leq c_{\exp}(k) \left[ \|\overline{\sigma}(\mathbf{A})\|_{4k-2}^{4k-2} + (pq)^k + \sum_{l=1}^{2k-2} p^{k-\lceil \frac{l}{2} \rceil} q^{k-\lfloor \frac{l}{2} \rfloor - 1} \|\overline{\sigma}(\mathbf{A})\|_{2l}^{2l} \right] \ ,
\end{aligned}$$

where we applied Holder inequality multiple times in the last line and we used that the number of sequences $\beta$ is less than $c_{\exp}(k)$. Applying again Holder inequality we have $\|\overline{\sigma}(\mathbf{A})\|_{2l}^{2l} \leq [\|\overline{\sigma}(\mathbf{A})\|_{4k-4}^{2l} q^{1-l/(2k-2)}] \wedge [\|\overline{\sigma}(\mathbf{A})\|_{4k-2}^{2l} q^{1-l/(2k-1)}]$ so that

$$\begin{aligned}
\mathrm{Var}\,(U_k) &\leq c_{\exp}(k) \left[ \sum_{s=0}^{k-1} p^{k-s} q^{k-s\frac{k}{k-1}} \|\overline{\sigma}(\mathbf{A})\|_{4k-4}^{4s} + p^{k-s-1} q^{k-s-\frac{2s+1}{2k-1}} \|\overline{\sigma}(\mathbf{A})\|_{4k-2}^{4s+2} \right] \\
&\leq c_{\exp}(k) \left[ (pq)^k + p^{k-1} q^{k-\frac{1}{2k-1}} \|\overline{\sigma}(\mathbf{A})\|_{4k-2}^2 + p\|\overline{\sigma}(\mathbf{A})\|_{4k-4}^{4k-4} + \|\overline{\sigma}(\mathbf{A})\|_{4k-2}^{4k-2} \right] \\
&\leq c_{\exp}(k) \left[ (pq)^k + [(pq)^k]^{1-\frac{1}{2k-1}} \left[ \|\overline{\sigma}(\mathbf{A})\|_{4k-2}^{4k-2} \right]^{\frac{1}{2k-1}} + p\|\overline{\sigma}(\mathbf{A})\|_{4k-4}^{4k-4} + \|\overline{\sigma}(\mathbf{A})\|_{4k-2}^{4k-2} \right] \\
&\leq c_{\exp}(k) \left[ (pq)^k + p\|\overline{\sigma}(\mathbf{A})\|_{4k-4}^{4k-4} + \|\overline{\sigma}(\mathbf{A})\|_{4k-2}^{4k-2} \right] \ ,
\end{aligned}$$

where we used that the sequences are monotone with respect to $s$ in the second line, that $p \geq q$ in the third line, and that $x^a y^{1-a} \leq x \vee y$ for any $a \in [0,1]$. This concludes the proof since $\|\overline{\sigma}(\mathbf{A})\|_{4k-4}^{4k-4} \leq \|\mathbf{A}\|_{4k-4}^{4k-4} + q$.

□

*Proof of Proposition 5.* The result is a consequence of the following lemma.

**Lemma 7.** *There exist coefficients $\alpha_s$ such that*

$$V_k = \mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})^k] + \alpha_0 + \sum_{l=1}^{k-1} \sum_{s=(s_1,\ldots,)\ ,1\le s_1\le s_2\le s_3;\sum s_i=l} \alpha_s \prod \mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})]^{s_i} \tag{2.22}$$

*satisfies $\mathbb{E}[U_k] = \mathbb{E}[V_k]$ for all $\mathbf{A} \in \mathbb{R}^{p\times q}$.*

Obviously, both $\mathbb{E}[V_k]$ and $\mathbb{E}[U_k]$ are polynomials with respect to the entries of $\mathbf{A}$. Since these two polynomials take the same value for all $\mathbf{A} \in \mathbb{R}^{p\times q}$ and the since the polynomial evaluation morphism is injective (see Corollary 1.6 in [69]), this implies that both $\mathbb{E}[U_k]$ and $\mathbb{E}[V_k]$ correspond to the same polynomial in $\mathbb{R}[X_{ij}, 1 \le i \le p; 1 \le j \le q]$.

We now deduce that $U_k = V_k$. Let $Y \sim \mathcal{N}(x, 1)$. For any non-negative integer $\beta$, there exist a polynomial $G_\beta$ of degree $\beta$ such that $\mathbb{E}[Y^\beta] = G_\beta(x)$. Besides, its term of degree $\beta$ is exactly $x^\beta$. To see this, it suffices to observe to use the binomial formula on $\mathbb{E}[Y^\beta] = \mathbb{E}[(x + (Y - x))^\beta]$.

Consider the vector space morphism $\phi : \mathbb{R}[(X_{ij})] \to \mathbb{R}[(X_{ij})]$ such that $\phi[\prod_{i=1,j=1}^{p,q} X_{ij}^{\beta_{ij}}] = \prod_{i=1,j=1}^{p,q} G_{\beta_{ij}}[X_{ij}]$. Then, $\mathbb{E}(V_k) = \phi[(V_k)](\mathbf{A})$ and $\mathbb{E}(U_k) = \phi[(U_k)](\mathbf{A})$. Thus $\phi(U_k) = \phi(V_k)$. We claim that $\phi$ is an injective morphism which implies that $U_k = V_k$ and concludes the proof.

Let us show that $\phi$ is injective. For any monomial of the form $Z = \prod_{i=1,j=1}^{p,q} X_{ij}^{\beta_{ij}}$. Observe that $\phi(Z) - Z$ has a total degree less than $\sum_{ij} \beta_{ij}$. Given a polynomial $P$ and the sum $P_l$ of monomials of largest total degree, we derive $\phi(P) - P_l$ has a total degree less than that of $P$. Thus, $\phi(P)$ has the same total degree as $P$ and $\phi$ is therefore injective.

□

*Proof of Lemma 7.* Given a sequence of integers $1 \le \alpha_1 \le \alpha_2 \ldots \le \alpha_r$. We argue that

$$\mathbb{E}\left[\prod_{t=1}^{r} \mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})^{\alpha_t}]\right] - \prod_{t=1}^{r} \mathrm{tr}[(\mathbf{A}^T\mathbf{A})^{\alpha_t}] \tag{2.23}$$

is a symmetric polynomial with total degree less than $\sum_i \alpha_i$ with respect to $(\sigma_i^2(\mathbf{A}),\ i = 1, \ldots, q)$. Since the space of symmetric polynomials is spanned by the Newton sums, there exist coefficients $\beta_s$ such that

$$\mathbb{E}\left[\prod_{t=1}^{r} \mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})^{\alpha_t}]\right] = \prod_{t=1}^{r} \mathrm{tr}[(\mathbf{A}^T\mathbf{A})^{\alpha_t}] + \sum_{s:\sum s_t < \sum \alpha_t} \beta_s \prod_{t=1}^{l(s)} \mathrm{tr}[(\mathbf{A}^T\mathbf{A})^{s_t}]$$

Then, we prove the existence of $V_k$ by induction. First, $\mathbb{E}[\mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})^k]] - \|\mathbf{A}\|_{2k}^2 = Z_1$ where $Z_1$ is of the form $Z_1 = \sum_{s:\sum s_t<k} \beta_s^{(1)} \prod_{t=1}^{l(s)} \mathrm{tr}[(\mathbf{A}^T\mathbf{A})^{s_t}]$. Then, considering each term of total degree $k-1$, we can estimate them by $\beta_s^{(1)} \prod_{t=1}^{l(s)} \mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})^{s_t}]$. The resulting bias has a total degree less or equal to $k-2$. At each step, one can correct the estimator to decrease the total degree of the bias. The result follows.

**Proof of** (2.23). Write $Z = \prod_{t=1}^{r} \mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})^{\alpha_t}]$. Since $Z$ is invariant by left and right orthogonal transformation of $\mathbf{Y}$, we may assume without loss of generality that $\mathbf{A}$ has null non-diagonal entries. As a consequence, $\mathbf{E}[Z]$ is a polynomial with respect to $\sigma_i(\mathbf{A})$. Since $Z$ is invariant by permutation of the rows and columns of $\mathbf{Y}$, $\mathbf{E}[Z]$ is symmetric. It remains to prove that (i) $\mathbf{E}[Z] - \prod_{t=1}^{r} \prod \mathrm{tr}[(\mathbf{A}^T\mathbf{A})^{\alpha_t}]$ has a total degree less than $2k$ and (ii) that, for all $i = 1, \ldots, q$, the order of $\sigma_i(\mathbf{A})$ in each monomial is even:

For $X \sim \mathcal{N}(x, 1)$, $\mathbb{E}[X^r] - x^r$ is polynomial of degree less than $r$. Since the decomposition of $Z$ leads to a sum of monomials in $\mathbf{Y}_{ij}$ of total degree equals to $\sum_{t=1}^{r} \alpha_t$, it follows that $\mathbb{E}[Z]$ contains the exact same monomial in $A_{ij}$ of total degree $\sum_{t=1}^{r} \alpha_t$ and the remaining terms have a total degree less than that. (i) follows.

Turning to (ii), we decompose again $Z$ into monomials. $Z$ writes as

$$Z = \sum_{(i^{(1)},j^{(1)})\in p^{\alpha_1}\times q^{\alpha_1}} \dots \sum_{(i^{(r)},j^{(r)})\in p^{\alpha_r}\times q^{\alpha_r}} \prod_{l=1}^{\alpha_1} \mathbf{Y}_{i_l^{(1)} j_l^{(1)}} \mathbf{Y}_{i_{l+1}^{(1)} j_l^{(1)}} \dots, \mathbf{Y}_{i_l^{(l)} j_l^{(l)}} \mathbf{Y}_{i_{l+1}^{(r)} j_l^{(r)}}$$

As a consequence, for any index $r \in \{1,\dots,p\}$, the set $\{\mathbf{Y}_{rs}; s = 1,\dots,q\}$ is visited an even number of times in each monomial. Besides, the expectation of one monomial is non zero only if each non diagonal element $\mathbf{Y}_{rs}$ is visited an even number of times since $\mathbf{A}_{rs} = 0$ and the normal distribution is symmetric. As a consequence of these two observations, each $\mathbf{Y}_{ii}$ must be visited an even number of times in the monomial so that its expectation is non zero. Since for $X \sim \mathcal{N}(x,1)$, $\mathbb{E}[X^{2r}]$ is an even polynomial in $x$, this implies that the expectation of each monomial of $Z$ only involves terms of the form $\sigma_i^{2r}(\mathbf{A})$ and (ii) follows.

□

*Proof of Corollary 2.* Since $l_r$ norms of vectors are non increasing with respect to $r$, we have $\|\mathbf{A}\|_{4k-2}^{4k-2} \le \|\mathbf{A}\|_{2k}^{4k-2}$ and $\|\mathbf{A}\|_{4k-4}^{4k-4} \le \|\mathbf{A}\|_{2k}^{4k-4}$. Hence, we derive from Proposition 7 and Chebychev inequality that

$$\mathrm{Var}\,(U_k)^{1/2} \le c_{\exp}(k)\left[(pq)^{k/2} + p^{1/2}\|\mathbf{A}\|_{2k}^{2k-2} + \|\mathbf{A}\|_{2k}^{2k-1}\right] . \tag{2.24}$$

We consider two cases depending on $\|\mathbf{A}\|_{2k}^{2k}$. If $\|\mathbf{A}\|_{2k}^{2k} \le (pq)^{k/2}$, then the above risk bound simplifies in $\mathbb{E}\left[|(U_k) - \|\mathbf{A}\|_{2k}^{2k}\right] \le c_{\exp}(k)(pq)^{k/2}$. Then, together with the inequality $|x^{1/(2k)} - y^{1/(2k)}| \le |x-y|^{1/(2k)}$ and Holder inequality, we have

$$\mathbb{E}\left[|(U_k)_+^{1/(2k)} - \|\mathbf{A}\|_{2k}|\right] \le \mathbb{E}\left[|U_k - \|\mathbf{A}\|_{2k}^{2k}|\right]^{1/2k} \le c_{\exp}[k](pq)^{1/4} . \tag{2.25}$$

Let us turn to the case where $\|\mathbf{A}\|_{2k}^{2k} \ge (pq)^{k/2}$. Since $|(1+x)^{1/(2k)} - 1| \le |x|$ for any $x \ge -1/2$, we have

$$\begin{aligned}
\mathbb{E}\left[|(U_k)_+^{1/(2k)} - \|\mathbf{A}\|_{2k}|\right] &= \mathbb{E}\left[\|\mathbf{A}\|_{2k}\left|[U_k/\|\mathbf{A}\|_{2k}^{2k}]^{\frac{1}{2k}} - 1\right|\right] \\
&\le \|\mathbf{A}\|_{2k}\,\mathbb{P}\left[U_k \le \|\mathbf{A}\|_{2k}^{2k}/2\right] + \|\mathbf{A}\|_{2k}^{1-2k}\,\mathbb{E}\left[|U_k - \|\mathbf{A}\|_{2k}^{2k}|\right] \\
&\le \|\mathbf{A}\|_{2k}\left[4\frac{\mathrm{Var}\,(U_k)}{\|\mathbf{A}\|_{2k}^{4k}} + \frac{\mathrm{Var}\,(U_k)^{1/2}}{\|\mathbf{A}\|_{2k}^{2k}}\right] \\
&\le c_{\exp}(k)\left[\frac{(pq)^k}{\|\mathbf{A}\|_{2k}^{4k-1}} + \frac{(pq)^{k/2}}{\|\mathbf{A}\|_{2k}^{2k-1}} + \frac{p}{\|\mathbf{A}\|_{2k}^3} + \frac{p^{1/2}}{\|\mathbf{A}\|_{2k}} + \frac{1}{\|\mathbf{A}\|_{2k}} + 1\right] \\
&\le c_{\exp}(k)(pq)^{1/4} ,
\end{aligned} \tag{2.26}$$

where we used (2.24) in the fourth line and the condition $\|\mathbf{A}\|_{2k} \ge (pq)^{1/4}$ in the last line.

□

*Proof of Proposition 6.* The definition of the estimators $U_2$ and $U_3$ is a consequence of the following moment identities

$$\mathbb{E}[\mathrm{tr}(\mathbf{Y}^T\mathbf{Y})] = \|\mathbf{A}\|_2^2 + pq ; \tag{2.27}$$

$$\mathbb{E}[\mathrm{tr}((\mathbf{Y}^T\mathbf{Y})^2)] = \|\mathbf{A}\|_4^4 + 2(p+q+1)\|\mathbf{A}\|_2^2 + pq(1+q+p) ; \tag{2.28}$$

$$\begin{aligned}
\mathbb{E}[\mathrm{tr}((\mathbf{Y}^T\mathbf{Y})^3)] &= \|\mathbf{A}\|_6^6 + 3(p+q+1)\|\mathbf{A}\|_4^4 + 3(p^2+q^2+3pq+3p+3q+4)\|\mathbf{A}\|_2^2 \\
&\quad + pq[p^2+q^2+3pq+3p+3q+4] .
\end{aligned} \tag{2.29}$$

After some tedious computations, we see that the expressions in the right-hand side in Proposition 6 are unbiased estimators of $\|\mathbf{A}\|_2^2$ and $\|\mathbf{A}\|_4^4$ respectively.

In the remainder of the proof, we show (2.27–2.29). The first identity has already been proved to analyze $U_1$. In order to derive the second and the third identity, we decompose $\mathbf{Y}\mathbf{Y}^T$ into a sum involving its expectation.

$$\mathbf{Y}^T\mathbf{Y} = \left(\mathbf{A}^T\mathbf{A} + p\mathbf{I}_q\right) + \left(\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q + \mathbf{E}^T\mathbf{A} + \mathbf{A}^T\mathbf{E}\right) =: \mathbf{S} + \mathbf{N} \ .$$

Since $\mathbf{S}$ is deterministic and $\mathbf{N}$ is centered, we have

$$\mathbb{E}[\mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})^2]] = \mathrm{tr}(\mathbf{S}^2) + \mathrm{tr}[\mathbb{E}(\mathbf{N}^2)] \ .$$

Since the Gaussian distribution is symmetric, we obtain by straightforward computation that

$$\mathbb{E}[\mathbf{N}^2] = \mathbb{E}[(\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q)^2] + \mathbb{E}[(\mathbf{A}^T\mathbf{E} + \mathbf{E}^T\mathbf{A})^2] = (q+1)\left[p\mathbf{I}_q + 2\mathbf{A}^T\mathbf{A}\right] \ . \tag{2.30}$$

Combining the two previous identities leads to (2.28). Turning to (2.29), we have

$$\mathbb{E}[\mathrm{tr}[(\mathbf{Y}^T\mathbf{Y})^2]] = \mathrm{tr}(\mathbf{S}^3) + \mathrm{tr}[\mathbb{E}(\mathbf{N}^3)] + 3\,\mathrm{tr}[\mathbf{S}\,\mathbb{E}(\mathbf{N}^2)] \ , \tag{2.31}$$

since $\mathbf{N}$ is centered. For $\mathbb{E}[\mathrm{tr}(\mathbf{N}^3)]$, we use a again that odd moments of centered normal distributions are null to derive

$$\mathbb{E}[\mathrm{tr}(\mathbf{N}^3)] = \mathbb{E}[\mathrm{tr}[(\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q)^3]] + 3\,\mathbb{E}\left[\mathrm{tr}[(\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q)(\mathbf{A}^T\mathbf{E} + \mathbf{E}^T\mathbf{A})^2]\right] \ . \tag{2.32}$$

Starting with the first expression, we derive from moments of standard Wishart distribution (see e.g. [73]) that

$$\mathbb{E}[\mathrm{tr}[(\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q)^3]] = pq[q^2 + 3q + 4] \ .$$

Tedious computations also lead us to

$$\mathbb{E}\left[\mathrm{tr}[(\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q)(\mathbf{A}^T\mathbf{E} + \mathbf{E}^T\mathbf{A})^2]\right] = \|\mathbf{A}\|_2^2[(q-1)(q-2) + 3(q-1)2 + 8] = \|\mathbf{A}\|_2^2[q^2 + 3q + 4] \ .$$

Combining the two previous identities in (2.32), we obtain

$$\mathbb{E}[\mathrm{tr}(\mathbf{N}^3)] = [q^2 + 3q + 4](pq + 3\|\mathbf{A}\|_2^2) \ .$$

Then, coming back to (2.31), and relying on (2.30), we get

$$\begin{aligned}
\mathbb{E}[\mathrm{tr}((\mathbf{Y}^T\mathbf{Y})^3)] &= \mathrm{tr}[(\mathbf{A}^T\mathbf{A} + p\mathbf{I}_q)^3] + [q^2 + 3q + 4](pq + 3\|\mathbf{A}\|_2^2) + 3(q+1)\,\mathrm{tr}[(\mathbf{A}^T\mathbf{A} + p\mathbf{I}_q)(2\mathbf{A}^T\mathbf{A} + p\mathbf{I}_q)]] \\
&= \|\mathbf{A}\|_6^6 + 3(p+q+1)\|\mathbf{A}\|_4^4 + 3(p^2 + q^2 + 3pq + 3p + 3q + 4)\|\mathbf{A}\|_2^2 \\
&\quad + p^3q + q^3p + 3p^2q^2 + 3p^2q + 3q^2p + 4pq \ ,
\end{aligned}$$

which is exactly (2.29).

□

### 2.7.4 Proofs for general norms $\|\mathbf{A}\|_s$

We start with a technical lemma.

**Lemma 8.** *Write $I(\mathbf{A})$ for the image of $\mathbf{A}$. Then, $\Pi_{I(\mathbf{A})}$ be any orthogonal projection matrix in $\mathbb{R}^p$ to $I(\mathbf{A})$. For any $i = 1, \ldots, q$, we have*

$$|\lambda_i(\mathbf{W}) - \sigma_i^2(\mathbf{A})| \leq 2\sigma_i(\mathbf{A})\|\Pi_{I(\mathbf{A})}\mathbf{E}\|_\infty + \|\Pi_{I(\mathbf{A})}\mathbf{E}\|_\infty^2 + \|\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q\|_\infty \tag{2.33}$$

*Proof of Proposition 8.* Define $\mathbf{W} = \mathbf{Y}^T\mathbf{Y} - p\mathbf{I}_q$, then $\sigma_i^2(\mathbf{Y}) - p = \lambda_i(\mathbf{Y})$ so that $T_s = [\sum_{i=1}^q (\lambda_i(\mathbf{W}))_+^{s/2}]^{1/s}$. For any $x \in \mathbb{R}$, we have $|\sqrt{(1+x)_+} - 1| \leq 2|x| \wedge \sqrt{|x|}$. It follows from the triangular inequality and Lemma 8 that

$$\begin{aligned}
|T_s - \|\mathbf{A}\|_s| &\leq \left[\sum_{i=1}^q |[\lambda_i(\mathbf{W})]_+^{1/2} - \sigma_i(\mathbf{A})|^s\right]^{1/s} \leq \sum_{i=1}^q |[\lambda_i(\mathbf{W})]_+^{1/2} - \sigma_i(\mathbf{A})| \\
&\leq \sum_{i=1}^q \|\Pi_{I(\mathbf{A})}\mathbf{E}\|_\infty + \|\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q\|_\infty^{1/2} \\
&\quad + \sigma_i(\mathbf{A}) \left[\left|\sqrt{1 + 2\frac{\|\Pi_{I(\mathbf{A})}\mathbf{E}\|_\infty}{\sigma_i(\mathbf{A})}} - 1\right| \vee \left|\sqrt{\left(1 - 2\frac{\|\Pi_{I(\mathbf{A})}\mathbf{E}\|_\infty}{\sigma_i(\mathbf{A})}\right)_+} - 1\right|\right] \\
&\leq q\left[5\|\Pi_{I(\mathbf{A})}\mathbf{E}\|_\infty + \|\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q\|_\infty^{1/2}\right] .
\end{aligned}$$

This leads us to $\mathbb{E}[|T_1 - \|\mathbf{A}\|_1| \leq q\left[5\,\mathbb{E}\left[\|\Pi_{I(\mathbf{A})}\mathbf{E}\|_\infty\right] + \mathbb{E}[\|\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q\|_\infty]^{1/2}\right]$ . Since $\mathbf{E}$ is distribution invariant by a left orthogonal transformation $\|\Pi_{I(\mathbf{A})}\mathbf{E}\|_\infty$ is distributed as the operator norm of a $\dim(I(\mathbf{A})) \times q$ noise matrix. Thus, we deduce from Lemma 9 that $\mathbb{E}\left[\|\Pi_{I(\mathbf{A})}\mathbf{E}\|_\infty\right] \leq 2\sqrt{q}$. The expectation of $\mathbb{E}[\|\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q\|_\infty$ has already been handled in Lemma 4. We conclude that

$$\mathbb{E}[|T_1 - \|\mathbf{A}\|_1|] \leq \mathbb{E}[\sum_{i=1}^q |(\sigma_i^2(\mathbf{Y}) - p)_+^{1/2} - \sigma_i(\mathbf{A})|] \leq cq(pq)^{1/4} .$$

□

*Proof of Lemma 8.* Without loss of generality, assume that the $\mathbf{A}$ matrix is in diagonal form, that $\mathbf{A}_{ii} = \sigma_i(\mathbf{A})$ for $i = 1, \ldots, q$ and $\mathbf{A}_{ij}$ is zero otherwise. Then, we deduce from the interlacing inequality (Corollary III. 1.5 in [12]) that the $i$-th largest eigenvalues of $\mathbf{W} = \mathbf{Y}^T\mathbf{Y} - p\mathbf{I}_q$ is less of or equal to to the first eigenvalue of the restriction of $\mathbf{W}$ to indices in $[i, \ldots, q] \times [i, \ldots, q]$. Writing $\mathbf{W}_{[i:q]}$ for this restriction, we arrive at

$$\begin{aligned}
\lambda_i(\mathbf{W}) &\leq \lambda_1(\mathbf{W}_{[i:qq]}) \\
&\leq \sigma_i^2(\mathbf{A}) + \|[\mathbf{E}^T\mathbf{A} + \mathbf{A}^T\mathbf{E} + \mathbf{E}^T\mathbf{E} - p\mathbf{I}_q]_{[i:q]}\|_\infty \\
&\leq \sigma_i^2(\mathbf{A}) + \|\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q\|_\infty + 2\|[\mathbf{E}^T\mathbf{A}]_{[i:q]}\|_\infty \\
&\leq \sigma_i^2(\mathbf{A}) + \|\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q\|_\infty + 2\sigma_i(\mathbf{A})\|[\mathbf{E}_{[1:q]}\|_\infty ,
\end{aligned}$$

where we applied again the interlacing inequality in the second and in the third line.

Turning the lower bound of $\lambda_i(\mathbf{W})$, we define $V_i$ the subspace of dimension $i$ spanned by the $i$-th largest eigenvectors of $\mathbf{A}^T\mathbf{A}$ (pick any such subspace if it is not unique). It follows from Courant-Fischer min-max theorem that

$$\begin{aligned}
\lambda_i(\mathbf{W}) &\geq \inf_{x \in V_i} (x^T\mathbf{W}x)/|x|_2^2 \\
&\geq \inf_{x:|x|_2=1} \sum_{j=1}^i x_j^2\sigma_j^2(\mathbf{A}) + x^T(\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q)x + 2\sum_{j_1,j_2=1}^i x_{j_1}x_{j_2}\sigma_{j_1}(\mathbf{A})\mathbf{E}_{j_1j_2} \\
&\geq \inf_{x:|x|_2=1} \sum_{j=1}^i x_j^2\sigma_j^2(\mathbf{A}) - 2[\sum_{j=1}^i x_j^2\sigma_j^2(\mathbf{A})]^{1/2}\|\mathbf{E}_{[1:q]}\|_\infty - \|\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q\|_\infty , \qquad (2.34)
\end{aligned}$$

where we applied Cauchy-Schwarz inequality and used that $\inf f + g \geq \inf f + \inf g$. The quantity $\sum_{j=1}^i x_j^2\sigma_j^2(\mathbf{A})$ lies in $[\sigma_i^2(\mathbf{A}); \sigma_1^2(\mathbf{A})]$. The function $x \mapsto x^2 - 2xz$ is decreasing for $x \leq z$ and

increasing for $x \geq z$ and its minimum equals $-z^2$. When $\sigma_i(\mathbf{A}) \geq \|\mathbf{E}_{[1:q]}\|_\infty$, the minimum of the left-hand side expression in (2.34) is achieved at $\sigma_i(\mathbf{A})$ and we have

$$\lambda_i(\mathbf{W}) \geq \sigma_i^2(\mathbf{A}) - 2\sigma_i(\mathbf{A})\|\mathbf{E}_{[1:q]}\|_\infty - \|\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q\|_\infty \geq \sigma_i^2(\mathbf{A}) - 2\sigma_i(\mathbf{A})\|\mathbf{E}_{[1:q]}\|_\infty - \|\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q\|_\infty - \|\mathbf{E}_{[1:q]}\|_\infty^2 .$$

When $\sigma_i(\mathbf{A}) < \|\mathbf{E}_{[1:q]}\|_\infty$, this implies that $\sigma_i^2(\mathbf{A}) - 2\sigma_i(\mathbf{A})\|\mathbf{E}_{[1:q]}\|_\infty < 0$, and we also have

$$\lambda_i(\mathbf{W}) \geq \sigma_i^2(\mathbf{A}) - 2\sigma_i(\mathbf{A})\|\mathbf{E}_{[1:q]}\|_\infty - \|\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q\|_\infty - \|\mathbf{E}_{[1:q]}\|_\infty^2 .$$

All in all, we have proved that

$$\left|\lambda_i(\mathbf{W}) - \sigma_i^2(\mathbf{A})\right| \leq \|\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q\|_\infty + 2\sigma_i(\mathbf{A})\|[\mathbf{E}_{[1:q]}\|_\infty + \|\mathbf{E}_{[1:q]}\|_\infty^2 .$$

Since $\mathbf{E}_{[1:q]}$ is a submatrix of $\Pi_{I(\mathbf{A})}(\mathbf{E})$ (recall that $\mathbf{A}$ is assumed to be diagonal in the proof), we get the desired result. □

*Proof of Proposition 9.* The first inequality in (2.14) is straightforward. Regarding the second inequality, observe that for a diagonal matrix $\mathbf{A}$, its singular values are given by the absolute values of the its diagonal entries, so that $\|\mathbf{A}\|_1 = \sum_{i=1}^q |\mathbf{A}_{ii}|$. As a consequence, $\inf_{\widehat{T}} \sup_{\mathbf{A}\in\mathcal{D}_{p,q}} \mathbb{E}_{\mathbf{A}}[|\widehat{T} - \|\mathbf{A}\|_1|]$ corresponds to the minimax risk of estimating the $l_1$ norm of vector $\theta$ of size $q$ in a model $Y_i = \theta_i + \epsilon_i$ where the $\epsilon_i$'s are i.i.d. and follow the normal distribution. Fortunately, this problem has already been considered by Cai and Low [18]. In their theorem 3, they prove that the minimax square risk is higher than $c'q^2/\log(q)$. Following their arguments and replacing the square loss by the absolute loss, we derive

$$\inf_{\widehat{T}} \sup_{\mathbf{A}\in\mathcal{D}_{p,q}} \mathbb{E}[|\widehat{T} - \|\mathbf{A}\|_1|] \geq c\frac{q}{\sqrt{\log(q)}} .$$

□

### 2.7.5 Proof of Proposition 10

Recall that $\operatorname{tr}[\mathbf{Y}^T\mathbf{Y} - p\mathbf{I}_q] = \|\mathbf{A}\|_2^2 + 2\operatorname{tr}[\mathbf{A}^T\mathbf{E}] + (\operatorname{tr}[\mathbf{E}^T\mathbf{E}] - pq)$. The second expression follows a centered normal distribution with variance $4\|\mathbf{A}\|_2^2$ whereas the last expression follows a $\chi^2$ distribution with $pq$ degrees of freedom. From Lemma 11, we deduce that, with probability higher than $1 - 4e^{-t}$,

$$\left|\operatorname{tr}[\mathbf{Y}^T\mathbf{Y} - p\mathbf{I}_q] - \|\mathbf{A}\|_2^2\right| \leq 2\sqrt{pqt} + 2t + 2\|\mathbf{A}\|_2\sqrt{2t} .$$

Regarding the operator norm, we start from (2.17)

$$\left|\sigma_1(\mathbf{Y}^T\mathbf{Y} - p\mathbf{I}_q) - \|\mathbf{A}^T\mathbf{A}\|_\infty\right| \leq 2\|\mathbf{A}^T\mathbf{E}\|_\infty + \|\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q\|_\infty .$$

From Lemma 10, we deduce that, with probability higher than $1 - e^{-t}$, $\|\mathbf{E}^T\mathbf{E} - p\mathbf{I}_q\|_\infty \leq q + 2\sqrt{pq} + 4\sqrt{2pt} + 2t$. In the proof of Lemma 3, we have shown that $\|\mathbf{A}^T\mathbf{E}\|_\infty / \|\mathbf{A}\|_\infty$ is stochastically dominated by the operator norm of a $q \times q$ matrix with independent normal entries. Invoking again Lemma 10, we derive that $2\|\mathbf{A}^T\mathbf{E}\|_\infty \leq 2\sigma_1(\mathbf{A})(2\sqrt{q} + \sqrt{2t})$ with probability higher than $1 - e^{-t}$. Putting everything together, this yields

$$\frac{\|\mathbf{A}\|_2^2}{\sigma_1^2(\mathbf{A})} \cdot \frac{1 - [2\sqrt{pqt} + 2t]/\|\mathbf{A}\|_2^2 - 2\sqrt{2t}/\|\mathbf{A}\|_2}{1 + 2\frac{2\sqrt{q}+\sqrt{2t}}{\sigma_1(\mathbf{A})} + \frac{3\sqrt{pq}+4\sqrt{2pt}+2t}{\sigma_1^2(\mathbf{A})}} \leq \widehat{\mathrm{ER}}_{2,\infty}(\mathbf{A}) \leq \frac{\|\mathbf{A}\|_2^2}{\sigma_1^2(\mathbf{A})} \cdot \frac{1 + [2\sqrt{pqt} + 2t]/\|\mathbf{A}\|_2^2 + 2\sqrt{2t}/\|\mathbf{A}\|_2}{1 - 2\frac{2\sqrt{q}+\sqrt{2t}}{\sigma_1(\mathbf{A})} - \frac{3\sqrt{pq}+4\sqrt{2pt}+2t}{\sigma_1^2(\mathbf{A})}} ,$$

which, assuming that both $\|\mathbf{A}\|_2$ and $\|\mathbf{A}\|_\infty$ are large enough, implies that

$$\frac{\left|\widehat{\mathrm{ER}}_{2,\infty}(\mathbf{A}) - \mathrm{ER}_{2,\infty}(\mathbf{A})\right|}{\mathrm{ER}_{2,\infty}(\mathbf{A})} \lesssim \frac{\sqrt{pqt}}{\|\mathbf{A}\|_2^2} + \frac{\sqrt{pt} + \sqrt{pq}}{\|\mathbf{A}\|_\infty^2} + \frac{\sqrt{q} + \sqrt{t}}{\|\mathbf{A}\|_\infty}$$

## 2.A Technical inequalities

For bounding the singular values of $\mathbf{E}$, we shall rely on the following results (taken from [28]).

**Lemma 9.** *Let $\mathbf{E}$ be a $p \times q$ whose entries are independent and follow a standard normal distribution. Then,*

$$\mathbb{E}\left[|\sigma_1(\mathbf{E})|\right] \leq (\sqrt{p} + \sqrt{q})$$

**Lemma 10.** *Let $\mathbf{E}$ be a $p \times q$ whose entries are independent and follow a standard normal distribution. Then, for any $t > 0$, we have*

$$\max\left\{\mathbb{P}\left(\sigma_1\left(\mathbf{E}\right) \geq \sqrt{p} + \sqrt{q} + \sqrt{2t}\right), \mathbb{P}\left(\sigma_q\left(\mathbf{E}\right) \leq \sqrt{p} - \sqrt{q} - \sqrt{2t}\right)\right\} \leq e^{-t} \ . \tag{2.35}$$

The following lemma is taken from [70]

**Lemma 11.** *Let $Z$ be distributed as $\chi^2(p)$ random variable. For any $t > 0$, we have*

$$\mathbb{P}[Z \geq p + 2\sqrt{pt} + 2t] \leq e^{-t} \ .$$

# CHAPTER 3

# SEGMENTATION ON TREES

## 3.1 Introduction

Given a process $(y_s)$ indexed by the nodes of a graph $G$, the problem of segmentation is that of finding regions of homogeneous distributions in the graph $G$ such that the distribution of the data is homogeneous in the graph. When the graph $G$ is a line (or a chain graph), this boils down to the celebrated change-point detection problem [121] which finds important applications in sound recognition, finance, genomics,...(see [135], [3] and [92] for examples of applications). In image analysis, segmenting an image amounts to partitionning the image into homogeneous regions and the corresponding graph $G$ is usually a two-dimensional grid.

In some applications, data exhibit a tree-like structure. Think for instance of species traits along a phylogenetic tree [89]. River network is another example of tree structure, so that abundance measurements along rivers fall in that framework. In this manuscript, we argue that many methodologies for change-point detection on time series can be extended to trees, thereby making the segmentation problem significantly simpler for general graphs. Whereas there is rich literature on the chain cases or on general graphs, the tree case have received little attention despite the article of Maravalle et al on clustering on trees [85]. The purpose of this manuscript is to partially fill this gap.

### 3.1.1 Model

Suppose we are given a tree T and that we observe a process $y_s \in \mathcal{Y}$ indexed by the nodes $s$ of T. The problem of segmenting the tree T is that of finding a partition of the nodes into connected components in such a way that random variable $(y_s)$ inside the same connected component are identically distributed. This model is quite general in the way the distribution of $y_s$ may vary between two regions. In applications, one often works under further parametric assumptions. In particular, we shall sometimes consider the specific mean-change problem, where $\mathcal{Y} = \mathbb{R}^p$ and

$$y_s = \mu_s^* + \epsilon_s \ . \tag{3.1}$$

where $\mu_s^*$ is the unknown mean of $y_s$ and the $(\epsilon_s)$ are independent realizations of a centered random Gaussian vector with known covariance $\sigma^2 \mathbf{I}_p$. For (3.1), only the mean of $(y_s)$ varies in the tree T. Other parametric models will be introduced and discussed along this manuscript.

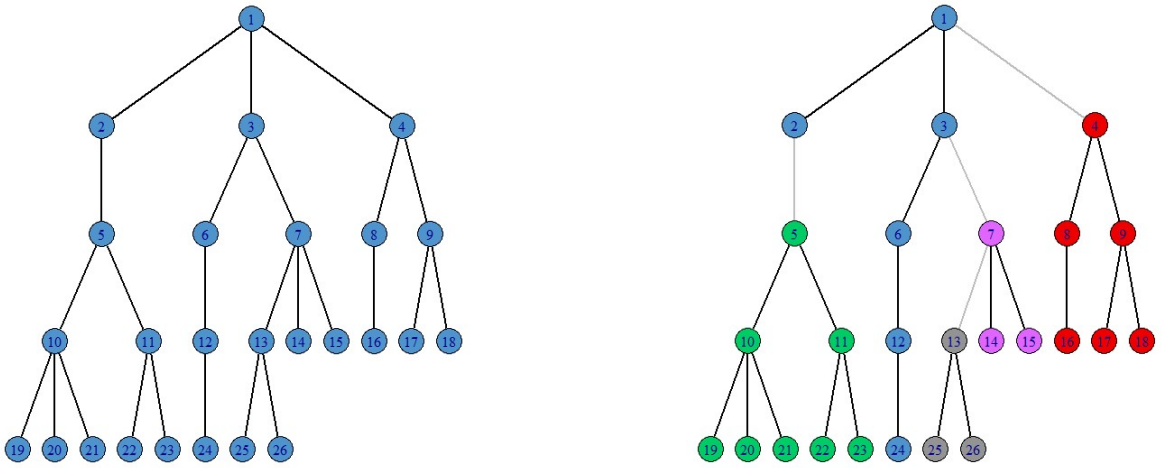
**Graph formalism.** For a graph $G = (\mathcal{V}, \mathcal{E})$, $\mathcal{V}$ stands for its nodes set $\mathcal{E}$ stands for its collection of (undirected) edges, that is of subsets of $\mathcal{V}$ of size 2. We say that a connected graph $\mathrm{T} = (\mathcal{V}, \mathcal{E})$ is a tree if it does not contain any cycle. with and edge set $\mathcal{E}$. To alleviate the notation, we identify henceforth a tree T and its set of nodes set. In other wordes, we also write T for $\mathcal{V}$. To remove any ambiguity, we sometimes specify whether we consider the set T or the tree T.

Given a subset of nodes $S \subset T$, the induced subgraph $S$ is made of edges of T that connect nodes in $S$. The induced subgraph $S$ is not necessarily connected and therefore consists of a collection of disconnected trees, which is called a forest. If $S$ is connected, then we say that $S$ is a sub-tree of T. The size $|T|$ of a tree T corresponds to its number of nodes.

**Segmentation.** We say that a partition $\mathbf{P} = \{S_1, \ldots, S_q\}$ of the set $T$ is a segmentation of T if all $S_j$'s are connected (or equivalently are sub-trees of T). In this work, all considered partitions are segmentations and we use indifferently both terminologies. We write $\mathcal{P}(\mathrm{T})$ (or simply $\mathcal{P}$ when there is no ambiguity) for the collection of such partitions of the set $T$. For a partition $\mathbf{P} \in \mathcal{P}$, its size $|\mathbf{P}|$ stands for its number of sub-trees.

Given a partition $\mathbf{P} = \{S_1, \ldots, S_q\}$ of T into $q$ sub-trees, we denote $\mathcal{J}_\mathbf{P}$ the set of crossing edges, that is the set of edges $\{s, t\}$ such that $s$ and $t$ do not belong to the same sub-tree $S_i$. We call such an edge a break edge.

Since T has a tree structure, we claim that the number $|\mathcal{J}_\mathbf{P}|$ of break edges is $q-1$. Conversely, any subset $\mathcal{J}$ of edges of size $q-1$ defines a segmentation $\mathbf{P}_\mathcal{J}$ of size $q$. See the appendix for a proof of this claim. As a consequence, it is equivalent to estimate a segmentation or a set of break edges. In our work, we shall mostly work with segmentation, but we shall sometimes rely on the break edge representation.



(a) (b)

Figure 3.1 – Figure (a) represent a tree of root 1. All nodes are represented in blue. Figure (b) represent a partition $\mathbf{P}$ of the same tree. The set of all the nodes represented in a same color and the set of the edges connecting the same color nodes define the sub-trees of $\mathbf{P}$. The grey edges represent the breaks associated to $\mathbf{P}$. These are the edges from the set $\mathcal{J}_\mathbf{P}$.

**Our Problem.** Given observation $(y_s)_{s \in T}$ indexed by the nodes of the tree, we denote $\underline{\mathcal{J}}$ the set of break edges, that is the edges such that the random variables $y_t$ and associated endpoints of the edges are not identically distributed. The corresponding segmentation $\underline{\mathbf{P}}$ of T is the minimal partition breaking down T into sub-trees on which the random variables are identically distributed. In this manuscript, we consider the problem of recovering the segmentation $\underline{\mathbf{P}}$ from a single observation of the process $(y_s)_{s \in \mathrm{T}}$.

**Example of the chain graph.** Let us consider the specific case where T is the chain graph of size $n$: the set of nodes is $\{1, \ldots, n\}$ whereas the edge set is defined as $\mathcal{E} = \{\{i, i+1\}, i = 1, \ldots, n-1\}$. The chain graph boils down to the classical change-point detection problem for time series. Then, a segmentation of the nodes is made of segments of the form $[(\tau_s + 1); \tau_{s+1}]$ with $\tau_0 = 0$ and $\tau_K = n$ and the break edges are the edges $\{\tau_s, \tau_s + 1\}$ for $s = 1, \ldots, K-1$. Hence, $\mathcal{P}$ stands for the collection

of partition of $\{1, \dots, n\}$ into segments. As stated earlier, one of the purposes of this manuscript is to extend some change-point detection methods for the chain graph to the general tree setting.

### 3.1.2 Segmentation by cost minimization

An omnibus method for performing change-point detection on the chain graph is based on the cost minimization framework. Let us introduce it in our tree segmentation setting. In what follows, we say that a function $\mathrm{Cost}_{\mathrm{T}} : \mathcal{P} \to \mathbb{R}$ is a cost function, if it satisfies a linear decomposition with respect to the element of the partitions, that is, for any $\mathbf{P}$ we have the decomposition of $\mathrm{Cost}_{\mathrm{T}}(\mathbf{P})$ as

$$\mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) = \sum_{S \in \mathbf{P}} \mathrm{C}_S \ , \tag{3.2}$$

where $C_S$ is only a function of $S$ and of the observations $\mathbf{y}_S = (y_i)_{i \in S}$. Note that the cost function $\mathrm{Cost}_{\mathrm{T}}$ can therefore be defined through the $(C_S)$'s for all $S$ sub-trees of T. Examples of cost functions include:

- [*Least-square cost* [4]] $\mathcal{Y} = \mathcal{H}$ is a Hilbert space endowed with the norm $\|.\|$. Then, the least-square cost

$$C_S = \arg\min_{u \in \mathcal{H}} \sum_{s \in S} \|y_i - u\|^2 = \sum_{s \in S} \left\| y_i - \frac{\sum_{j \in S} y_j}{|S|} \right\|^2 \ . \tag{3.3}$$

  The corresponding Cost function is the least-square criterion over all mean vector that are constant inside each sub-tree $S \in \mathbf{P}$. Such cost function is particularly suited to detect changes in expectation in the $(y_s)$ when those follow a Gaussian distribution.

- [*Linear penalized Least-square cost*] We have still $\mathcal{Y} = \mathcal{H}$ and are given some $\beta > 0$. Define

$$C_S = \sum_{s \in S} \left\| y_i - \frac{\sum_{j \in S} y_j}{|S|} \right\|^2 + \beta \ . \tag{3.4}$$

  The corresponding Cost function is the least-square criterion penalized by the number $|\mathbf{P}|$ of components. Up to an additive constant $\beta$, this is equivalent to penalizing the number $|\mathbf{P}| - 1$ of break edges.

- [*Minimum of negative log-likelihood*]. We are given a parametric model $\mathbb{P}_\theta$ with $\theta \in \Theta$. For a given partition $\mathbf{P} = (S_1, \dots S_q)$, we assume that there exist $(\theta_1, \dots, \theta_q)$ such that any $s \in S_j$, $y_s \sim \mathbb{P}_{\theta_j}$, that is the parameters of the distributions are constants inside each sub-tree of the partition. Writing $l(y, \theta_{\mathbf{P}})$ for the log-likelihood of the data, the minimum negative log-likelihood

$$\mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) = \min_{\theta_{\mathbf{P}} \in \Theta^{|\mathbf{P}|}} l(y, \theta_{\mathbf{P}}) = \sum_{S \in \mathbf{P}} \min_{\theta \in \Theta} l(\mathbf{y}_S, \theta) \tag{3.5}$$

  is a valid cost function. One may define such a function for any parametric model. For the chain graph, Gaussian, Poisson and negative binomial models are the most classical ones [22]. As in the previous case, one can alternatively use a penalized version $C_S = \min_{\theta \in \Theta} l(\mathbf{y}_S, \theta) + \beta$.

Given a cost function, the problem of cost minimization is then to find a partition $\mathbf{P}^*$ minimizing the cost function.

$$\mathbf{P}^* \in \arg\min_{\mathbf{P} \in \mathcal{P}} \mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) \ . \tag{3.6}$$

If the cost function does not include any penalty term, it may be of interest to solve a restricted version of (3.6) where we consider the collection $\mathcal{P}_K$ of partitions of size less or equal to $K$.

As explained in the previous section, $|\mathcal{P}| = 2^{n-1}$, so that a naive minimization procedure in (3.6) would have an exponential complexity. For the specific case of the chain graph, dynamic programming algorithms [55] solve the above problem in quadratic time (see below for further explanations).

### 3.1.3 Our Contribution

Our main contribution is twofold. First, we are interested in the computational aspects of general cost minimization procedure on a general tree. Although, there is rich literature on that topic for change point detection on the chain graph, we are not aware on results for general trees. Unfortunately, extensions of Dynamic programming algorithm [10] to general trees suffer from an exponential complexity. This is why we build on the recent pruning strategies of Killick et al [64] to leverage the algorithm complexity. Numerical illustrations suggest that our PELT tree algorithm achieves quadratic complexity in practice. Second, we focus on the multivariate mean-change Gaussian setting (3.1) on a general tree. We introduce a dedicated penalty such that the minimizer $\widehat{\mathbf{P}}$ of the correspond penalized least-square cost achieve near optimal statistical performance. This extends previous results of Lebarbier [71] on the univariate case ($p = 1$) on the chain graph. Finally, our methodology is applied to Fish abundance measurements along the Loire river network in France.

### 3.1.4 Related Literature

We first discuss the literature on change-point detection on the chain graph and then we mention some work on general (non necessarily trees) graphs.

#### Cost Minimization on the chain graph

**Versatility of Cost minimization.** As explained earlier, cost minimization procedure are versatile and encompass many setting including parametric models, multivariate observations... Arlot et al. [4] have even shown that, for any arbitrary space $\mathcal{Y}$ and a kernel function $k : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, one can rely on a least-square cost function in the associated reproducing kernel Hilbert space to estimate the change-points, thereby making the method applicable to many machine learning problems. For the chain tree T, a partition $\mathbf{P}$ of size $q$ can be represented through the jumps $\tau_1, \ldots, \tau_q$. Equipped with the convention $\tau_0 = 1$ and $\tau_q = n$, the additive decomposition (3.2) of the cost function now writes:

$$\mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) = \sum_{i=0}^{q-1} \mathrm{C}_{(\tau_i:\tau_{i+1}]} \ . \tag{3.7}$$

**Dynamic Programming.** The number of possible segmentations of the chain graph of size $n$ is exponential. In his pioneering paper, Bellman [10] introduced the so-called Bellman equations, allowing to find the minimal cost partition of any given size $K$. This contribution which is at the root of the general dynamic programming algorithm can be summarized with the following sentence: A minimum cost partition $\mathbf{P}_K^*$ of size $K$ on the chain graph $[1{:}n]$ is the union of an optimal partition of size $K-1$ on the chain graph $[1{:}\tau]$ and of the segment $[(\tau+1){:}n]$ for some $\tau$, so that it is possible to find iteratively optimal partition on $[1{:}t]$ given the optimal partition over $[1{:}s]$ for all $s < t$. Relying on Bellman's equations, Auger and Lawrence in [5] provide an algorithm for finding all minimal cost partition $\mathbf{P}_k^*$ for $1 \leq k \leq K$ with a time complexity of order $O(Kn^2)$, provided that the evaluation of a cost function $C_S$ requires constant time. More recently, Jackson et al. [55], have adapted Bellman's equations to find a minimum cost partition $\mathbf{P}^*$ without any constrain on its size. To make sense, this cost function must include a penalty term. In Section 3.2, we shall remind the reader on the work of Jackson et al. [55] as a warm-up for the general tree case.

**Pruning methods.** In massive dataset analysis, one cannot apply dynamic quadratic algorithms because of its quadratic complexity. Only quasi-linear algorithms are feasible when $n$ is too large. A flourishing line of research initiated in [64] aims at improving the dynamic to lower its complexity. The quadratic complexity of dynamic algorithm takes its root at Bellman's equations where to find the optimal partition of over $[1:t]$, one has to evaluate the cost of all segments of form $[(s+1){:}t]$ with $s < t$. Among all these $s$, some of them lead to terrible (high cost) partitions. If we could knew them in advance, we would not have to evaluate those $s$. Pruning techniques build upon this

heuristic and allow to evaluate the costs $[(s+1):t]$ on a data-driven subset $R_t$ of possible values for $s < t$. Killick et al [64] have introduced the first such pruning procedure. The general idea is that, if some $s$ enforces a too high cost at time $t$, then $s$ can be pruned for all time $t' > t$. See Section 3.2 for more details. The resulting method, called PELT allows to find exactly a minimal cost partition. The complexity is still quadratic in worst case situations. Nevertheless, it is proved to be almost linear in scenarios where the true distribution contains many true change-points. Under stronger assumptions on the cost function Rigaill [103] and Maidstone et al [83] have introduced functional pruning methods, which are at least as fast as PELT algorithm. As these methods are more involved, we leave their extension to the tree case for future work.

**Penalization and selection of the size of the partition.** If one follows Auger and Lawrence' approach, where each minimal cost partition $\mathbf{P}_k^*$ is computed for $k \leq K$, it is still needed to select a size $\widehat{k}$ to obtain a proper estimator of $\underline{\mathbf{P}}$. If one follows Jackson et al.' [55]'s approach, one already gets a single estimator $\mathbf{P}^*$ by minimizing the cost function, but this cost function has to already include a linear penalty term, otherwise $\mathbf{P}^*$ would be the trivial partition with subtrees of size 1. In both situations, it is therefore required, at least implicitly, to select the size of the partition. One classical approach to achieve this goal is through penalization. Given a penalty function $\mathrm{pen} : \mathcal{P} \to \mathbb{R}^+$, this amounts to solving the following minimization problem.

$$\widehat{\mathbf{P}} \in \arg\min_{\mathbf{P}} \left[ \mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) + \mathrm{pen}(\mathbf{P}) \right] . \tag{3.8}$$

Suppose that the penalty function $\mathrm{pen}(\mathbf{P})$ only depends on the size $|\mathbf{P}|$ of the partition, that is $\mathrm{pen}(\mathbf{P}) = f(|\mathbf{P}|)$. Then, one can follow Auger and Lawrence approach by computing $\mathbf{P}_k^*$ for each $1 \leq k \leq n$ and then choosing $\widehat{\mathbf{P}} = \mathbf{P}_{\widehat{k}}^*$ with $\widehat{k} \in \arg\min_k \mathrm{Cost}_{\mathrm{T}}(\mathbf{P}_k^*) + f(k)$ (see for instance [71]). If the penalty function (3.8) is linear in the sense, that for $\mathbf{P} = (S_1, \ldots, S_q)$, we have $\mathrm{pen}(\mathbf{P}) = \sum_{i=1}^q g(S_i, \mathbf{y}_{S_i})$ for some function $g$, then the penalty can be included into the cost function and (3.8) can be solved using Jackson et al's algorithm [55].

In the case of univariate ($\mathcal{Y} = \mathbb{R}$) time series on the chain graph, there is large body of work on the choice of the penalty function. When the $(y_s)$'s follow a Gaussian distribution with common variance $\sigma^2$ it has been explained in Lebarbier [71] that penalizing the least-square cost with a penalty function $\mathrm{pen}(\mathbf{P}) = \sigma^2 \left[ c_1 |\mathbf{P}| + c_2 \log(n/|\mathbf{P}|) \right]$ with suitable constants $c_1$ and $c_2$ allows to estimate near optimally the mean vector of $(y_s)$, thereby certainly recovering a good estimator of the partition. Extensions of this penalty to unknown variance $\sigma^2$ and more generally to exponential distributions have been addressed in [71] and [22]. The same kind of results have been obtained by [4, 40] for Kernel change point detection. This penalty choice only depends on $|\mathbf{P}|$, but it is not linear and cannot therefore be included in the cost function as in Jackson et al. [55]. Nevertheless, approximation schemes have been proposed by Killick et al. [64]. See also Section 3.4 for more details.

In an univariate subGaussian model with common variance, Wang et al. [128] have proposed a BIC like penalty $\mathrm{pen}(\mathbf{P}) = c_1 \sigma^2 |\mathbf{P}| \log(n)$ with a suitable constant $c_1$. Assuming that all changes in the mean are large enough, they proved that the corresponding estimator $\widehat{\mathbf{P}}$ has same number of segments as the true partition $\underline{\mathbf{P}}$ and location of each change point in $\widehat{\mathbf{P}}$ is close to those of $\underline{\mathbf{P}}$.

### Other Change-point detection methods on the chain graph

Change-point detection methods on the chain graph are not restricted to cost minimization procedures. See [121] for a recent review on the topic. Let us simply discuss two popular approaches.

**Binary Segmentation (BS).** Binary segmentation is a greedy dichotomous procedure first introduced in [124]. It amounts to recursively cut the chain graph into two segments by minimizing a criterion such as the CUSUM statistic [39]. It is usually applied in mean-change problems (3.1) although extensions to changes in covariance have been proposed [127]. A stopping criterion allows

to halt the algorithm when no statistically significant breaking edge is detected. Binary segmentation enjoys a nice quasi-linear complexity $O(n\log(n)$ in most cases. Unfortunately, it is proved to be inconsistent unless with very restrictive hypothesis on the location of the breaks. This lead Fryzlewicz [39] to propose a modifications WBS which turns out to be consistent.

**Fused Lasso.** For mean-change (3.1) univariate data on the chain graph, Tibshirani et al [118] have proposed the fused Lasso procedure, which amounts to minimize the following penalized criterion

$$\widehat{\theta}_\lambda = \arg\min_{\theta\in\mathbb{R}^n} \sum_{i=1}^{n}(y_i-\theta_i)^2 + \lambda\sum_{i=1}^{n-1}|\theta_{i+1}-\theta_i| \ , \tag{3.9}$$

where $\lambda>0$ is a tuning parameter. Then, a segmentation $\widehat{\mathbf{P}}_\lambda$ is simply given by the break edges $\widehat{\theta}_\lambda$. The fused Lasso criterion is convex and is provably computed in quasi linear time. The penalization $|\theta_{i+1}-\theta_i|$ in (3.9) forces the differences $|\widehat{\theta}_{i+1}-\widehat{\theta}_i|$ to be sparse. In other words, it enforces the vector $\widehat{\theta}$ to be piecewise constant. Extensions to exponential family models were later proposed [77]. If the true break edges are regularly spaced on the chain, Fused Lasso is able to recover them [47] at the near optimal rate. Unfortunately, when the break edges are unequally spaced, it behaves badly compared to penalized least-squares procedures [34].

#### Segmentation over general graphs

In many important problems, including image and video analysis, data $(y_s)$ are indexed by a general graph which is not tree structured. Typical structures in image analysis includes the 2 and 3 dimensional grids. Unfortunately, it is not possible to extend our PELT Tree algorithm in this setting. More generally, all known exact cost minimization procedures suffer from an exponential complexity for general graphs. In the specific case of mean changes (3.1), it is possible to extend the Fused Lasso estimator to this setting by introducing the penalty $\sum_{\{i,j\}\in\mathcal{E}}|\theta_i-\theta_j|$ . This method is sometimes referred as total variation denoising [53] or trend filtering on graph [131]. The corresponding criterion is still convex and can be minimized efficiently. In the specific case of univariate data ($p=1$) in mean-change model (3.1), approximation schemes have been proposed to minimize the least-squares cost penalized by the number of break edges [34]. However, this approach does not easily extend to more general settings (e.g. non univariate data). In summary, segmentation over general graphs seems, at least from an algorithmic point of view, much more challenging than the tree case considered here.

### 3.1.5 Organization

In Section 3.2, we remind the reader of the dynamic programming algorithm on the chain graph and the pruning refinement PELT of Killick et al. [64]. Section 3.3 extends both the dynamic algorithm and PELT on a general tree. In Section 3.4, we assume that the data $(y_s)$ follow a Gaussian model (3.1) and propose a penalty function so that the resulting estimator achieves an oracle inequality. Numerical illustrations are given in Section 3.5, whereas Section 3.6 is dedicated to the application to Fish abundance data. Some of the proofs are postponed to the appendix.

## 3.2 Cost minimization on the chain graph

In this section, we consider the cost minimization problem on the chain graph of size $n$. We shall remind the reader on the dynamic programming algorithm and of the recent refinement of Killick et al. [64]. Let us introduce some notation. For any integers $t\leq s$, $[t{:}s]$ stands for the set $\{t,t+1,\ldots,s\}$, whereas $(t{:}s]$ is short for the set $\{t+1,t+2,\ldots,s\}$. If $s=t$, we take the convention $(t,s]=\emptyset$. We also write $[1:n]$ for the chain graph of size $n$. Our objective is to find a partition $\mathbf{P}^*$ of the chain graph with minimal cost

$$\mathbf{P}^* \in \underset{\mathbf{P}\in\mathcal{P}([1:n])}{\arg\min}\ \mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) \ ,$$

Any partition $\mathbf{P} \in \mathcal{P}([1{:}n])$ is of the form $\{[1{:}\tau_1], (\tau_1{:}\tau_2], \ldots, (\tau_{k-1}{:}\tau_k]\}$ with $\tau_k = n$. For that partition, the corresponding change-points are $\tau_1, \ldots, \tau_{k-1}$. In the specific case where $k = 1$, there is no change point. In this section, we shall alternate between the representation of a segmentation in terms of its change-points or in terms of the partition. The former being more natural for the chain graph and the latter being more suited to general trees. This will serve as a gentle introduction to the case of general trees considered in the next section.

### 3.2.1 Optimal Partition (Dynamic programming over the chain graph)

We describe some key ideas of the dynamic programming algorithm on the chain graph, also called Optimal Partition algorithm [55]. For any integer $1 \leq s \leq n$, define

$$F(s) = \min_{\mathbf{P} \in \mathcal{P}([1:s])} \mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) \ , \tag{3.10}$$

the minimal cost of a partition of the chain graph $[1{:}s]$. Thus, $F(n)$ is the minimal cost for the whole chain graph. We use the convention $F(0) = 0$ and $\mathcal{P}(\emptyset) = \{\emptyset\}$.

The number of partitions of $[1{:}n]$ into segments, that is the size of $\mathcal{P}([1{:}n])$, is $2^{n-1}$, so that a naive algorithm would require an exponential number of operations to compute $F(n)$. The fundamental property that underlines the dynamic programming algorithm is that $F(n)$ can be easily computed given the previous values $F(s)$ for $1 \leq s < n$.

**Lemma 12.** *For any $s = 1, \ldots, n$, we have*

$$F(s) = \min_{0 \leq \tau \leq s-1} \left[ F(\tau) + \mathrm{C}_{[(\tau+1):s]} \right] \ . \tag{3.11}$$

We may interpret the index $\tau_s^*$ achieving the minimum of (3.11) as the last change-point of a minimum cost partition over $[1{:}s]$. Although this lemma is well-known and is a variation of Bellman's seminal work [10], we still provide its proof as a warm-up for the tree case.

*Proof of Lemma 12.* Write $G(s)$ for the right-hand side of (3.11). Consider any partition

$$\mathbf{P} = \{[1{:}j_1], (j_1{:}j_2], \ldots, (j_k{:}s]\}$$

of $[1{:}s]$. Then, the first $k$ segments form a partition $\mathbf{P}_1$ of $[1{:}j_k]$. By decomposing the $\mathrm{Cost}_{\mathrm{T}}$ function into a sum of elementary costs, we have

$$\mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) = \mathrm{Cost}_{\mathrm{T}}(\mathbf{P}_1) + C_{[(j_k+1):s]} \geq F(j_k) + C_{[(j_k+1):s]} \geq G(s) \ ,$$

by definition of $F(j_k)$. Taking the minimum over all partitions $\mathbf{P}$, we obtain that $F(s) \geq G(s)$. Conversely, take $\tau'$ achieving the minimum in $G(s)$ and consider a partition $\mathbf{P}'$ of $[1{:}\tau']$ achieving $F(\tau')$. Then, $\mathbf{P} = \mathbf{P}' \cup \{[\tau'+1{:}s]\}$ is a partition of $[1{:}s]$ whose cost is equal to $G(s)$. As a consequence, $F(s) \leq \mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) = G(s)$. The result follows. □

The following Algorithm (aka Optimal Partitionning in [55]) relies on the recursive formula (3.11) to compute $F(n)$. Along the algorithm, we store $\tau_s^*$ an index $\tau$ achieving the minimum in the right-hand in (3.11), so that it is possible to recover an optimal partition $\mathbf{P}^*$ (or more precisely the break points of $\mathbf{P}^*$) by a simple backtracking step. Let us briefly explain this backtracking step. For the sake of the discussion, let us assume that the minimum cost partition $\mathbf{P}^*$ is unique. In view of the proof of Lemma 12, $\mathbf{P}^*$ is shown to be the concatenation of $(\tau_n^*{:}n]$ and the optimal partition of $[1{:}\tau_n^*]$. In turn, the optimal partition of $[1{:}\tau_n^*]$ contains the set $[(\tau_{\tau_n^*}^* + 1){:}\tau_n^*]$. Iterating the procedure allows us to reconstruct the partition.

**Algorithm 1** Dynamic Programming Algorithm for the chain graph.

**Input :** T, $(y_t)_{t\in\mathcal{V}(\mathrm{T})}$, $\beta$.
Set $F(0) = 0$
**for** $s$ in (1:$n$) **do**
  Compute $F(s) = \min_{0\leq\tau\leq t-1}\left[F(\tau) + \mathrm{C}_{[(\tau+1):s]}\right]$
  Compute $\tau_s^* \in \arg\min_{0\leq\tau\leq t-1}\left[F(\tau) + \mathrm{C}_{[(\tau+1):s]}\right]$
**end for**
Set $J^* = \emptyset$; Set $s = \tau_n^*$
**while** $s > 0$ **do**
  Set $J^* = \{t\} \cup J^*$
  Set $s = \tau_s^*$
**end while**
**return** $(F(n), J^*)$

Algorithm 1 outputs both the minimum cost $F(n)$ and the corresponding set $J^* = \{j_1, \dots j_{|J^*|}\}$ of break points. The associated partition $\mathbf{P}^*$ has then a size $|J^*| + 1$ and is defined

$$\mathbf{P}^* = \{[1{:}j_1], (j_1{:}j_2], \dots, (j_{|J^*|}, n]\} \ .$$

In Algorithm 1, the computation of the vector $\tau^*$ and of all $F(s)$ requires $O(n^2)$ operations (assuming the computation of the cost of a segment is constant). The backtracking step is linear.

### 3.2.2 PELT algorithm [64]

The quadratic complexity of Algorithm 1 can be prohibitive in some applications such as finance or genomics where the size $n$ of the chain is very large. For this reason, Killick et al. [64] introduced a new procedure (called PELT), which in some cases, achieves a near-linear time complexity. The idea underlying PELT is that, in Equation (3.11), it is perhaps not necessary to compute the minimum of $F(\tau) + \mathrm{C}_{[(\tau+1):s]}$ over **all** $\tau$. Indeed, if we could know in advance that, for some $\tau < s$, the sum $F(\tau) + \mathrm{C}_{[(\tau+1):s]}$ is much larger than $F(s)$, then we could leave those $\tau$ aside.

In this subsection, we shall assume that the cost function $C$ satisfies a sub-additive property. More precisely, we assume that there exists $\kappa \in \mathbb{R}$ such that, for any $s < \tau \leq t$,

$$C_{[s:\tau]} + C_{[(\tau+1):s]} + \kappa \leq C_{[t:s]} \ . \tag{3.12}$$

As argued in [64], this assumption is mild. If the cost function $C_{[t:s]}$ is the negative maximum log-likelihood of the data $(y_t, \dots, y_s)$ in some parametric model, then Assumption (3.12) is satisfied with $\kappa = 0$. If the cost function function $C_{[t:s]}$ is the negative maximum log-likelihood penalized by $\beta$, then Assumption (3.12) holds with $\kappa = -\beta$.

**Lemma 13** ([64]). *Assume that (3.12) is satisfied. If for some $\tau < t$, we have*

$$F(\tau) + C_{[(\tau+1):t]} + \kappa > F(t) \ , \tag{3.13}$$

*then, for any $s > t$, we have*

$$F(s) < F(\tau) + C_{[(\tau+1):s]} \ .$$

*In other words, $\tau$ does not achieve the minimum in the recursions (3.11), this for all $s > t$.*

One may interpret Lemma 13 as follows. If any partition of [1:$t$] having $\tau$ as its last change-point has a too large cost (in the sense of (3.13)), then $\tau$ cannot be the last-change point of any optimal partition of [1:$s$] for all $s > t$. Thus, under (3.13), it is not necessary to consider anymore the index $\tau$ in the recursion step of the dynamic programming algorithm.

*Proof of Lemma 13.* Consider any triplet $\tau < t < s$ such that (3.13) is satisfied. Then, it follows from Assumption (3.12) that

$$\begin{aligned} F(\tau) + C_{[(\tau+1):s]} &\geq F(\tau) + C_{[(\tau+1):t]} + C_{[(t+1):s]} + \kappa \\ &\overset{\text{by (3.13)}}{>} F(t) + C_{[(t+1):s]} \ , \end{aligned}$$

the latter being greater or equal to $F(s)$ by the recursion equation (3.11). The result follows. □

Relying on the above Lemma, Killick et al. introduced PELT algorithm described below. Here, $R_s$ corresponds to the set of values $\tau < s$ that have not been pruned by the condition (3.13).

**Algorithm 2** PELT

Set $F(0) = 0$ and $R_1 = \{0\}$
**for** $s$ in $(1{:}n)$ **do**
  Compute $F(s) = \min_{\tau \in R_s} \left[F(\tau) + \mathrm{C}_{[(\tau+1):s]}\right]$
  Compute $\tau_s^* \in \arg\min_{\tau \in R_s} \left[F(\tau) + \mathrm{C}_{[(\tau+1):s]}\right]$
  Set $R_{s+1} = \{\tau \in R_s \colon F(\tau) + C_{[(\tau+1):s]} + \kappa \leq F(s)\} \cup \{s\}$
**end for**
Set $J^* = \emptyset$; Set $s = \tau_n^*$
**while** $s > 0$ **do**
  Set $J^* = \{s\} \cup J^*$
  Set $s = \tau_s^*$
**end while**
**return** $(F(n), J^*)$

The computational cost of PELT is roughly of the order $\sum_{s=1}^{n} |R_s|$. As a consequence, if the pruning condition (3.13) is never satisfied, that is if $R_s = \{1, \ldots, s-1\}$, then the complexity is the same as for the vanilla dynamic programming algorithm. Conversely, if most of the jumps $\tau$ are pruned so that $|R_s|$ is of constant order, then the time complexity of PELT is linear. For instance, Killick et al. have proved that, when the cost is the penalized negative log-likelihood in a parametric exponential model and when the number of jumps is linear in $n$, then PELT has a linear complexity. See [64] for more details.

## 3.3 Minimizing the cost on a general tree

Turning to the problem of segmentation on general trees, we now consider the cost minimization problem (3.6) $\arg\min_{\mathbf{P} \in \mathcal{P}(\mathrm{T})} \mathrm{Cost}_{\mathrm{T}}(\mathbf{P})$ on a general tree T.

We start by extending the Dynamic programming algorithm to general trees. The main difference with the chain graph case is that we will not iterate over the last change-point, but over the current connected component of the partition. In other words, we shall rather work with the partition of the nodes into connected components than with the break edges. As explained in the introduction, a partition is uniquely characterized by its break edges, so this change of viewpoint does not change the problem.

Another difference with the previous section is that the tree T is not naturally endowed with a complete ordering of the nodes. We shall partially fix this difference in Section 3.3.1 by introducing a partial ordering on the nodes.

As explained below, the vanilla Dynamic Programming algorithm introduced in Section 3.3.2 has an exponential time complexity. This makes it all the more important to introduce a pruning method. PELT Tree extension is described in Section 3.3.3. In the specific case of the penalized least-squares cost function, we introduce in Section 3.3.4 a refinement of this algorithm to lower its space complexity.

### 3.3.1 Preliminary: ordering the Tree

To introduce a partial ordering '$\preceq$' on the nodes of T, we assume that the tree T is rooted at a some node $o$. If it is not the case, we can root it at an arbitrary node. This allows us to transform T in a rooted directed tree where the edges are now directed so that an edge $\{s,t\}$ is directed from $s$ to $t$ if $s$ is closer to the origin than $t$[1]. Then, for two nodes $s$ and $t$, we say $s \prec t$ if there exists a directed path from $s$ to $t$, which means that $t$ is a descendant of $s$. This defines a partial ordering on the set of nodes. The minimal node of T is the origin $o$, whereas the maximal nodes of T are the leaves of T. We say that $t$ is a child of $s$, if there exists a directed edge from $s$ to $t$.

In the sequel, we denote $\mathrm{ch}(s)$ for the collection of children of a node $s$, whereas $\mathrm{D}(s)$ stands for the collections of strict descendants of a node $s$.

$$\mathrm{D}(s) = \{t \in \mathrm{T} \,|\, s \prec t\} \ . \tag{3.14}$$

Let us introduce some further notation. Given a node $s$, $\mathrm{T}_s$ stands for the induced sub-tree of T rooted at $s$ and made of all the descendants of $s$. In other words, the node set of $\mathrm{T}_s$ is $\{s\} \cup D(s)$. Given two sub-forests $S_1$ and $S_2$ of T, we define the sub-forests $S_1 \cap S_2$, $S_1 \cup S_2$, and $S_1 \setminus S_2$ induced by the corresponding set of nodes.

### 3.3.2 Dynamic Programming for trees

Let us now extend the dynamic programming algorithm to any rooted tree T. Given any sub-tree $S$ of T, we define $F(S)$ as the minimal cost of all partitions $\mathbf{P}$ of $S$

$$F(S) = \min_{\mathbf{P} \in \mathcal{P}(S)} \mathrm{Cost}_S(\mathbf{P}) \ .$$

More generally, if $R$ is a sub-forest of T (i.e. $R$ is not necessarily connected), we write $\mathcal{CC}(R) = (S_1, \ldots, S_r)$ for the collection of its connected components. The optimal cost over the forest $R$ is defined as

$$F(R) = \sum_{S \in \mathcal{CC}(R)} F(S) \ . \tag{3.15}$$

The fundamental property that allows to extend the dynamic algorithm to general trees is given below. Given any rooted tree T, denote $\mathcal{S}(\mathrm{T})$ the collection of sub-trees of T that **contain the root**.

**Lemma 14.** *For any rooted tree* T*, we have*

$$F(\mathrm{T}) = \min_{S \in \mathcal{S}(\mathrm{T})} \left[ F(\mathrm{T} \setminus S) + \mathrm{C}_S \right] \ . \tag{3.16}$$

*Proof of Lemma 14.* Consider any partition $\mathbf{P}$ of T and pick the sub-tree $S$ in this partition that contains the root. Then,

$$\mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) = \sum_{S' \in \mathbf{P}} C_{S'} = C_S + \sum_{S' \in \mathbf{P} \setminus \{S\}} C_S$$

Then, $\mathbf{P} \setminus \{S\}$ is a partition of $\mathrm{T} \setminus S$ and we have therefore $\mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) \geq C_S + F(\mathrm{T} \setminus S)$. This implies that $F(\mathrm{T}) \geq \min_{S \in \mathcal{S}(\mathrm{T})} F(\mathrm{T} \setminus S) + \mathrm{C}_S$. Conversely, consider any partition $\mathbf{P}^*$ induced by the minimum of $F(\mathrm{T} \setminus S) + \mathrm{C}_S$. By definition of $F(T)$, we have $F(T) \leq \mathrm{Cost}_{\mathrm{T}}(\mathbf{P}^*) = \min_{S \in \mathcal{S}(\mathrm{T})} F(\mathrm{T} \setminus S) + \mathrm{C}_S$. The result follows. □

[1] We can always direct an edge since, in a tree, there exists a unique path from $o$ to $t$. If this path goes through $s$, then $s$ is closer to the origin, otherwise $t$ is closer to the origin.

Using update rule (3.16) we can propose a simple algorithm finding the minimum cost partition of a tree T. The idea is to browse the tree node by node from leaves to root and to compute $F(\mathrm{T}_s)$ for each $s \in \mathrm{T}$ thanks to (3.16). Then, we run a backtracking routine to compute the optimal partition $\mathbf{P}^*$. First we present the pseudo code for the dynamic programming algorithm, then we describe the backtracking routine. In what follows, we take the convention that $F(\emptyset) = 0$. In a tree, the height of a node $s$ is the distance from $s$ to its farthest descendant. The height of a leaf is 0, the parent of a leaf has height 1 and the root of the tree has maximal height. The height of T is defined as the height of its root.

**Algorithm 3** Dynamic Programming algorithm

**for** $0 \leq h \leq \mathrm{height}(\mathrm{T})$ **do**
  **for** $s$ such that $\mathrm{height}(s) = h$ **do**
    Compute $F(\mathrm{T}_s) = \min_{S \in \mathcal{S}(\mathrm{T}_s)} [F(\mathrm{T}_s \setminus S) + \mathrm{C}_S]$
    $S_s^* = \arg\min_{S \in \mathcal{S}(\mathrm{T}_s)} [F(\mathrm{T}_s \setminus S) + \mathrm{C}_S]$.
  **end for**
**end for**
**Backtracking step**

We claim that the connected components $\mathcal{CC}(T_s \setminus S)$ are sub-trees of the form $\mathrm{T}_t$ for some $t$ descendants of $s$. Indeed, consider any $S' \in \mathcal{CC}(T_i \setminus S)$. If $t$ belongs to $S'$, then all its descendants $\mathrm{D}(t)$ cannot belong to $S$ since $S$ is connected and the unique path from $s$ to nodes in $\mathrm{D}(t)$ goes through $t$. Thus, $D(t) \cup \{t\} \subset S'$. Take any node $t \in S'$ whose parent $k$ belongs to $S$. Then, $D(t) \cup \{t\} = S'$, otherwise this contradicts that $k \notin S'$. This proves the claim.

As a consequence of this claim, the quantity $F(T_s \setminus S) = \sum_{S' \in \mathcal{CC}(\mathrm{T}_s \setminus S)} F(S')$ in Algorithm 3 is a sum of $F(\mathrm{T}_t)$ for some $t \in \mathrm{D}(s)$. Since we browse the tree from leaves to the root, we already computed all these $F(T_t)$ at the previous steps. At the last step of the first routine, we are at the root $o$ of the tree T and we therefore get the desired minimal cost $F(T)$.

At the last step, the only information we have about the optimal partition of T is its cost $F(\mathrm{T})$ and the sub-tree $S_o^*$ of a corresponding partition that contains the root $o$ of T. In practice, we are also interested in recovering the optimal $\mathbf{P}^*$ segmentation. To do so, we need a backtracking step that is able to retrieve the elements of the partition using the collection of $S_s^*$ for $s \in \mathrm{T}$ stored in the main routine of the algorithm.

The idea is to browse the tree from the root to the leaves. First, one considers $S_o^*$, where $o$ is the root T. From Lemma 14, we deduce that $S_o^*$ is an element of a minimal cost partition of T. We store this first sub-tree in a list denoted $\mathcal{L}_{opt}$. Then, the complementary forest $\mathrm{T} \setminus S_o^*$ of $S_o^*$ is made $q$ sub-trees of T, that are of the form $\mathrm{T}_{s_1}, \ldots, \mathrm{T}_{s_q}$. For $j = 1, \ldots, q$, we add $S_{s_j}^*$ to the list $\mathcal{L}_{opt}$ and then we consider the forests $\mathrm{T}_{s_j} \setminus S_{s_j}^*$ for $j = 1, \ldots, q$. We continue until we have covered the whole tree. At the end, the list $\mathcal{L}_{opt}$ of sub-trees is equal to the optimal partition $\mathbf{P}^*$. We provide below a pseudo-algorithm for the backtracking step.

**Algorithm 4** Backtracking step - Simple Dynamic Programming algorithm

$\mathcal{L}_{opt} = \emptyset$.
$\mathcal{L}_r = \{o\}$
**while** $\mathcal{L}_r \neq \emptyset$ **do**
  Get any $s \in \mathcal{L}_r$.
  Add the sub-trees $S_s^*$ to $\mathcal{L}_{opt}$.
  $\mathcal{L}_r = \mathcal{L}_r \setminus \{s\}$.
  Add to $\mathcal{L}_r$ the roots of the non-empty trees of $\mathcal{CC}(\mathrm{T}_s \setminus S_s^*)$
**end while**

Unfortunately, the naive dynamic programming Algorithm has an exponential time complexity. Indeed, the computation of $F(T_s)$ requires to minimize over the collection $\mathcal{S}(T_s)$ of sub-trees of $\mathrm{T}_s$

that contain $s$. For the chain graph $|\mathcal{S}(T_s)| \leq n$, since a sub-tree of $\mathrm{T}_s$ is segment of the form $[t{:}s]$. However, with a general tree T, $|\mathcal{S}(T_s)|$ can be much larger. Let us for instance compute $u_h = |\mathcal{S}(T_o)|$ for a balanced binary tree with depth $h$. One can show that $u_h$ satisfies the recursion $u_{h+1} = (u_h + 1)^2$. Since $u_2 = 4$ and $u_{h+1} \geq u_h^2$, it follows that $u_h \geq 2^{2^{h-2}}$. Since the number $n$ of nodes of T is $2^{h+1} - 1$, we conclude that computational complexity is higher than $2^{(n+1)/8}$. The complexity is even higher when the nodes of the tree have higher degrees. As a consequence, the dynamic programming algorithm is not feasible for large trees and it is therefore of utmost importance to prune the set $\mathcal{S}(\mathrm{T}_s)$ to make the procedure tractable.

### 3.3.3 PELT Tree Algorithm

We now explain the general principle behind a PELT Tree pruning method. As in the chain graph case, we require that the cost satisfies a sub-additive property. Fix some $\kappa \leq 0$. we assume henceforth that, for any sub-trees $S$, $S_1$, and $S_2$ of T such that $S_1 \cup S_2 = S$ and $S_1 \cap S_2 = \emptyset$, we have

$$\mathrm{C}_{\mathrm{T}} \geq \mathrm{C}_{S_1} + \mathrm{C}_{S_2} + \kappa \ . \tag{3.17}$$

When T is the chain graph, then this property matches Condition (3.13) for PELT, except that we now assume that $\kappa \leq 0$. In particular, when the cost is the minimum negative log-likelihood penalized by $\beta$, Property (3.17) is valid with $\kappa = -\beta$.

**Theorem 1.** *Assume that the cost satisfies (3.17). Consider two nodes $s$ and $t$ in the tree T with $t \in \mathrm{D}(s)$. Let $\mathbf{P}_s$ denote a partition of $\mathrm{T}_s$ and let denote $\mathbf{P}_t$ denote the induced partition of $\mathrm{T}_t$, that is $\mathbf{P}_t = \{S \cap \mathrm{T}_t,\ S \in \mathbf{P}_s\}$. Then, if the cost $\mathbf{P}_t$ is too suboptimal in the sense that*

$$\mathrm{Cost}_{\mathrm{T}_t}(\mathbf{P}_t) + \kappa > F(\mathrm{T}_t) \tag{3.18}$$

*then $\mathrm{Cost}_{\mathrm{T}_s}(\mathbf{P}_s) > F(\mathrm{T}_s)$ which implies that $\mathbf{P}_s$ is not a minimum cost partition of $\mathrm{T}_s$.*

*Proof of Theorem 1.* We denote $\mathbf{P}'$ the partition of $\mathrm{T}_s \setminus \mathrm{T}_t$ stemming from $\mathbf{P}_s$, that is $\mathbf{P}' = \bigcup_{S \in \mathbf{P}_s} \{S \setminus \mathrm{T}_t\}$. If $\mathbf{P}_t$ satisfies equation (3.18), then

$$\mathrm{Cost}_{\mathrm{T}_t}(\mathbf{P}_t) + \mathrm{Cost}_{\mathrm{T}_s \setminus \mathrm{T}_t}(\mathbf{P}') + \kappa > F(\mathrm{T}_t) + \mathrm{Cost}_{\mathrm{T}_s \setminus \mathrm{T}_t}(\mathbf{P}') \geq F(\mathrm{T}_s) \ , \tag{3.19}$$

by (3.16). Since the elements of the partition $\mathbf{P}_s$ are connected components in the tree $\mathrm{T}_s$, at most one element $S$ of $\mathbf{P}_s$ satisfies $S \cap T_t \neq \emptyset$ and $S \nsubseteq T_t$. Indeed, this connected component contains a path from $\mathrm{T}_t$ to $\mathrm{T}_s \setminus T_t$ and must therefore contain the root $t$ of $\mathrm{T}_t$. It is therefore unique if it exists.

If there no such $S$,then $\mathbf{P}_s = \mathbf{P}_t \cup \mathbf{P}'$ and we have $\mathrm{Cost}_{\mathrm{T}_t}(\mathbf{P}_t) + \mathrm{Cost}_{\mathrm{T}_s \setminus \mathrm{T}_t}(\mathbf{P}') = \mathrm{Cost}_{\mathrm{T}_s}(\mathbf{P}_t)$. Since $\kappa \leq 0$, it follows from (3.19) that $\mathrm{Cost}_{\mathrm{T}_s}(\mathbf{P}_s) > F(T_s)$ and $\mathbf{P}_s$ is therefore suboptimal. If there is one such $S$, we have by Property (3.17) that

$$\mathrm{Cost}_{\mathrm{T}_t}(\mathbf{P}_t) + \mathrm{Cost}_{\mathrm{T}_s \setminus \mathrm{T}_t}(\mathbf{P}') + \kappa = C_{S \cap T_t} + C_{S \setminus T_t} + \kappa + \sum_{S' \in \mathbf{P}_j \cup \mathbf{P}' :\ S' \cap S = \emptyset} C_{S'} \leq \sum_{S' \in \mathbf{P}_s} C_{S'} = \mathrm{Cost}_{\mathrm{T}_s}(\mathbf{P}_s) \ ,$$

which implies again that $\mathrm{Cost}_{\mathrm{T}_s}(\mathbf{P}_s) > F(\mathrm{T}_s)$. □

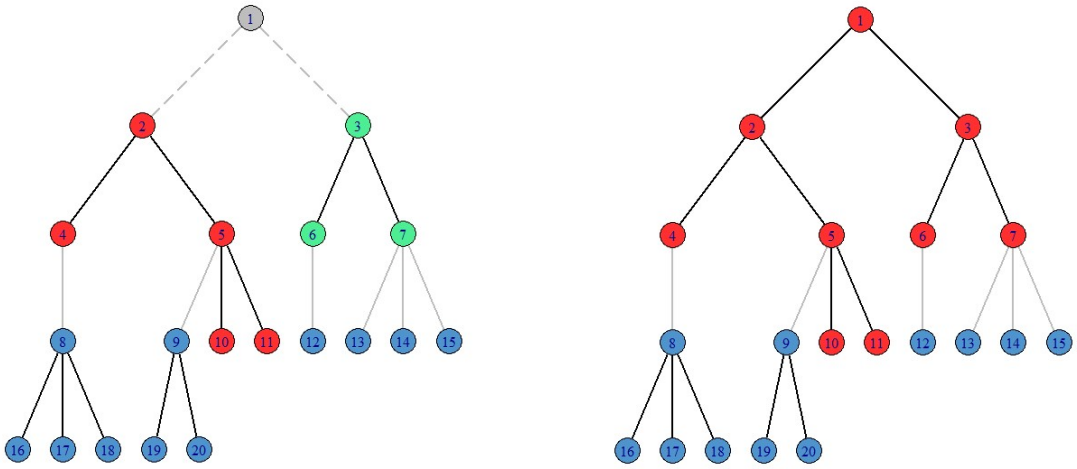
An important consequence of Theorem 1 is the following. Consider a connected component $S \in \mathcal{S}(T_t)$. If

$$C_S + F(T_t \setminus S) + \kappa > F(T_t) \tag{3.20}$$

Then, for any ancestor $s$ of $t$, no optimal partition of $\mathrm{T}_s$ contains a connected component of the form $S \cup R$ where $R \cap T_t = \emptyset$. This also implies that, in the dynamic programming algorithm, it is not necessary consider any subset of this form $S \cup R$ in the minimization over $\mathcal{S}(\mathrm{T}_s)$.

This condition (3.20) is at the heart of our pruning step and will allow us to recursively build a pruned subset $\mathcal{S}_{\mathrm{pr}}(T_s)$ of $\mathcal{S}(T_s)$. Assume that we are at step $s$. One can show that the non-pruned

collection of $\mathcal{S}(T_s)$ is made of connected components $S$ of the form $S = \cup_{t\in \mathrm{ch}(s)} S_t \cup \{s\}$, where $S_t \in \mathcal{S}(T_t) \cup \{\emptyset\}$. To see this, we may observe that $S_t$ is defined as $S \cap T_t$.



(a) (b)

Figure 3.2 – This figure illustrates that a last sub-tree of $\mathrm{T}_1$ is the union of the node 1 and last sub-trees of $\mathrm{T}_2$ and $T_3$. In (a), the red nodes are the nodes of a last sub-tree of $\mathrm{T}_2$ . The green nodes are the nodes of a last sub-tree of $\mathrm{T}_3$. The node 1 is grey and the edges connecting it to other nodes are dotted because we do not take it into account in this figure. Other nodes are represented in blue. In (b) the red nodes are the nodes of the last sub-tree of $\mathrm{T}_1$ made of the red and green sub-trees of figure (a) plus the node 1. In both (a) and (b) the plain and grey edges represent the breaks between the last sub-trees and the rest of the tree while the plain black edges represent edges within sub-trees of $\mathrm{T}_1$, $\mathrm{T}_2$ or $\mathrm{T}_3$.

In the pruned algorithm, we shall use a similar construction, except that we use non pruned sub-trees $S_t$ in $\mathcal{S}_{\mathrm{pr}}(T_t) \cup \{\emptyset\}$. In other words, we shall consider

$$\mathcal{S}_{\mathrm{pr},1}(T_s) := \left\{ S = \cup_{t\in \mathrm{ch}(s)} S_t \cup \{s\} \ , \quad S_t \in \mathcal{S}_{\mathrm{pr}}(T_t) \cup \{\emptyset\} \right\} \ . \tag{3.21}$$

In the special case where $s$ is a leaf, (3.21) entails that $\mathcal{S}_{\mathrm{pr},1}(T_s) = \{\{s\}\}$. Then, we claim that minimizing over this pruned set $\mathcal{S}_{\mathrm{pr},1}(T_s)$ still allows to find the minimum $F(T_s)$.

$$F(T_s) = \min_{S\in \mathcal{S}_{\mathrm{pr},1}(T_s)} \left[ F(\mathrm{T}_s \setminus S) + \mathrm{C}_S \right] \ . \tag{3.22}$$

Afterwards, we remove from $\mathcal{S}_{\mathrm{pr},1}(T_s)$ all sub-trees $S$ that satisfy the sub-optimality condition corresponding to (3.20). In other words, we define

$$\mathcal{S}_{\mathrm{pr}}(T_s) := \{ S \in \mathcal{S}_{\mathrm{pr},1}(T_s) \quad \text{s.t. } C_S + F(T_s \setminus S) + \kappa \leq F(T_s) \} \ . \tag{3.23}$$

To justify our pruning procedure, it suffices to prove the claim (3.22).

*Proof of* (3.22). In fact, we prove that any pruned set $S \in \mathcal{S}(T_s) \setminus \mathcal{S}_{\mathrm{pr},1}(T_s)$ is suboptimal in the sense $F(\mathrm{T}_s \setminus S) + \mathrm{C}_S > F(T_s)$. Consider any such pruned set $S$. For any descendant $t$ of $s$ in $S$, denote $S^{(t)}$ the restriction of $S$ to $\mathrm{T}_t$. Let $(s, s_1, s_2, \ldots, s_r)$ be any path from $s$ to a leaf of $S$. Let $t$ be the deepest node in $S$ such that $S^{(t)}$ is a pruned set of $T^{(t)}$. This implies that the $S^{(k)}$'s for $k \in \mathrm{ch}(t) \cap S$ are not pruned and therefore belong to $\mathcal{S}_{\mathrm{pr}}(T_k)$. As a consequence, we have $S^{(t)} \in \mathcal{S}_{\mathrm{pr},1}(T_t)$. Since $S^{(t)}$ has been pruned, we have by (3.23) that $C_{S^{(t)}} + F(T_t \setminus S^{(t)}) + \kappa > F(T_t)$. By Theorem 1 and its remark below, this implies that $C_S + F(T_s \setminus S) > F(T_s)$ which conclude the proof. □

For completeness, we provide below PELT Tree Algorithm. In this algorithm, we take again the convention $F(\emptyset) = 0$.

**Algorithm 5** PELT Tree algorithm

**for** $0 \leq h \leq \text{height}(\mathrm{T})$ **do**
  **for** $s$ such that $\text{height}(i) = h$ **do**
    Compute $\mathcal{S}_{\text{pr},1}(T_s)$ from $\mathcal{S}_{\text{pr}}(T_t)$ with $t \in \text{ch}(s)$ using (3.21).
    Compute $F(T_s) = \min_{S \in \mathcal{S}_{\text{pr},1}(T_s)} [F(\mathrm{T}_s \setminus S) + \mathrm{C}_S]$.
    Set $S_s^* \in \arg\min_{S \in \mathcal{S}_{\text{pr},1}(T_s)} [F(\mathrm{T}_s \setminus S) + \mathrm{C}_S]$ .
    Prune the set $\mathcal{S}_{\text{pr},1}(T_s)$ to obtain $\mathcal{S}_{\text{pr}}(T_s)$ as in (3.23).
  **end for**
**end for**
**Backtracking step**

The backtracking step is the same as in the previous subsection. The computational complexity of the procedure is of the order of $\sum_{s \in \mathrm{T}} |\mathcal{S}_{\text{pr},1}(T_s)|$. As a consequence, if the subsets $|\mathcal{S}_{\text{pr},1}(T_s)|$ are pruned enough, then this will drastically reduce the complexity. Our numerical experiments in Section 3.5 suggest that, in typical settings, PELT Tree achieves a quadratic complexity instead of the exponential complexity for the vanilla dynamic programming algorithm.

### 3.3.4 Improvement for Penalized least-square cost

In the previous subsection, we explained how to adapt the dynamic programming and the different pruning methods to general trees. Even if we expect PELT Tree to have a reasonable time complexity, we still have to compute and store all the collections $\mathcal{S}_{\text{pr}}(T_s)$. Indeed, in the recursive equation (3.22), we need to compute all costs $C_S$ for $S \in \mathcal{S}_{\text{pr},1}(T_s)$.

In this subsection, we explain how to drastically lower the memory cost of the procedure when the cost is the penalized least-squares one. We assume here that observations $y_s$ in $\mathbb{R}^p$. For a connected component $S$, the cost $C_S$ satisfies $C_S = \sum_{s \in S} \|y_s - \bar{y}_S\|^2 + \beta$ where $\bar{y}_S = \frac{1}{|S|} \sum_{s \in S} y_s$.

Consider any $S \in \mathcal{S}_{\text{pr},1}(T_s)$. From the construction (3.21) of this set $\mathcal{S}_{\text{pr},1}(T_s)$, we have $S = \cup_{t \in \text{ch}(s)} S_t \cup \{s\}$ for some $S_t \in \mathcal{S}_{\text{pr}}(T_t) \cup \{\emptyset\}$. In (3.22), we have to evaluate

$$F(\mathrm{T}_s \setminus S) + \sum_{t \in S} \|y_t - \bar{y}_S\|^2 + \beta \ . \tag{3.24}$$

First, we observe that $F(\mathrm{T}_s \setminus S)$ is the sum of optimal cost for some sub-trees of $\mathrm{T}_s$. In fact, we have

$$F(\mathrm{T}_s \setminus S) = \sum_{t \in \text{ch}(s)} F(\mathrm{T}_t \setminus S_t) \ . \tag{3.25}$$

Therefore, if for all $t \in \text{ch}(s)$ we stored $F(\mathrm{T}_t \setminus S_t)$ for each $S_t \in \mathcal{S}_{\text{pr}}(\mathrm{T}_t) \cup \{\emptyset\}$, we are able to compute $F(\mathrm{T}_s \setminus S)$ easily using (3.25). Second, the cost $C_S$ in (3.24) is only a function of $\theta_S = \sum_{t \in S} y_t$, $\kappa_S = \sum_{t \in S} \|y_t\|^2$ and $|S|$. Interestingly, these quantities are easily updated on the fly. Indeed, we have

$$\theta_S = y_s + \sum_{t \in \text{ch}(s)} \theta_{S_t} \ ; \quad \kappa_S = \|y_s\|^2 + \sum_{t \in \text{ch}(s)} \kappa_{S_t} \ ; \quad |S| = 1 + \sum_{t \in \text{ch}(s)} |S_t| \ . \tag{3.26}$$

From both remarks, we deduce that, in order to compute $F(\mathrm{T}_s)$ at step $s$, we only need to store the following information for all $S \in \mathcal{S}_{\text{pr},1}(\mathrm{T}_s)$ :

$$\mathcal{I}_S(\mathrm{T}_s) = \{F(\mathrm{T}_s \setminus S), \theta_S, \kappa_S, |S|\} \ , \tag{3.27}$$

with the convention $\mathcal{I}_\emptyset(\mathrm{T}_s) = \{F(\mathrm{T}_s), 0, 0, 0\}$. In what follows, we denote $\mathcal{I}(\mathrm{T}_s) = \{\mathcal{I}_S(\mathrm{T}_s) | S \in \mathcal{S}_{\text{pr}}(\mathrm{T}_s) \cup \{\emptyset\}\}$. From the recursive equation (3.25) and (3.26), we obverse that $\mathcal{I}(T_s)$ is easily computed from the $\mathcal{I}(T_t)$'s with $j \in \text{ch}(s)$.

Still, a naive implementation of the algorithm with $\mathcal{I}_S$ requires to store all sub-trees $S \in \mathcal{S}_{\mathrm{pr},1}(\mathrm{T}_s)$. This turns out to be unnecessary. Indeed, the computation of $\mathcal{I}_S(T_s)$ only requires the knowledge of $\mathcal{I}_{S_t}(T_t)$ for $t \in \mathrm{ch}(s)$, whereas the knowledge of the topology of $S$ does not matter. This allows us to introduce the following new representation of $\mathcal{I}(T_s)$. Now a set $S \in \mathcal{S}_{\mathrm{pr},1}(\mathrm{T}_t)$ is only defined through its index $q$. For $q = 1, \dots, |\mathcal{S}_{\mathrm{pr},1}(\mathrm{T}_s)|$, we now consider

$$\mathcal{I}_q(\mathrm{T}_s) = \{F_q, \theta_q, \kappa_q, N_q, \mathbf{b}^q, \mathbf{id}^q\} \ , \tag{3.28}$$

where $N_q$ stands for the size of the corresponding $S$, $\mathbf{b}^q \in \{0,1\}^{|\,\mathrm{ch}(s)|}$ is such that, for a child $t$ of $s$, $\mathbf{b}^q_t = 0$ if $S_t = \emptyset$, that is if $t \notin S$. The second $|\,\mathrm{ch}(s)|$-vector $\mathbf{id}^q$ contains the indices $q_j$ corresponding to $S_t$ in $\mathcal{I}(T_t)$. Equipped with this new notation, we deduce that the recursive equation (3.22) becomes

$$F(\mathrm{T}_s) = \min_{1 \le q \le |\mathcal{I}(\mathrm{T}_s)|} \left[ F_q + \kappa_q - \frac{\|\theta_q\|^2}{N_q} + \beta \right] \ . \tag{3.29}$$

In summary, the update formula for $\mathcal{I}_q(\mathrm{T}_s)$ is the following

$$\mathcal{I}_q(\mathrm{T}_s) = \left\{ F_q = \sum_{t \in \mathrm{ch}(s)} F_{\mathbf{id}^q_t},\ \theta_q = y_s + \sum_{t \in \mathrm{ch}(s)} \theta_{\mathbf{id}^q_t},\ \kappa_q = \|y_s\|^2 + \sum_{t \in \mathrm{ch}(s)} \kappa_{\mathbf{id}^q_t}, N_q = 1 + \sum_{t \in \mathrm{ch}(s)} N_{q_t}, \mathbf{b}^q, \mathbf{id}^q \right\} . \tag{3.30}$$

To sum up, we arrive at the following algorithm. Below, we use the convention $\sum_{t \in \emptyset} a_t = 0$.

**Algorithm 6** PELT Tree for least squares cost

$\mathcal{I}(\emptyset) = \{0, 0, 0, 0, 0, 0\}$
**for** $0 \le h \le \mathrm{height}(\mathrm{T})$ **do**
  **for** $s$ such that $\mathrm{height}(s) = h$ **do**
    Compute $\mathcal{I}(\mathrm{T}_s)$ relying on $\mathcal{I}(\mathrm{T}_t)$ for $t \in \mathrm{ch}(s)$ with (3.30)
    Compute $F(\mathrm{T}_s) = \min_{1 \le q \le |\mathcal{I}(\mathrm{T}_s)|} \left[ F_q + \kappa_q - \frac{\|\theta_q\|^2}{N_q} + \beta \right]$ as in (3.29).
    Compute $q^*_s$ an index achieving the above minimum.
    Add $\mathcal{I}_0(\mathrm{T}_s) = \left\{ F(T_s), y_s, y_s^2, 1, \{0\}^{|\,\mathrm{ch}(s)|}, \{0\}^{|\,\mathrm{ch}(s)|} \right\}$ to account for the empty set.
    Prune $\mathcal{I}(\mathrm{T}_s)$ by removing indices $q$ such that $F_q + \kappa_q - \frac{\|\theta_q\|^2}{N_q} > F(\mathrm{T}_s)$
  **end for**
**end for**
**Backtracking step**

In the above algorithm, the pruning step is the counterpart of (3.20) with $\kappa = -\beta$. Although this version of PELT tree is only described for least-squares criterion, the general approach can be extended to other cost functions. In general, we only require that the cost $C_S$ on $(y_s)_{s \in S}$ is a function of several statistics and that these statistics can be updated on the fly. This property holds for instance when the cost is negative log-likelihood for Poisson random variables. See the numerical section (Section 3.5).

Since we do not store the sub-trees $S^*_s$ anymore, we need to rely on a new backtracking algorithm. The general idea is the following: we start from the root $o$ of T and consider the optimal index $q^*_o$. Although $\mathcal{I}_{q^*_o}$ does not contain a full description of $S^*_o$, it contains the indices $(\mathbf{id}^{q^*_o}_t)$ for $t \in \mathrm{ch}(o)$ of the restrictions of $S^*_o$ to $\mathrm{T}_t$. Looking at the corresponding indices allow us to explore $S^*_o$ step by step. If at some point, one of the explored indices $(\mathbf{id}^q_t)$ is null this means that $\mathrm{T}_t$ does not intersect with $S^*_o$, and that we have encountered a break edge. At $t$, we need therefore to start exploring a new connected component whose index is $q^*_t$ (as defined in Algorithm 6). Iterating this procedure, we shall step by step visit all the nodes. The pseudo-algorithm (7) described below is written in terms of the break edges instead of the partition. As explained in the introduction, the knowledge of the break edges is equivalent to that of the partition.

**Algorithm 7** Backtracking algorithm - Least square criterion

Initialize an empty list $\mathcal{L}^b$ for the breaks of the optimal segmentation of the tree.
Initialize an empty list $\mathcal{L}$ accounting for the current indices.
$\mathcal{L}[o] = q_o^*$
**for** height(T) $\geq h \geq 1$ **do**
  **for** $s$ such that height$(s) = h$ **do**
    $q = \mathcal{L}[s]$
    **for** $t \in \mathrm{ch}(s)$ **do**
      **if** $\mathbf{b}_t^q = 0$ **then**
        Add the break edge $(s, t)$ to $\mathcal{L}^b$
        Set $\mathcal{L}[t] = q_t^*$.
      **else**
        Set $\mathcal{L}[t] = \mathbf{id}_t^q$.
      **end if**
    **end for**
  **end for**
**end for**

## 3.4 Partition selection for Gaussian Models

In this section, we focus on mean segmentation models with Gaussian noise. More specifically, we assume in this section that $y_s \in \mathbb{R}^p$ and that, for each node $s \in \mathrm{T}$,

$$y_s = \mu_s^* + \epsilon_s \ , \tag{3.31}$$

where $\mu_s^* \in \mathbb{R}^p$ is the unknown mean and $\epsilon_s$ is a centered Gaussian random vector with covariance matrix $\sigma^2 \mathbf{I}_p$. Here, $\mathbf{I}_p$ stands for the $p$-dimensional identity matrix and $\sigma^2$ is known. In this model, the true partition $\underline{\mathbf{P}} = \{\underline{S}_1, \ldots, \underline{S}_K\}$ is the minimal partition such that the mean sequence $\mu_s^*$ is constant on each sub-tree $\underline{S}_k$.

This section is dedicated to crafting a statistically sound estimator of $\underline{\mathbf{P}}$. For a partition $\mathbf{P} = (S_1, \ldots, S_K)$ of T, we consider its (non-penalized) least-squares cost.

$$\mathrm{Cost}_{ls}(\mathbf{P}) = \sum_{i=1}^{K} \sum_{s \in S_i} \Big\| y_s - \sum_{t \in S_i} \frac{y_t}{|S_i|} \Big\|_2^2$$

Then, given a penalty function $\mathrm{pen} : \mathcal{P} \to \mathbb{R}^+$, we select a partition $\widehat{\mathbf{P}}$ by minimizing the penalized criterion

$$\widehat{\mathbf{P}} \in \arg\min_{\mathbf{P} \in \mathcal{P}} \mathrm{Cost}_{ls}(\mathbf{P}) + \mathrm{pen}(\mathbf{P}) \ . \tag{3.32}$$

If we choose $\mathrm{pen}(\mathbf{P}) = \beta |\mathbf{P}|$ for some $\beta > 0$, then $\mathrm{pen}(\mathbf{P})$ can be integrated into the cost function and $\widehat{\mathbf{P}}$ can be interpreted as a cost minimizer partition. However, for non-linear penalty function, this estimator $\widehat{\mathbf{P}}$ is not necessarily a cost minimizer. Leaving temporarily the computational aspects aside, we first focus on suitable choice of the function pen(.) so that $\widehat{\mathbf{P}}$ is as good as possible.

In what follows $\|y\|_{\mathrm{T}}^2$ stands for $\sum_{s \in \mathrm{T}} \|y_s\|^2$.

### 3.4.1 Oracle inequality

To do so, we shall build on Lebarbier's work [71] for univariate observation on the chain graph and more generally on the Birgé Massart model selection theory [13]. In order to use this framework, we need some new notation.

Consider any partition $\mathbf{P} = (S_1, \ldots, S_K)$ of T. Define the subspace $V_{\mathbf{P}}$ of vectors $\mu$ that are piece-wise constant according to this partition. In other words, for any $i = 1, \ldots, K$ and any $t_1$, $t_2$ in $S_i$, we have $\mu_{t_1} = \mu_{t_2}$. The least-squares estimator $\widehat{\mu}_{\mathbf{P}}$ in $V_{\mathbf{P}}$ is then defined as

$$\widehat{\mu}_{\mathbf{P}} = \arg\min_{\mu \in V_{\mathbf{P}}} \|y - \mu\|_T^2 \tag{3.33}$$

Note that $(\widehat{\mu}_{\mathbf{P}})_s$ is equal to the mean of the observation in the sub-tree $S_i$ of T that contains $s$, so that

$$\|y - \widehat{\mu}_{\mathbf{P}}\|_T^2 = \sum_{i=1}^{K} \sum_{s \in S_i} \Big\| y_s - \sum_{t \in S_i} \frac{y_t}{|S_i|} \Big\|_2^2 = \mathrm{Cost}_{ls}(\mathbf{P}) \ .$$

Second, we introduce $\mu_{\mathbf{P}}$ as the projection of $\mu^*$ onto $V_{\mathbf{P}}$, that is we have $\mu_{\mathbf{P}} = \arg\min_{\mu \in V_{\mathbf{P}}} \|\mu^* - \mu\|_T^2$. For any $s \in S_i$, $(\mu_{\mathbf{P}})_s = (\sum_{t \in S_i} \mu_t^*)/|S_i|$. The quantity $\|\mu_{\mathbf{P}} - \mu^*\|_T^2$ corresponds to the square norm of the bias of the estimator $\widehat{\mu}_{\mathbf{P}}$ of $\mu^*$. The following lemma states the bias-variance decomposition of the estimator $\widehat{\mu}_{\mathbf{P}}$.

**Lemma 15.** *For any partition $\mathbf{P}$ in $\mathcal{P}$, we have*

$$\mathbb{E}\left[\|\widehat{\mu}_{\mathbf{P}} - \mu^*\|_T^2\right] = \|\mu_{\mathbf{P}} - \mu^*\|_T^2 + \sigma^2 p |\mathbf{P}| \ .$$

Next, we provide an oracle-like inequality for the least-square estimator $\widehat{\mu}_{\widehat{\mathbf{P}}}$ based on the selected partition.

**Theorem 2.** *Fix any $\eta > 1$, $\theta > 0$, and $\zeta > 0$. There exists two constants $c_\eta$ and $c'_\eta$ only depending $\eta$ such that the following holds. If the penalty function satisfies*

$$\mathrm{pen}(\mathbf{P}) \geq \eta\sigma^2 \left[ (1+\theta)|\mathbf{P}|p + 2(1+\theta^{-1})[|\mathbf{P}| - 1]\left( (1+\zeta) + \log\left( \frac{n-1}{|\mathbf{P}|-1} \right) \right) \right] \ , \tag{3.34}$$

*for all $\mathbf{P} \in \mathcal{P}$, then the selected partition $\widehat{\mathbf{P}}$ (from (3.32)) satisfies*

$$\mathbb{E}\left[\|\widehat{\mu}_{\widehat{\mathbf{P}}} - \mu^*\|_F^2\right] \leq c_\eta \inf_{\mathbf{P} \in \mathcal{P}} \left[ \|\mu_{\mathbf{P}} - \mu^*\|_T^2 + \mathrm{pen}\left(|\mathbf{P}|\right) \right] + c' \frac{e^\zeta}{e^\zeta - 1} \sigma^2 \ . \tag{3.35}$$

The proof of this theorem is postponed to the appendix.

**Remarks**. Theorem 2 states that the selected partition $\widehat{\mathbf{P}}$ has an associated least-square estimator $\widehat{\mu}_{\widehat{\mathbf{P}}}$ which performs nearly the best among least-square estimators $\widehat{\mu}_{\mathbf{P}}$. In particular, specifying the right-hand side of (3.35) for $\mathbf{P} = \underline{\mathbf{P}}$ (the true partition) and choosing a penalty as in (3.34), we deduce from Lemma 15

$$\mathbb{E}[\|\widehat{\mu}_{\widehat{\mathbf{P}}} - \mu^*\|_F^2] \leq c_{\eta,\zeta,\theta} \left[ \mathbb{E}\left[\|\widehat{\mu}_{\underline{\mathbf{P}}} - \mu^*\|_T^2\right] + \sigma^2 [|\underline{\mathbf{P}}| - 1] \log\left( \frac{n-1}{|\underline{\mathbf{P}}|-1} \right) \right] \ .$$

Hence, the risk of $\widehat{\mu}_{\widehat{\mathbf{P}}}$ is nearly as good as the estimator associated to the unknown true partition up to an additive $[|\underline{\mathbf{P}}| - 1]\log(\frac{n-1}{|\underline{\mathbf{P}}|-1})$ term. This suggests that $\widehat{\mathbf{P}}$ is a good estimator $\underline{\mathbf{P}}$. In the specific case where $p = 1$ (univariate data) and T is a chain graph, we recover the oracle inequality [71]. Interestingly, the penalty in (3.34) only depends on the dimension $p$ of the signal through the variance term $|\mathbf{P}|p$.

### 3.4.2 Implementation

Unfortunately, the penalty $\mathrm{pen}(\mathbf{P})$ introduced in the previous section is not linear with respect to $\mathbf{P}$. Thus, we cannot incorporate it into the cost function and then apply PELT Tree Algorithm. Nevertheless, $\mathrm{pen}(\mathbf{P})$ writes as a function $f(|\mathbf{P}|)$ where $f$ is strictly concave and differentiable. For the problem of change point detection on the chain graph, this motivated Killick et al [64] to run a

gradient descent-like algorithm amounting iteratively minimize the least-square cost with a linearized penalty. Here, we readily adapt their approach in our setting.

The general idea is to improve iteratively the estimator. From a current estimator $\mathbf{P}_l$, we compute

$$\mathbf{P}_{l+1} \in \arg\min_{\mathbf{P}\in\mathcal{P}}[\mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) + |\mathbf{P}|f'(|\mathbf{P}_l|)] \ ,$$

with PELT-Tree algorithm. Then, we iterate by replacing $f'(|\mathbf{P}_l|)$ by $f'(|\mathbf{P}_{l+1}|)$. The algorithm stops when the size of the partition $|\mathbf{P}_{l+1}|$ does not change.

We provide below the algorithm for the problem of minimizing a general cost function penalized by a strictly concave penalty. Then Proposition 1 below states that the procedure stops at a local minimum of the penalized criterion.

**Algorithm 8** Tree Segmentation with Model Selection

Initialize $K = 0$ and $\beta = f'(0)$
Compute $\widehat{\mathbf{P}} \in \arg\min_{\mathbf{P}\in\mathcal{P}}[\mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) + \beta|\mathbf{P}|]$ by Algorithm 5
**while** $K \neq |\widehat{\mathbf{P}}|$ **do**
  Set $K = |\widehat{\mathbf{P}}|$
  Set $\beta = f'(K)$
  Compute $\widehat{\mathbf{P}} \in \arg\min_{\mathbf{P}\in\mathcal{P}}[\mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) + \beta|\mathbf{P}|]$ by Algorithm 5
**end while**
**return** $(\widehat{\mathbf{P}},\ \mathrm{Cost}_{\mathrm{T}}(\widehat{\mathbf{P}}) + \mathrm{pen}(\widehat{\mathbf{P}}))$

**Remark**. When we use a least-square cost function, the above algorithm can be improved by relying on Algorithm 6 instead of Algorithm 5.

**Proposition 1.** *Assume that the penalty function $f$ is strictly concave and differentiable. Then, the penalized cost of $\widehat{\mathbf{P}}$ is decreasing at each step of Algorithm 8 and is strictly decreasing at each step before the last one. Furthermore, any global minimizer $\widehat{\mathbf{P}}_{\mathrm{pen}}$ of the penalized cost is a fixed point of the above recursion, that is*

$$\widehat{\mathbf{P}}_{\mathrm{pen}} \in \arg\min_{\mathbf{P}\in\mathcal{P}}[\mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) + f'(|\widehat{\mathbf{P}}_{\mathrm{pen}}|)|\mathbf{P}|)]$$

Unfortunately, the above proposition does not certify that the partition $\widehat{\mathbf{P}}$ converges to a global minimum of the penalized cost.

## 3.5 Numerical experiments

We evaluate of the performance of our PELT Tree Algorithm with model selection 8 on synthetic data. Since we have no algorithm using our methodology on trees to compare it too, we will give raw performances such as time of computation, number of partitions pruned at each step and accuracy. Throughout this section, the tree T is a rooted balanced binary tree. We present the results of three different experiments which aim to assess the performance in term of pruning, computational time and accuracy of Algorithm 8 in particular statistical models.

**Statistical models** We consider two statistical models:

- The Gaussian setting is analogous to that studied in Section 3.4. The observations $(y_s)_{s\in\mathrm{T}}$ follow a normal distribution with mean $\mu_s^*$ and variance $\sigma^2$.

$$y_s \sim \mathcal{N}(\mu_s^*, \sigma) \quad \forall s \in \mathrm{T} \tag{3.36}$$

  We write $\underline{\mathbf{P}}$ for the minimal partition of T such that the mean $\mu_s^*$ is constant on each sub-tree of $\underline{\mathbf{P}}$.

- The Poisson setting is closer to our analysis of fish abundance data in the next section. The observation $y_s$ at node $s$ is a $p$-dimensional vector of non-negative integers. The coordinates $y_{s,j}$ are independent and follow a Poisson distribution :

$$y_{s,j} \sim \mathcal{P}\text{oi}\left(e^{\alpha_s + \mathbf{b}_{s,j}}\right), \qquad s \in \mathrm{T},\ j = 1, \dots, p\ , \tag{3.37}$$

where $\alpha_s$ is a 'site' effect and the parameter $\mathbf{b}_{s,j}$ is a parameter depending on both the site $s$ and the component $j$. There exists a (minimal) partition $\underline{\mathbf{P}}$ of T into regions such that, for each $j = 1, \dots, p$, $\mathbf{b}_{s,j}$ is constant inside each component of the partition. This setting slightly deviates from our definition of segmentation problems in the introduction because the nuisance parameter $\alpha_t$ is not assumed to be constant on elements of the partitions.

**Cost and Penalty functions** We present here the algorithm based on Algorithm 8 that was implemented for our experiments.

In the Gaussian setting, the cost function considered is the Penalized least-square. This setting is studied in Section 3.3.4 and we saw in Section 3.4 that it is possible to use Algorithm 6 instead of Algorithm 5 in the model selection Algorithm 8. Hence, we use the modified Algorithm 8 with a Penalized least square cost.
For the penalty function, its form is based on the work of Lebarbier in [71]. Lebarbier estimates the change-points in a signal on chains with a Gaussian setting similar to ours using a Penalized least square cost and model selection on a concave penalty depending on unknown constants. Our settings and goals being similar to the one in [71], we chose the same form of penalty :

$$\mathrm{pen}(\mathbf{P}) = \frac{|\mathbf{P}|}{n}\left(c_1 \log\left(\frac{n-1}{|\mathbf{P}|-1}\right) + c_2\right), \tag{3.38}$$

with $c_1$ and $c_2$ constants. Lebarbier empirically estimates that $c_1 = 2$ and $c_2 = 5$ gives the best results with her univariate Gaussian setting. Hence, we set the penalty function of Algorithm 8 to (3.38) with this choice of values for $c_1$ and $c_2$.

In the Poisson setting, the observations follow a multivariate Poisson distribution.
The cost function used for this setting is the penalized minus log-likelihood function. We now explain how some simplification of this function can be made in order to make it easier to compute in Algorithm 8.
Given a subset $S \subset T$, we write $\mathbf{y}_S$ for the observations in $S$ and $l(\mathbf{y}_S, \theta_S)$ for the log-likelihood of $\mathbf{y}_S$ in Model (3.37) where we assume that all $\mathbf{b}_{s,j}$ for $s \in S$ are equal to some $\mathbf{b}_S \in \mathbb{R}^p$. Here, $\theta_S$ is short for $(\alpha_s, s \in S; \mathbf{b}_S)$. We consider the cost function $C_S$ defined by

$$C_S = \sum_{s \in S}\sum_{j=1}^{p} y_{s,j} \log\left(\frac{\sum_{t\in S}\sum_{i=1}^{p} y_{t,i}}{\sum_{t\in S} y_{t,j}}\right)\ . \tag{3.39}$$

The following proposition, proved in the appendix, states that minimizing the cost $\sum_{S\in\mathbf{P}} C_S$ is equivalent to finding the partition with the smallest negative log-likelihood.

**Proposition 2.** *For any partition* $\mathbf{P}$*, the cost function* $C_S$ *defined in* (3.39) *satisfies*

$$\sum_{S\in\mathbf{P}} \min_{\theta_S} -l(\mathbf{y}_S, \theta_S) = \sum_{S\in\mathbf{P}} C_S + \sum_{s\in T}\sum_{j=1}^{p}\left(y_{s,j}\left(1 - \log\sum_{i=1}^{p} y_{s,i}\right) + \log(y_{s,j}!)\right). \tag{3.40}$$

Note that $C_S$ in (3.39) is only a function of $\sum_{s\in S} y_{s,j}$ for $j = 1, \dots, p$. Hence, it is possible to adapt the faster algorithm (Algorithm 6) to the Poisson setting, by only storing the partial $\sum_{s\in S} y_{s,j}$. This allows us to minimize the linearly penalized cost function by PELT Tree.

For this setting, we use Algorithm 8 in which we replace Algorithm 5 with the adaptation of Algorithm 6 to the Poisson setting. As stated before the cost function is 3.40 and we choose a penalty function inspired by (3.34) introduced in Section 3.4 for the multivariate Gaussian model. Our intuition is that the form of the penalty function calculated for the multivariate Gaussian model can also be used in the Poisson setting in the improved version of Algorithm 8. We do not prove it formally in this document but we empirically verify that this choice of penalty works for the multivariate Poisson model.
We still have to adapt the form of the penalty (3.34) to the Poisson setting. Since the variance of a Poisson distribution is equal to its mean, we are not in the case of a known variance. We replaced $\sigma$ by 1 in (3.34). The final form of the penalty function is :

$$\text{pen}(\mathbf{P}) = \eta\left(|\mathbf{P}| - 1\right)\left[(1+\theta)p + 2\left(1+\theta^{-1}\right)\left((1+\zeta) + \log\left(\frac{n-1}{|\mathbf{P}|-1}\right)\right)\right], \tag{3.41}$$

where $\eta > 1$, $\theta > 0$ and $\zeta > 0$ are some constants.
In our experiments, the dimension $p$ of the observations is set to 10 while the size $n$ of the tree T varies. While we choose $\eta$ and $\zeta$ to be as small as possible, compromise have to be made when choosing $\theta$ since both $\theta$ and $\theta^{-1}$ figure in (3.41). We choose our constants accordingly and trying to get the smallest penalty function as possible. We arbitrarily choose $\eta = 1.1$, $\zeta = \frac{1}{100}$ and $\theta = 0.5$.
We can replace the different constants term in (3.41) by three general constant $c_1'$, $c_2'$ and $c_3'$.

$$\text{pen}(\mathbf{P}) = \left(|\mathbf{P}| - 1\right)\left[c_1' \log\left(\frac{n-1}{|\mathbf{P}|-1}\right) + c_2' p + c_3'\right]. \tag{3.42}$$

For our choice of $\eta$, $\zeta$ and $\theta$, we obtain $c_1' = 1.65$, $c_2' = 6.6$ and $c_3' = 5.6$.

**Construction of the binary tree structure.** Now that the statistical models and the implemented algorithms are presented, we need to explain how the rooted balanced binary trees are constructed.
The nodes of the rooted balanced binary tree T are indexed by integers between 1, the root, and $n$ the number of nodes in T such that for any node $i$ of T that is not a leaf, its children are indexed by $2i$ and $2i+1$. This can be done by browsing the tree from the roots to the leaves.
We represent the structure of T as a list. This list is indexed by the nodes of T, that is to say by the integers between 1 and $n$ which represent the nodes. The $i$-th element of the list is a vector containing the children of the node indexed by $i$.
To simulate such a list, we first create an empty list of size $n$. Since T is a binary tree, there exists a certain integer $k$ greater than 1 for which $2^{k-1} - 1 < n \leq 2^k - 1$. Then, for $i$ between 1 and $2^{k-1} - 1$, the $i$-th element of the list is set to the vector $(2i, 2i+1)$ if $2i+1$ is smaller or equal to $n$, to the vector $(2i)$ if $2i$ is equal to $n$ and to 0 if $2i$ is greater than $n$. The nodes indexed by integers between $2^{k-1}$ and $n$ are the leaves of the tree, we set the corresponding elements of the list to 0. In the following, the nodes of T are referred as their index.

**Construction of the partition $\underline{\mathbf{P}}$ of T.** Once the structure of a tree T of size $n$ is constructed, we have to construct a partition of T. In the chain case, the performances of the change-point detection algorithm vary with the number of sub-trees. The case where the number of sub-trees is linear in the number of nodes gives the best results in terms of computational cost while the worst case seems to be when there is only one sub-tree in the true partition of T [64]. The latter partition corresponds to the case where there are no change in the signal and therefore no break between any nodes in T. We call this case the No Break case. We want to simulate two kind of partitions of T :

- The linear case where the number of sub-trees is linear with respect to the size $n$ of T.
- The No Break (NB) case where there is only one sub-tree : T.

**Simulation of the observations.** To simulate observations in the No Break case is simple in both Gaussian and Poisson settings. In the Gaussian setting, we chose at random a mean $\mu^*$ and simulate a $n$-Gaussian vector of mean $\mu^*$ and variance $\sigma\,\mathbf{Id}$ with $\sigma$ chosen arbitrarily. In the Poisson setting, we chose at random the $\alpha_s^*$ parameter for each node $s$ and a $p$-vector $(b^*)_{1\leq j\leq p}$. Then, for all sites $s$, $b_{s,j}^*$ is equal to $b_j^*$ for all $j$ between 1 and $p$ and we can simulate the observations as in the Poisson setting, that is to say, we create a matrix of observations of size $n \times p$ such that the $(s,j)$-th coefficient is a random realization of a Poisson distribution of mean $e^{\alpha_s^*+\mathbf{b}_j^*}$ for $i$ between 1 and $n$ and $j$ between 1 and $p$.

The simulation of observations in the linear number of change points is more involved, since we first have to divide T into different sub-trees and then simulate observations on each sub-tree. We decided to have only one constraint on the size of the sub-trees, that is that each sub-tree must be of size at least 3. In general, to construct a partition of T into $K$ sub-trees, we chose $K-1$ integers, representing nodes of T, denoted $n_1,\ldots,n_{K-1}$ such that $n_1 > n_2 > n_{K-1}$ and we set $n_K$ to 1. We set the first sub-tree as $n_1$ and its descendants in T, then for $i$ between 2 and $K$, we construct the $i$-th sub-tree as $n_i$ and its descendants minus the nodes in the already constructed sub-trees.

Since the sub-trees have to be of size at least 3, the $n_i$ cannot be leaves of T, that is to say, if T is of size $n$ and $k$ is such that $2^{k-1}-1 < n \leq 2^k-1$, the $n_i$ for $i$ between 1 and $K-1$ are chosen uniformly at random between 2 and $2^{k-1}-1$. It is not enough to ensure that the constructed partition will have all its sub-trees of size at least 3. For example, if $n_{K-1}=2$ and $n_{K-2}=3$, the associated partition contains the sub-tree $\{1\}$ which is of length 1. We decided to keep choosing the $n_i$ at random until the constructed partition contains only sub-trees of size at least 3. Since $K$ is reasonably low compared to $n$, this method is efficient and allows us to chose the sub-trees at random instead of setting arbitrary partitions. For any tree T, its true partition is denoted $\underline{\mathbf{P}}$.

Once the partition of T has been constructed, we can simulate the observations almost as in the No Break case. In the Gaussian setting, we chose at random $K$ parameters $\boldsymbol{\mu}_i^*$ for $i$ between 1 and $K$. Then, for $i$ between 1 and $K$, we consider the $i$-th sub-tree and for all sites $s$ in this sub-tree we set $\mu_s^* = \boldsymbol{\mu}_i^*$. We construct a vector of observations of size $n$ such that its coefficients are i.i.d and such that its $s$-th coefficient is a realization of a normal distribution of mean $\mu_s^*$ and variance $\sigma$. In the Poisson setting, we chose the $\alpha_s^*$ at random as in the No Break case and a $K \times p$ matrix $\mathbf{b}^*$ with random coefficients. Then, for $i$ between 1 and $K$, we consider the $i$-th sub-tree, and for all the nodes $s$ that belong to this sub-tree and all $j$ between 1 and $p$ we set $b_{s,j}^* = \mathbf{b}_{i,j}^*$. We create a $n \times p$ matrix of observations such that its coefficients are i.i.d. and such that its $(s,j)$-th coefficient is a realization of a Poisson distribution of mean $e^{\alpha_s^*+b_{s,j}^*}$.

**Presentation of our experiments on Algorithm 8.** We are now able to construct binary trees and to simulate observations on these trees according to a randomly chosen partition. In the following experiments, the parameters of the binary trees and the partitions vary in order to compute results on the performance of our algorithms. One of the outputs of Algorithm 8 is the estimated partition of T denoted $\widehat{\mathbf{P}}_{\text{pen}}$. We conduct three different experiments. The first experiment focus on the pruning step of Algorithm 8; its purpose is to demonstrate that the pruning method is efficient with our different settings. The second experiment focus on the computational time. We construct binary trees of different size and their partition by changing the parameters $n$ and $K$. Its purpose is to make sure that the computational cost of our algorithm is reasonable even in the No Break case. The third and last experiment aims to make sure that $\widehat{\mathbf{P}}_{\text{pen}}$ is a good estimator of $\underline{\mathbf{P}}$. To do so, we change the variance $\sigma$ or the range in which we choose the means of the observations and observe the consequences on the outputs of our algorithms.

In the third experiment, we assess the ability of our algorithm to estimate $\underline{\mathbf{P}}$. We stated that in order to construct a partition of T we have to choose $K-1$ nodes $n_1,\ldots,n_{K-1}$ in T. Also, $\widehat{\mathbf{P}}_{\text{pen}}$ is associated to a set of breaks, that is to say edges of T such that if we remove these edges from T, the

induced connected components are the sub-trees of $\widehat{\mathbf{P}}_{\text{pen}}$. We notice that, in the case of balanced binary trees, each $n_i$ for $i$ between 1 and $K-1$ corresponds to the edge of T between $i$ and its parent.
We want to compute the percentage of breaks the estimator $\widehat{\mathbf{P}}_{\text{pen}}$ has in common with $\underline{\mathbf{P}}$. We call accuracy this percentage. The way we compute the accuracy is the following. We denote $K_{\widehat{\mathbf{P}}_{\text{pen}}}$ the number of sub-trees in $\widehat{\mathbf{P}}_{\text{pen}}$ and $N_{\widehat{\mathbf{P}}_{\text{pen}}}$ the number of sub-trees of $\underline{\mathbf{P}}$ that are in $\widehat{\mathbf{P}}_{\text{pen}}$.
In the specific case where $K$ is equal to 1, which is when there are no change in the signal, we want to compute whether or not the algorithm detected that $\underline{\mathbf{P}} = \{\mathrm{T}\}$. Hence, if $K = 1$, the accuracy is equal to 1 if $K_{\widehat{\mathbf{P}}_{\text{pen}}}$ is equal to 1, and 0 otherwise.

$$\text{accuracy} = \begin{cases} \dfrac{N_{\widehat{\mathbf{P}}_{\text{pen}}}}{\max\left\{K-1, K_{\widehat{\mathbf{P}}_{\text{pen}}}-1\right\}} & \text{if} \quad K > 1 \\ \mathbb{1}_{K_{\widehat{\mathbf{P}}_{\text{pen}}}=1} & \text{if} \quad K = 1 \end{cases} \tag{3.43}$$

### 3.5.1 First experiment

Here, we consider a balanced binary tree with $n = 199$ nodes. We consider 4 settings:

- [(*Gauss*)] The true partition $\underline{\mathbf{P}}$ is made of $K = 21$ sub-trees. The observations follow the Gaussian model (3.36) with $\sigma^2 = 1$, whereas the value of the mean parameters in a sub-tree of the partition is sampled uniformly in $[-20; 20]$.
- [(*Poiss*)] The true partition is the same as above. The observations follow a Poisson model (3.37). The sites effects $\alpha_t$ are sampled uniformly in $[\log(1/2); \log(2)]$, and the parameters $\mathbf{b}$ in a sub-tree of the partition are sampled independently and uniformly among all integers in $[1; 100]$.
- [(*GaussNB*)] The true partition $\underline{\mathbf{P}}$ equals $\{\mathrm{T}\}$ (no jump). The observations follow the Gaussian model (3.36) with $\sigma^2 = 1$ whereas the mean parameter is sampled uniformly in $[-20; 20]$.
- [(*PoissNB*)] As previously, $\underline{\mathbf{P}} = \{\mathrm{T}\}$. The observations follow a Poisson model (3.37) with parameters sampled as in (Poiss).

In (Gauss) and (Poiss), the signal parameter ($\mu$ and $\mathbf{b}$) are chosen in a wide range of values so that the jumps are detectable. In these two settings, the tree T is fixed since the number of nodes $n$ is always equal to 199 but we make 50 iterations of the procedure with different partitions $\underline{\mathbf{P}}$ and sets of parameters $\mu_s^*$, $\alpha_s^*$ and $\mathbf{b}_{s,j}^*$ at each iteration. In (GaussNB) and (PoissNB) we only make one iteration of Algorithm 8 since the partition of T into 1 sub-tree is unique.

For the 4 settings, we want to assess two quantities : the number of possible partitions in $\mathcal{S}_{pr,1}(\mathrm{T}_s)$ versus the number of descendants of $s$ in T for each node $s$ of T.
In (Gauss) and (Poiss), the number of descendants of each node $s$ of T does not depend on the true partition of T or on the choice of the parameters, but the number of possible partitions in each $\mathcal{S}_{pr,1}(\mathrm{T}_s)$ does. We take the mean over all the 50 iterations of the number of possible partitions in $\mathcal{S}_{pr,1}(\mathrm{T}_s)$ in order to get only one value for each number of descendants of $s$.
In (GaussNB) and (PoissNB) we do not need to take the mean since we make only one iteration of the algorithm.

In figure 3.3, we plot the average number of possible partitions in $\mathcal{S}_{pr,1}(\mathrm{T}_s)$ versus the size of $\mathrm{T}_s$, that is to say the number of descendants of $s$, for each $s \in \mathrm{T}$, .
The number of partitions seems at worst linear in the number of descendants. The graphs are similar when the observations follow a Gaussian distribution and a Poisson distribution. However, we observe a difference of behavior of the graphs between the No Break case and the linear case. Indeed, the algorithm prunes less partitions when $\underline{\mathbf{P}}$ is the partition with only one sub-tree. This is coherent with the chain case. Indeed, the case where there are no breaks in the true partition of the data

gives the worst computational cost in change-points detection problem on chains. The algorithms that use pruning always prune less partitions when there are no break in the true partition, even with the right choice of the penalty parameter $\beta$.

We observe that in this experiment, the pruning of our algorithm is sufficiently efficient to make the complexity of our algorithm linear in the size of T.

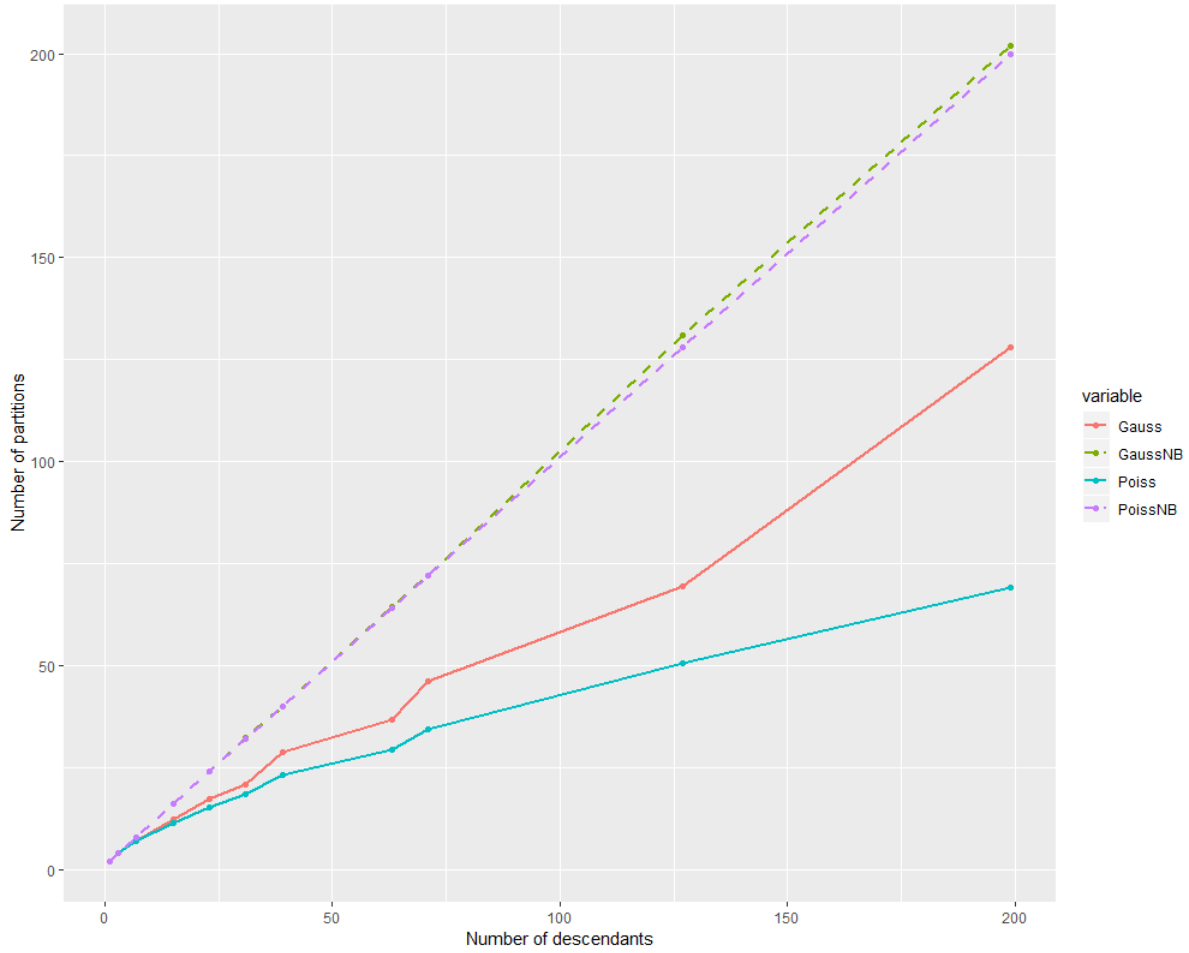



Figure 3.3 – Number of partitions in $\mathcal{S}_{pr,1}(\mathrm{T}_s)$ versus number of descendants of $s$ in T for rooted balanced binary trees of size $n = 199$. We run our algorithm on the 4 settings introduced earlier. For all settings, the number of possible partitions is taken after the pruning step in Algorithm 8. For (Gauss) and (Poiss), the number of sub-trees in $\underline{\mathbf{P}}$ is set to $K = 21$. In these two settings, we made 50 iterations of the algorithm, with different partitions $\underline{\mathbf{P}}$ and different sets of parameters at each iteration, and then took the mean over all iteration of the number of possible partitions for all $s \in \mathrm{T}$. The number of descendants varies between 0, for the leaves, and 198, for the root.

### 3.5.2 Second experiment

The purpose of this experiment is to assess the computational cost of our algorithm on rooted balanced binary trees of different size.
We consider different values for the size $n$ of T with $n$ in $(39, 79, 119, 159, 199, 299, 399, 599, 799, 999, 1499, 1999)$.

For each value of $n$, we consider the four settings (Gauss), (Poiss), (GaussNB) and (PoissNB) of the previous experiment except that in (Gauss) and (Poiss) the number of sub-trees in the partition $\underline{\mathbf{P}}$ is now set to $\lfloor \frac{n}{5} \rfloor + 1$. This makes the size of the partitions linear with $n$.

To construct the tree T, its partitions and to simulate the observations on T we follow the same procedure presented in the introduction of this subsection and the first experiment. As in the previous experiment, there is no need to iterate multiple times our algorithm in the (GaussNB) and (PoissNB) settings. In the two other settings (Gauss) and (Poiss), we are aware that the computational cost of one iteration of Algorithm 8 can differ from one partition of T to another, and from one set of parameters to another. To overcome this issue, for each value of $n$, we iterate 10 times our algorithm with different partitions $\underline{\mathbf{P}}$ and different sets of parameters at each iteration. Then, we take the mean of the computational cost over all the iterations in order to obtain one value for each $n$ . The computational cost increases with the size of T and we noticed that the variation of the computational cost is low from one iteration to another for the same value of $n$. This justifies the fact that we choose to realize only 10 iterations of our algorithm for each value of $n$.

To measure the computational cost of Algorithm 8 we use the tictoc package in R [54]. In Figure 3.4, we plot the computational time of Algorithm 8 in seconds versus the size $n$ of T in our four settings.
We observe a similar behavior as in Figure 3.3 in both (Gauss) and (Poiss) settings. However, for a constant signal, the computational time seems at best quadratic in the size of the tree. The distribution, Gaussian or Poisson, of the observations does not seem to impact the computational cost of the algorithm. There is an important difference between the computational complexity of the constant scenario versus the linear number of breaks scenario which is much faster. This confirms the observation in the first experiment that PELT Tree prunes much more aggressively the candidate sub-trees when there are many breaks.

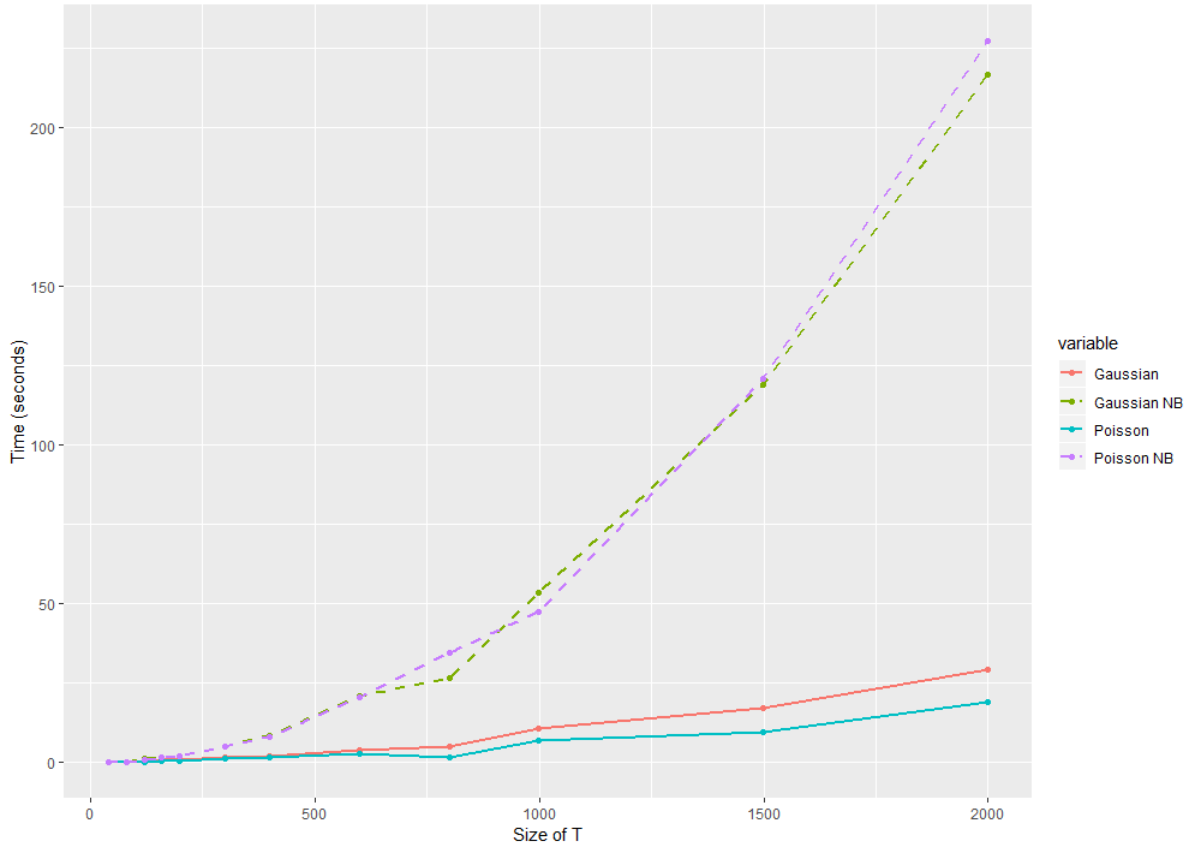



Figure 3.4 – Computation cost of Algorithm 8 (in seconds) versus the size of T. We plot the graph for all 4 settings (Gauss) (in plain red), (Poiss) (in plain blue), (GaussNB) (in dotted green) and (PoissNB) (in dotted violet). We used concave penalty with constant $c_1 = 2$ and $c_2 = 5$ for the Gaussian setting and $c_1 = 1.65$, $c_2 = 6.6$ and $c_3 = 5.6$ for the Poisson setting. In both settings, we used the version of Algorithm 8 for the penalized least square cost function ((Gauss) and (GaussNB)) and for the penalized minus log-likelihood ((Poiss) and (PoissNB)). For each setting and for each size of T, we iterate 10 times the algorithm and take the mean over all the iterations.

### 3.5.3 Third experiment

In the third and last experiment on the performance of Algorithm 8 with synthetic data, we consider a rooted balanced binary tree of size $n = 199$. We consider the same 4 settings as in the first experiment except that we now make vary the variance of the observations.

The goal is to prove that Algorithm 8 is capable of retrieving the partition $\underline{\mathbf{P}}$ of T under certain constraints on the variance $\sigma$ in (Gauss) and on the amplitude of the signal in (Poiss). We also aim to study the impact of the variance and the amplitude on the accuracy (3.43). In the first two experiments, we studied the performance of Algorithm 8, for the Gaussian and Poisson settings, in terms of pruning and computational cost but we did not study its ability to retrieve the true partition of T.

We construct a rooted balanced binary tree of size 199 and a partition $\underline{\mathbf{P}}$ of size 21 of T as indicated in the introduction of this section. In this experiment, both the size of T and $\underline{\mathbf{P}}$ are constant throughout the experiment for (Gauss) and (Poiss). For the No Break cases (GaussNB) and (PoissNB), we consider as previously the only partition of T of size 1: {T}.

In the previous Gaussian experiments $\sigma$ was set to 1. We now consider values of $\sigma$ in $(0.1, 0.5, 1, 2, 5, 7, 10)$. For each value of $\sigma$ we simulate 100 datasets with means $\mu_i^*$ and variance $\sigma$. In each iteration, we estimate $\underline{\mathbf{P}}$ thanks to Algorithm 8 for the (Gauss) setting. The output of this algorithm is the estimated partition $\widehat{\mathbf{P}}_{\text{pen}}$. Then, we compute the accuracy of our estimator thanks to (3.43) and

consider mean of the accuracy.

In (Poiss), we chose high jumps of the mean in the distribution of our observations. In this setting, the mean is equal to the variance. We chose to test the performance of our algorithm if we multiply the parameters $\mathbf{b}_{i,j}$ by a small factor for all $i$ between 1 and 21 and all $j$ between 1 and $p$. We expect the accuracy to drop if this factor is too small. We choose at random the parameters $\alpha_s$ and $\beta_{s,j}$ as in the previous experiment for (Poiss). Once the parameters are set, we set a range of values for the multiplicative factor $a$ which can take values in $(0.01, 0.1, 0.2, 0.8, 1, 2, 5)$. As in (Gauss), we perform 50 simulations of observations according to (Poiss) with a modification on the $\mathbf{b}$ parameter. For all $s$ nodes of T, the parameter $\mathbf{b}^*_{s,j}$ at site $s$ is equal to $a\mathbf{b}^*_{i,j}$ where $s$ belongs to the $i$-th sub-tree. In each iteration, we compute the estimator $\widehat{\mathbf{P}}_{\text{pen}}$ of $\underline{\mathbf{P}}$, thanks to Algorithm 8 improved for (Poiss), and calculate the accuracy of the estimator thanks to (3.43). At the end of the 100 iterations, we take the mean over all the calculated accuracy.

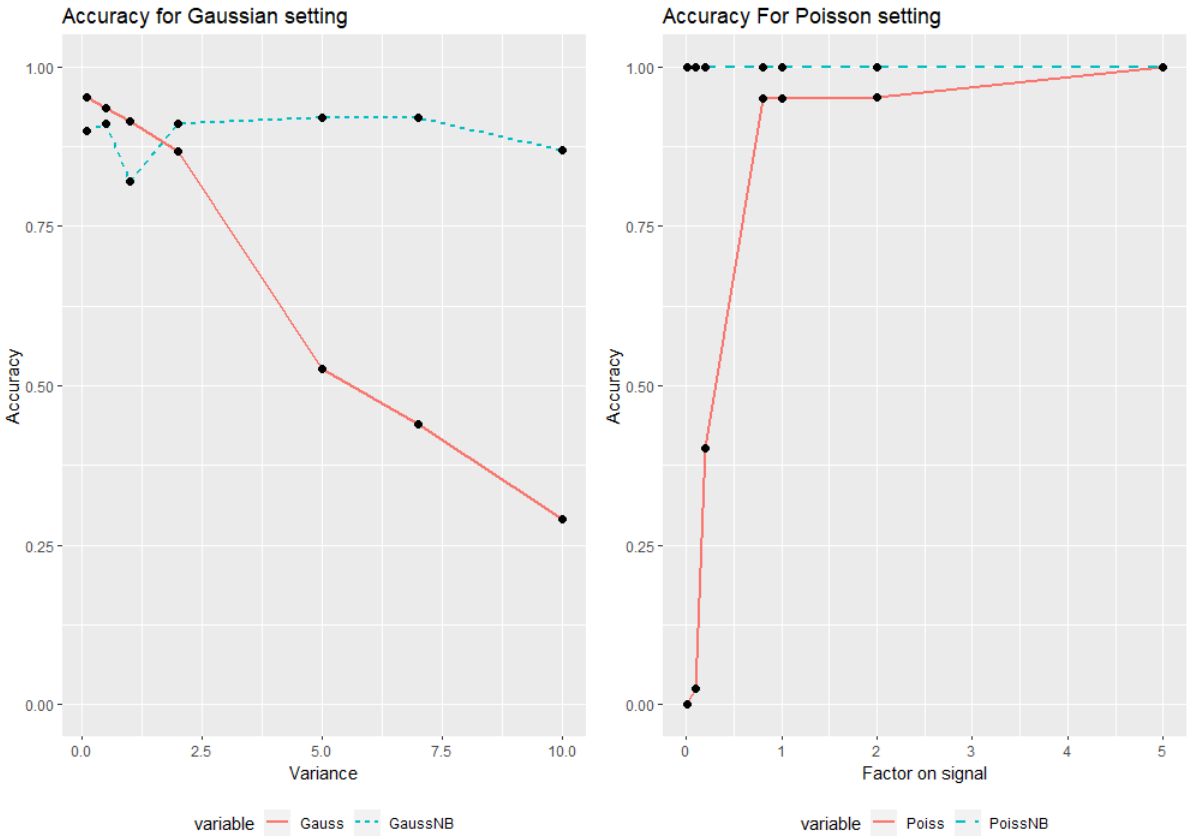



Figure 3.5 – On the (left) figure, Accuracy of Algorithm 8 with penalized least square cost versus the variance $\sigma$. The variance $\sigma$ takes its values in $(0.1, 0.5, 1, 2, 5, 7, 10)$. We plot the accuracy for the (Gauss) (plain red) and (GaussNB) (dotted blue) settings. We choose a concave penalty with constants $c_1 = 2$ and $c_2 = 5$. On the (right) figure, Accuracy of Algorithm 8 with penalized minus log-likelihood versus the multiplicative factor $a$. The factor $a$ takes its values in $(0.01, 0.1, 0.2, 0.8, 1, 2, 5)$. We plot the accuracy for the (Poiss) (plain red) and (PoissNB) (dotted blue) settings. We choose a concave penalty with constants $c_1 = 1.65$, $c_2 = 6.6$ and $c_3 = 5.6$. For each setting, for each value of $\sigma$, we iterate 100 times the algorithm and take the mean over all the iterations

In Figure 3.5, we can see, as expected, that the accuracy rapidly drops in (Gauss), when the variance rises. In (GaussNB) setting, the accuracy does not seem impacted by the change in variance. In (Poiss), if the signal is too small due to a small multiplicative factor, it becomes impossible for our algorithm to retrieve the true partition $\underline{\mathbf{P}}$. In (PoissNB) the accuracy is not affected by the factor $a$. We expected such a result since the No Break case corresponds to the case where the signal

is constant. The more the constant $a$ is small, the more the signal is close to a constant.

## 3.6 Fish abundance in Loire river network

We consider abundance measures along the river network of Loire river in France. The data are provided by The Museum of Natural History and were collected by the French Agency for the Biodiversity (AFB) and l'ONEMA (Office National de l'Eaux et des Milieux Aquatiques).

**Protocol and Motivations** The data comes from what is called electrofishing. A small currant of electricity is spread through the river attracting and then paralyzing the fishes. Once the fishes are floatting at the surface of the water, one can sample the abundances of the fishes by counting them. Our dataset gathers complete fishing observations which means that the observations are sampled accross all the width of the river. The rivers must be at most 0.7 metter deep.

The sites on which the abundances are sampled must be representative of the section of the river it belongs to. Their repartitions is homogeneous on a national scale. On each site, at least one sample must be performed each year, preferably at the end of the summer and in the same hydrological conditions (depth of the river, currant, etc.). The sample is done by several individuals each of them responsible of a 1 metter square surface. All the kind of habitats within the site must be prospected. The full protocol from the RHP (Réseaux Hydrobiologique et Piscicole) can be found at this url (http://hebergement.u-psud.fr/solene.thepaut/spip.php?article2&var_mode=calcul).

There exists different motivations for the collect of fishing data by the AFB. There are discussed along with the protocol at the same url (http://hebergement.u-psud.fr/solene.thepaut/spip.php?article2&var_mode=calcul).
The main motivations for the collection of such data are the following.

- The AFB wants to keep track of the biomass of fishes in the rivers of the Loire basin. It is of general interest to have a dataset with measures of abundances of fishes each year.
- It allows to monitor species that are of ecological interest.
- The impact of natural events and human activities can be assessed efficiently.
- They can caracterise the inter-annual fluctuations of the species and estimates the long term tendencies thanks to the data collected.
- The dataset can be used to identify the factors that drive the composition of the biomass.

All the previous motivations have the same general purpose of monitoring and preservation of the biomass of fishes.

The spatial synchrony of populations is a point of interest for ecologists. One of the main reasons is the awareness of the potential mechanisms responsible for spatial synchrony. In [], there are three main reasons for spatial synchrony : dispersal among the population, competition or pretator/prey interactions between species and exogenous factor such as the weather or resources.
It would be of great interest to be able to determine the actual importance of each mechanism in the observed spatial synchrony among disjointed populations. An efficient review for existing work on spatial synchrony can be found in []. In this article, a statistical methods is provided to highlights groups of sites with synchroneous dynamics which is crucial to understand the link between global changes and population dynamics.
Unfortunately, this work does not exploit the particular structure of trees.
In our case, a tree seems to be the best choice for the representation of the river stream network on which we observe abundances of fishes.

Our motivations are the same as the authors in [], that is to understand and identify the main drivers of spatial synchrony by studying the variations of spatial synchrony in population through space but with constraints on the relationship between the different sites.

Our Tree Segmentation algorithms provides a way to identify groups of sites on which the variations of abundances of the species are synchroneous in the specific case where the data has to be represented by a tree.

**Description of the data** Abundances of several species of fishes have been measures yearly during 15 years, from 1990 to 2010, on 110 sites along the streams network according to the protocol introduced previously. The general goal is to find a partition of the river network into region where variations of the species abundances are homogeneous.

We focus our analysis on two specific species of fishes, the dowel (https://en.wikipedia.org/wiki/Dowel) and the minnow (https://en.wikipedia.org/wiki/Minnow) (denoted respectively GOU and VAI in the dataset for their French name : Goujon and Vairon) and analyze them separately.

More precisely, we have at our disposition the following datasets.

- Data on fish abundances in river. This folder contains data on the two most abundant fish species: GOU and VAI. The stations has been selected to be in Loire basin. The dataset contains 110 stations, 1282 fishing operations. 198′000 VAI and 148′000 GOU have been catched.
- Species abundance. This file contains the species abundance by station and fishing operation for GOU and VAI. A station is a site.
- Stream. This file contains the delineation of the french streams.

In a first sub-section, we detail how we construct the tree representing the river network. Then, we explain how we deal with missing data in our dataset. Finally, we give the results we obtain with algorithm 5 on the fishing data and analyze them.
The analysis performed in the last sub-section is only preliminary. The full analysis is on going with ecologists from the Museum d'Histoire Naturelle in Paris.

### 3.6.1 Constructing the tree

Using the coordinates of these sites along with the river network, we are able to build a tree rooted at the river mouth of Loire with 110 nodes. Unfortunately, the sites are not always located at meeting points of affluents. As a consequence, some nodes have more than 10 children. Such high degree nodes raise computational issues since, even with a pruning strategy, the corresponding set of partitions considered by PELT Tree is huge. Besides, partitions along the corresponding tree do not necessarily translate as a segmentation of the river network. Assume for instance that rivers $A$ and $B$ first meet together and then meet river $C$. If we do not have any observation between the meeting points, the corresponding node will have three children corresponding to these three rivers. A segmentation of the corresponding tree can put together rivers $A$ and $C$ while leaving $B$ aside, while this does not make sense on a segmentation of the river network. For these reasons, we decided to add so-called 'artificial' nodes at some intersections of the river network. Such nodes do not contain any information. That way, we give the algorithm a chance to further prune bad partitions and the tree is more representative of the true river network.

Another issue with the construction of the tree T is the presence of missing data. In some sites, we have yearly abundance measured between 1989 and 2014, while others have few or questionable observations. These bad sites compromise the good flow of our algorithm and we decide to consider

them as 'artificial' nodes. They are still in the tree for computational reasons but do not contain any information. Henceforth, we call T the tree reconstructed with this procedure.

In the raw data set, measurements were made on the year 1989 and 2014. We decided to remove the first years from our observations, because measurements were questionable in the first years. In preliminary analyses, we noticed that the data between 1998 and 2006 seem most trustworthy, with fewer missing data and outliers. This lead us to keep only the 110 sites on which we have at least 5 observations between 1998 and 2006. Among these 110 sites, 14% have one missing observation, and 6% have between 2 and 4 missing observations.

The final tree is of size 140, with 30 artificial nodes. For GOU, the rest of the tree is composed as 56 bad sites treated as artificial nodes in the algorithm and 54 sites with good observations. For VAI, we have 51 bad sites and 59 valid sites.

The stream network and the sites are represented on a map Figure 3.6. This map only features the true sites, that is to say non artificial sites on the river streams. The tree we created to represent the network with true and artificial nodes is represented by Figure 3.7. On Figures 3.9 and 3.8 we can see on the created tree which sites are true sites and which sites are artificial. Moreover, we can see which sites are treated as empty sites for each species.

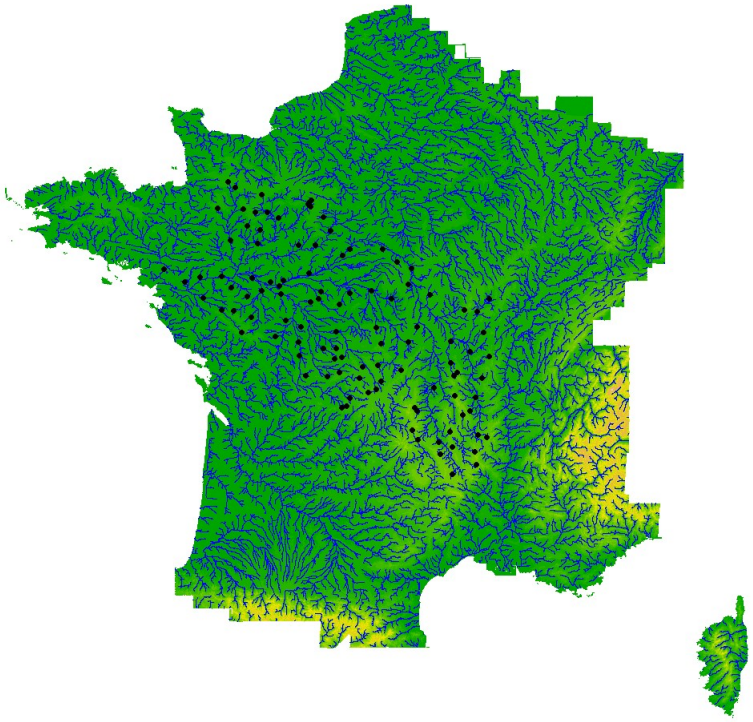


Figure 3.6 – Map of the river network on the Loire basin. The river streams are represented with blue lines and the sites with black dots. We use this map to create a tree representing the river network.

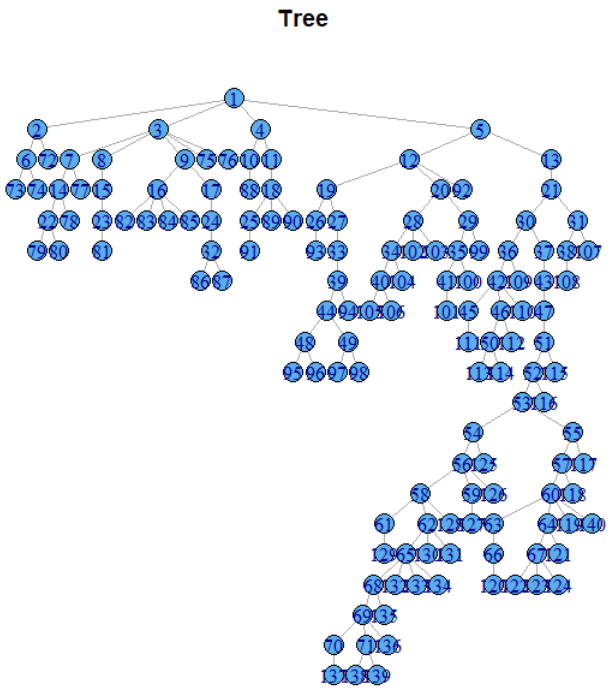



Figure 3.7 – Tree representing the river network. This tree has 140 nodes. The nodes are the sites on which we have observations plus some artificial nodes that allow us to better represent the river network. There is an edge between 2 nodes if they are linked by a river stream without any node between them.

On the view of the trees for VAI and GOU and the quality of the data, we present only the results for the minnow (VAI) species since there are more valid sites on the VAI tree 3.8. With barely 50% of valid sites, we consider that the data for the dowel species does not allow us to perform Tree Segmentation on a sufficient part of the river network.

We observe in Figure 3.8 that it is possible to get rid of some of the bad and artificial sites. If a bad or artificial site is a leaf, then it is treated as an empty site in our algorithm and does not provide any useful information to find the optimal partition. To delete these sites from the tree, we iterate the following procedure until all the leaves are valid sites: considering the tree T obtained from the previous step, we remove from T all the leaves that are bad or artificial sites. The tree obtained for VAI is represented in Figure 3.10.

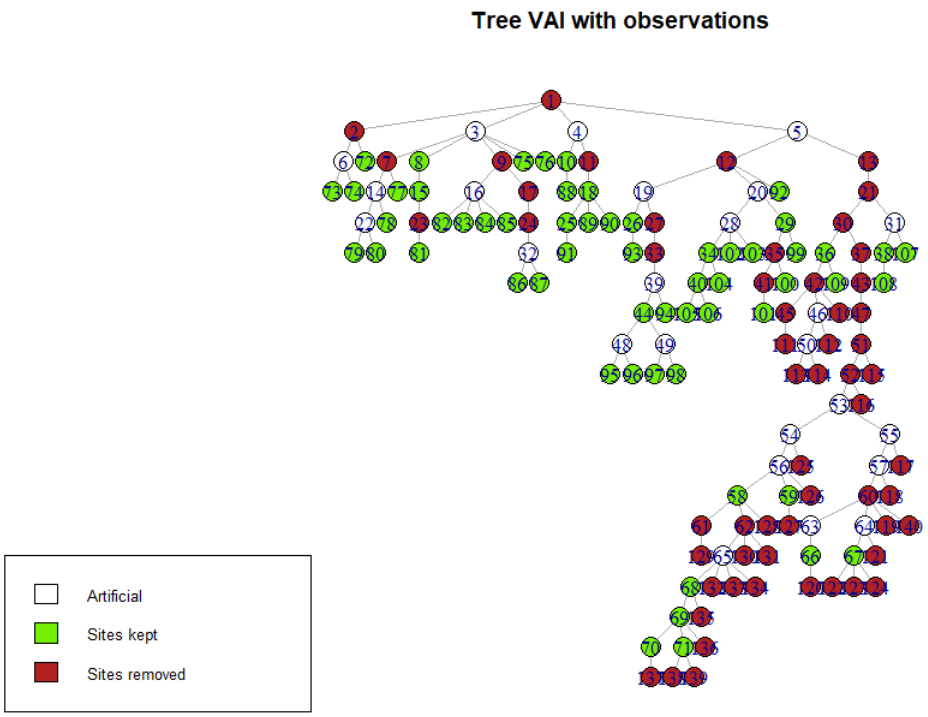



Figure 3.8 – Tree representing the sites and their types for the minnow dataset . The valid sites are represented as green nodes, the artificial nodes as white nodes and the bad sites as red nodes. We observe that several leaves of the tree are bad or artificial sites.

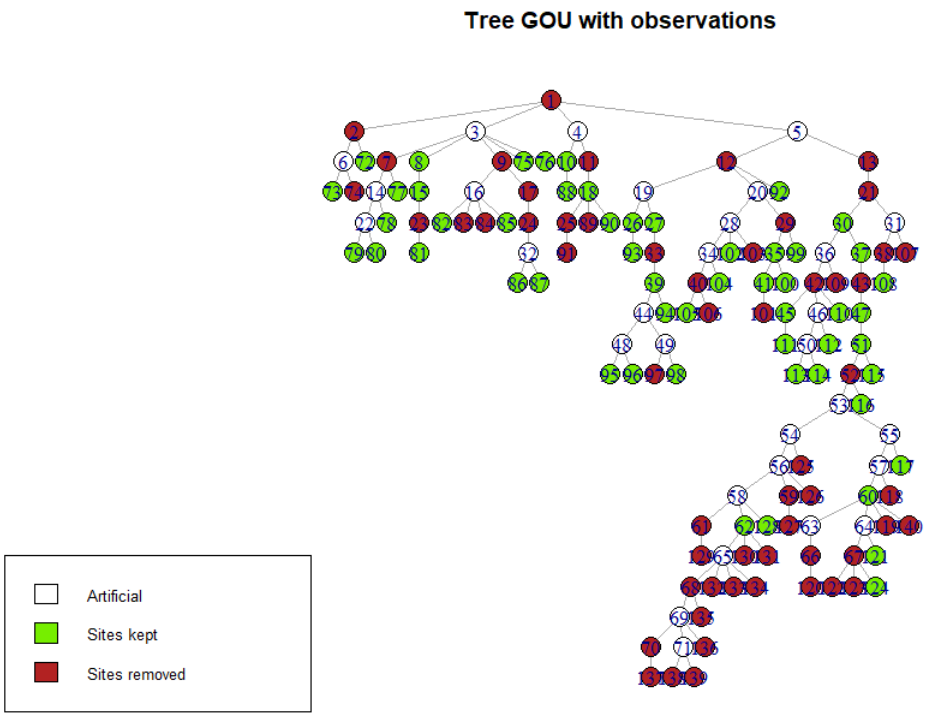



Figure 3.9 – Tree representing the sites and their types for the dowel dataset. The valid sites are represented as green nodes, the artificial nodes as white nodes and the bad sites as red nodes. We observe that several leaves of the tree are bad or artificial sites.

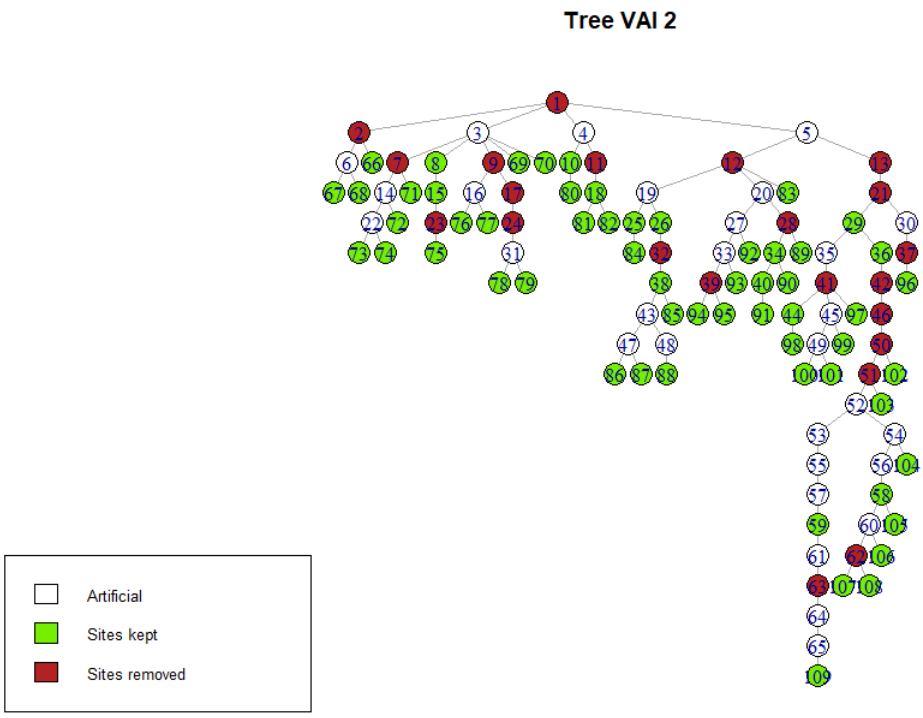



Figure 3.10 – Tree representing the sites and their types for the minnow dataset after removing the useless leaves. As in Figure 3.8, the valid sites are in green, the artificial sites are in white and the bad sites in red. We can see that some of the remaining nodes are still useless to estimate the partition of the tree.

On Figure 3.10, we notice that the empty sites that have only one child can also be removed since they add computational complexity to the algorithm without giving any information to find the optimal partition. For this step, we removed and reordered the sites by hand. The final tree for VAI is represented in Figure 3.11.

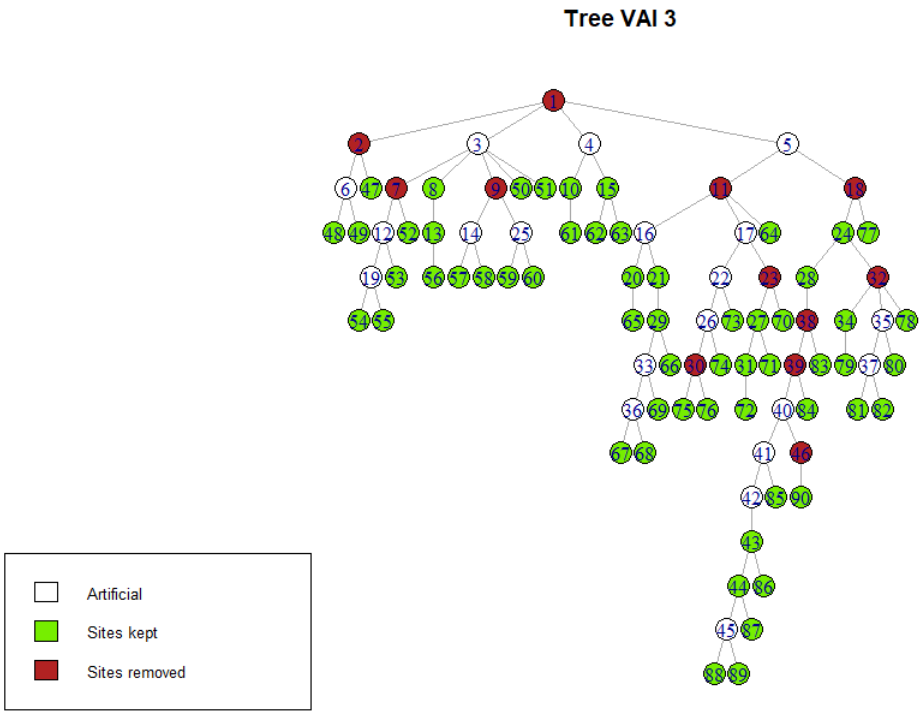



Figure 3.11 – Final tree $\mathrm{T}_{VAI}$ representing the river network with only useful nodes for the minnow datasets. The color code is the same as in Figures 3.8 and 3.10. This tree is used to estimate the partition of the river network.

These trees were represented with the igraph package in R [26].

### 3.6.2 Modeling the abundance distributions and accounting for missing Data

At each non empty site $s \in T$ and for a specific species, the abundance measurements are denoted $y_{s,j}$ for years $j = 1, \dots p$ with $p = 9$, the years ranging from 1998 to 2006. We model the data with multivariate Poisson distribution as in the numerical experiments (Section 3.5)

$$y_{s,j} \sim \mathcal{P}\mathrm{oi}\left(e^{\alpha_s + \mathbf{b}_{s,j}}\right) \ , \quad j = 1, \dots p \ , \tag{3.44}$$

where $\alpha_s$ stands for a site effect and $\mathbf{b}_{s,j}$ is a year dependent parameter. We are interested in finding the true minimal segmentation $\underline{\mathbf{P}}$ of the tree T such that the parameters $(\mathbf{b}_{s,j})$, $j = 1, \dots, p$ are constant on each region of the partition. With that formalism, the common coefficient $\mathbf{b}_{s,j}$ corresponds to a regional and year effect $\mathbf{b}_{S,j}$.

In the previous section, we have derived a cost function $C_S$ (3.39) corresponding to the minimum of the negative log-likelihood $\min_{\theta_S} -l(\mathbf{y}_S, \theta_S)$ (see Proposition 2 for more details). Unfortunately, we cannot use it here, because this cost function requires all the $y_{s,j}$ to be observed.

In our dataset, we have some years and sites where no counting session have occurred (14% of the sites with one missing observation and 6% with between 2 and 4 observations). Besides, we have artificial nodes without any observation.
We think of two approaches to solve this issue. The first one is very easy and fast to implement. It is not recommended when there are too many missing data but it gives arbitrary good enough estimates of the missing observations. The second method is more accurate and specific but depends on a gradient-descent like algorithm and needs to be performed at each step of the algorithms to

compute the penalised log-likelihood of the data. In all considerations for the expected high computational time of algorithms 5 or 8 on T and the relatively low number of missing observations on the selected years, we choose the first method for our application on missing data. Moreover, we have a higher risk of overloading the memory, which can be a problem with the fishing data, by using the second method. The first method is explained in the following and we introduce the second method in appendix 3.C.

We simulate the missing data with a simple and fast method that do not increase the complexity of the algorithms 5 and 8 or overload the memory. For each valid site $s \in \mathcal{V}(\mathrm{T})$, we denote $Y_s$ the set of all years such that there is an observation on the site $s$ and $S^{(j)}$ the set of all sites such that there is an observation for the year $j$. For each valid site $s$, we compute the mean $\bar{y}_s$ of the abundances of the species VAI over the years in $Y_s$.

$$\bar{y}_s = \frac{1}{|Y_s|} \sum_{j \in Y_s} y_{s,j}, \tag{3.45}$$

where $|Y_s|$ is the number of sites in $Y_s$.
Then, for each year $j$ between 1998 and 2006, we compute the mean $\bar{y}_j$ of the abundances of the species VAI over the valid sites $s \in S^{(y)}$.

$$\bar{y}_j = \frac{1}{|S^{(j)}|} \sum_{s \in S^{(j)}} y_{s,j}, \tag{3.46}$$

where $|S^{(j)}|$ is the number of year in $S^{(j)}$.
From the $\bar{y}_j$, we compute the mean $\bar{y}$ of the $\bar{y}_j$ on all the year $j$ between 1998 and 2006.

$$\bar{y} = \frac{1}{9} \sum_{j=1998}^{2006} \bar{y}_j. \tag{3.47}$$

We compute one coefficient $c_j$ for each year $j$ between 1998 and 2006 as it follows.

$$c_j = \frac{\bar{y}_j}{\bar{y}}. \tag{3.48}$$

Finally, the value of the missing observation on the site $s$ for the year $j$ denoted $y_{s,j}^{\text{missing}}$ is set equal to the round value of $c_j$ times $\bar{y}_s$.

$$y_{s,j}^{\text{missing}} = \lfloor c_j \bar{y}_s \rfloor \tag{3.49}$$

We replace all missing data by the value calculated <ith (3.49) prior to the algorithm. Then, we use the dataset with both real and simulated data in algorithms 8 or 5 as if there were no missing data.

### 3.6.3 Summary of the procedure

We analyze the abundances of the minnow and dowel species denoted VAI and GOU in our datasets. We have access to 3 files that allow us to construct the river network and to locate the fishing sites on the network. The first file is a Digital Elevation Model of France, the second and third files are shape files with information about the river network and the location of the sites. We use the openSTARS apckage in R [61] and follow the protocol from [90] to retrieve the river network and we obtain the map on Figure 3.6. As mentioned earlier, the fishing sites were not enough to construct a tree that represent well the river network. Hence, we created 30 artificial sites that allow us to construct the tree on Figure 3.7.

Once the general tree constructed, we perform a small exploratory analysis on the abundances datasets. The sites with few observations compromise the output of an algorithm aiming to find the

optimal partition of the river network. Such sites are declared as empty and have the same role as the artificial sites.
The dowel species only has 54 valid sites while the minnow species has 59. This represents 50% and 55% of the real sites. From a general point of view, the minnow dataset seems more promising than the dowel dataset, so from this point, we only describe the procedure and the results for the minnow species.

The next step is to construct the final tree T that will be used to find partition of the river network. We remove the useless nodes as explained in Subsection 3.6.1. The final tree for the minnow species has 90 nodes, 59 of them are valid sites, and 31 of them are artificial sites or sites with too few observations to be used as valid sites. We represent it in Figure 3.11.
In order to differentiate the valid sites from the other sites in the algorithm, we create a weight vector such that its $s$-th coordinate is equal to 1 if $s$ is a valid site and 0 otherwise.

Then, we compute the matrix of observations of the abundances of minnows on the sites of T. From the dataset containing the abundances of the minnow species we construct a matrix with 90 rows and 9 columns. Each row contains the time series of abundances of minnows between the year 1998 and 2006. The rows of the matrix correspond to the observations for the 59 valid sites and 31 empty rows for the other sites. We replace the missing data following the last procedure of Subsection 3.6.2.

We use Algorithm 5 with the implementation for the penalized minus log-likelihood for Poisson distribution as in (3.55). Considering the quality of the data, we decide not to use the model selection Algorithm 8 for two main reasons.

- The collected data are subject to a lot of noise and we know that the protocol to collect the data is not necessarily the same on different sites. By using Algorithm 8 the optimal $\beta$ found was 52 and the resulting partition had 42 breaks. This is explained by the bad quality of the data, and especially the noise of the observations. So, we decided to manually choose the $\beta$ parameter with Algorithm 5 in order to compute partitions of smaller size. We hope that by tuning ourselves the parameter $\beta$ of the linear penalty in Algorithm 5 we will obtain a more accurate partition of the river network.

- We do not know the true partition of the river network. By changing the $\beta$ parameter ourselves, we are able to compute thanks to Algorithm 5 partition of different size. These partitions can then be studied by ecologists who can justify the partitions with geographical and ecological elements and choose the one that makes more sense from an ecological point of view.

Hence, we use the linear penalty from Algorithm 5 with different values for the parameter $\beta$. We iterate 4 times the version modified for Poisson distribution of Algorithm 5, each time with a different $\beta$. The possible values for $\beta$ are 280, 1000, 1100 and 1200.
The values for $\beta$ are chosen after testing different possibilities. With the model selection algorithm, the optimal $\beta$ is equal to 52. We already saw that this value of $\beta$ gives a too large partition of the data. Hence, we decided to choose large values for $\beta$. With a $\beta$ greater than 1500, no breaks were found in the distribution and we noticed that for a $\beta$ between 300 and 900, the Algorithm 5 does not prune enough partition and overload the memory of the computer. At the end, we took 4 different values of $\beta$, smaller than 1500, greater than 50 and avoiding the range $300 - 900$. We make sure that the partitions obtained for each value are different from the others and that they bring new information for our analysis of the results.
The results are presented in the next subsection.

### 3.6.4 Results

We first present the sites and their disposition on the Bassin de la Loire. Figure 3.6 shows the disposition of the sites where the fishing data have been collected. Figure 3.12 shows the sites on

which we have relevant observations for the minnow species. The fact that we do not have a high proportion of valid sites can be problematic for the estimation of a partition of the river network. Indeed, we can observe that consecutive valid sites can be geographically far from one another and we assume that changes can occur in the distribution within the distance between two valid points. We insist here on the fact that the data for the dowel species have been put aside for now for quality reasons and that we only present the results obtained for the minnow species.

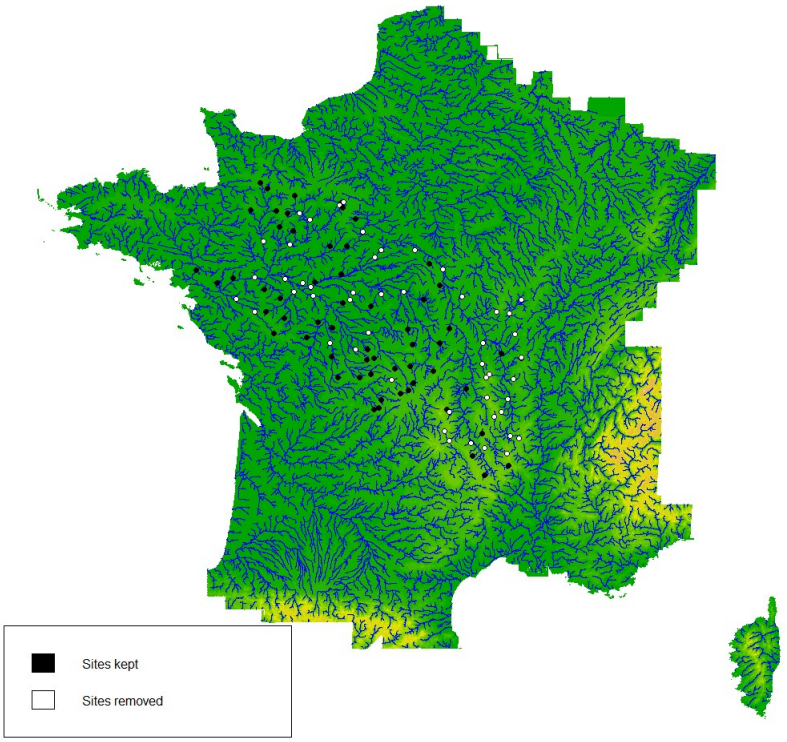



Figure 3.12 – Map of the river network and the location of the sites. On this map, only the real sites and their location are represented. Sites are colored depending on their relevance for the minnow species. The black sites are the sites on which we consider we have enough observation of abundances of minnow (valid sites). The white sites are sites on which we do not have enough observations. It appears on this map that only a little bit more than half of the sites are valid sites.

We apply the modification of Algorithm 5 to the Poisson distribution (3.39) for $\beta \in (280, 1000, 1100, 1200)$ on the matrix containing the time series in order to estimate a partition of T into homogeneous regions. We denote $\mathbf{P}_{\beta'}$ the estimated partition by our algorithm with the penalty parameter $\beta$ set to $\beta'$.

We can find in Figure 3.13 the results of the pruning on the tree T by our algorithm. For each value of $\beta$ and for each value of the number of descendants of $s \in \mathrm{T}$, we plot the average number of partitions in $S_{pr,1}(\mathrm{T}_s)$ versus the number of descendants. We notice that the pruning is more efficient when $\beta = 280$ or $\beta = 1200$. For values of $\beta$ between 300 and 900, the list of possible partitions is so huge that it overloads the memory, hence we were not able to use our algorithm for this range of $\beta$.

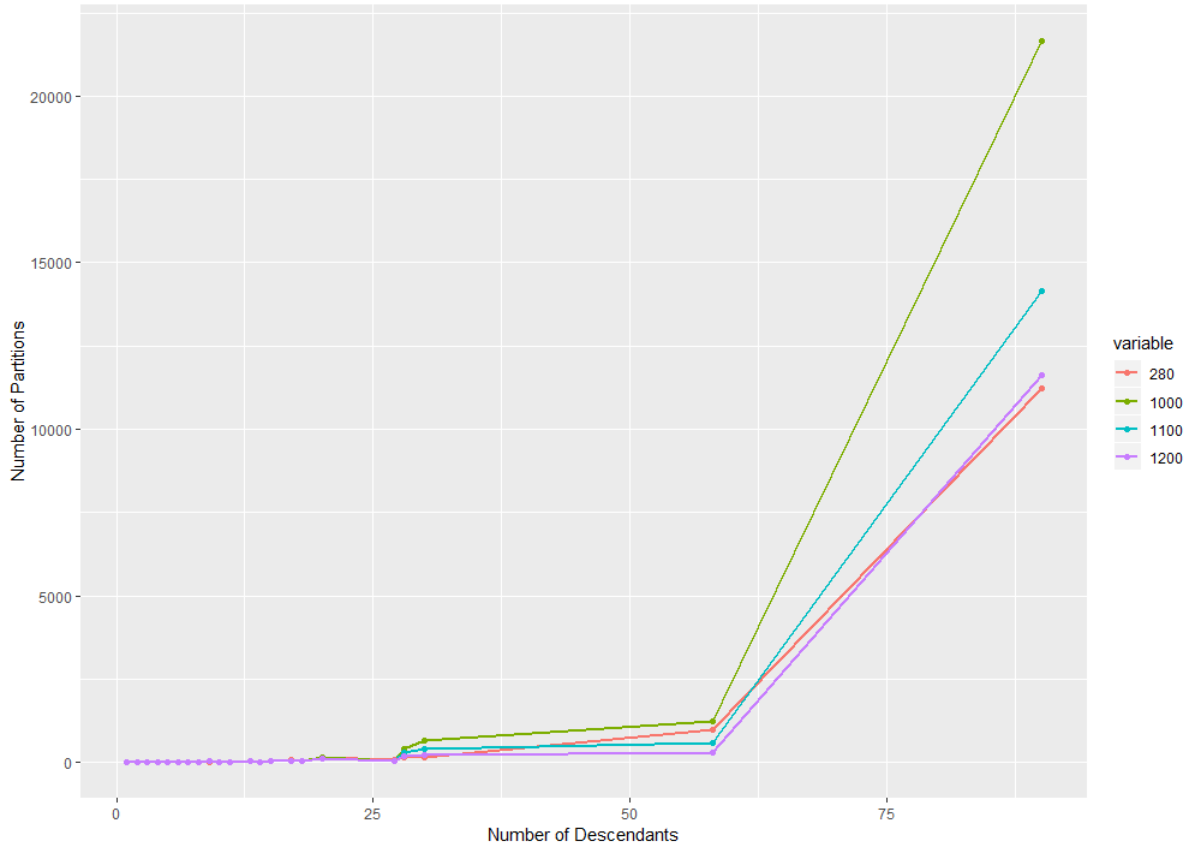
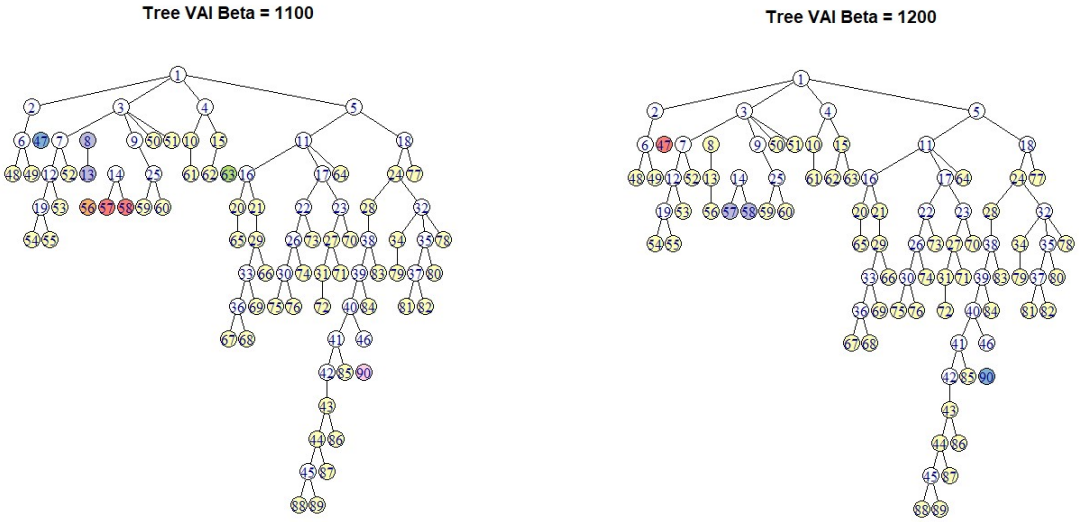



Figure 3.13 – Average number of possible partitions in $\mathcal{S}_{pr,1}(\mathrm{T}_s)$ for $s$ node of $\mathrm{T}_{VAI}$ versus number of descendants of the node $s$ of $\mathrm{T}_{VAI}$. We use Algorithm 5 adapted to a penalized minus log likelihood for Poisson distribution and with $\beta$ in $\{280, 1000, 1100, 1200\}$. For a certain number of descendants $n_1$, the number of possible partitions is the average number of partitions of $\mathcal{S}_{pr,1}(\mathrm{T}_s)$ for each $s$ such that its number of descendants is $n_1$.

The results for $\beta = 1200$ and $\beta = 1100$ are represented in Figure 3.14a and 3.14b. The partitions found for these values of $\beta$ are questionable. There are several sub-trees of size 1 and most of the breaks of these partitions are located between a leaf and its parent. It does not provide much information about the change in means of the distribution between nodes that are not leaves. These results may suggest that the parameter $\beta$ is too high.




(a) (b)

Figure 3.14 – In Figure 3.14a : Representation of $\mathbf{P}_{1100}$. There are 7 sub-trees. Nodes belonging to the same sub-tree of $\mathbf{P}_{1100}$ are represented in the same color. There is 1 large sub-tree and 6 sub-trees of size 3 or less. The small sub-trees are mostly composed of only 1 node which is a leaf of $T_{VAI}$. In Figure 3.14b : Representation of $\mathbf{P}_{1200}$. There are 4 sub-trees. Nodes belonging to the same sub-tree of $\mathbf{P}_{1200}$ are represented in the same color. There is 1 large sub-tree, 1 sub-tree of size 3 and 2 sub-trees of size 1.

The partition found with $\beta = 1000$ are presented in Figure 3.15, as for the two previous values of $\beta$, most of the sub-trees are of size 1 while the size of the partition is 8. We stated that this result is probably the consequence of the high value of $\beta$, but it also raises questions. If we chose a too high value of $\beta$, it means that only the most obvious changes in the mean of the distribution are detected by the algorithm. Hence, even if we lower the value of $\beta$ we expect the estimated partition to be made of several sub-trees of size 1. We represented the partition on the map of France in Figure 3.16. The analysis is the same, the partition is made of one large sub-tree, and several sub-trees of size 1 or 2. The small sub-trees are located at the extremities of the river network, which suggests that the abundances on the leaves follow a Poisson distribution with a different mean than on the rest of T.

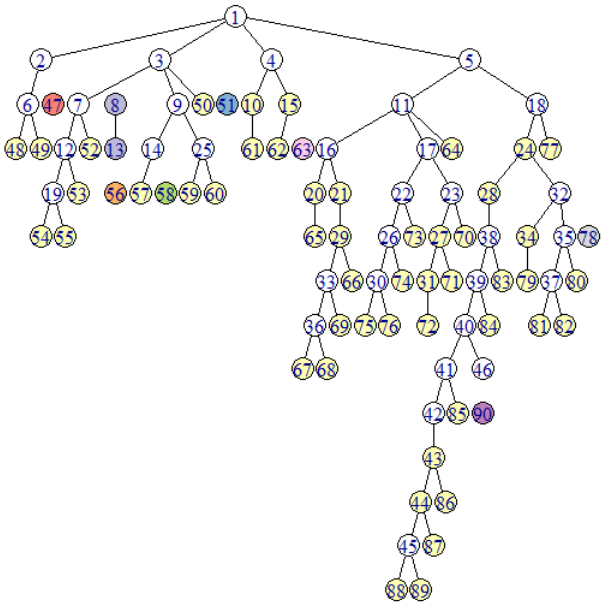



Figure 3.15 – Representation of $\mathbf{P}_{1000}$. There are 9 sub-trees. Nodes belonging to the same sub-tree of $\mathbf{P}_{1100}$ are represented in the same color. There is 1 large sub-tree and 8 sub-trees of size 2 or less. The white sites are the bad sites that were treated as empty sites during the algorithm.

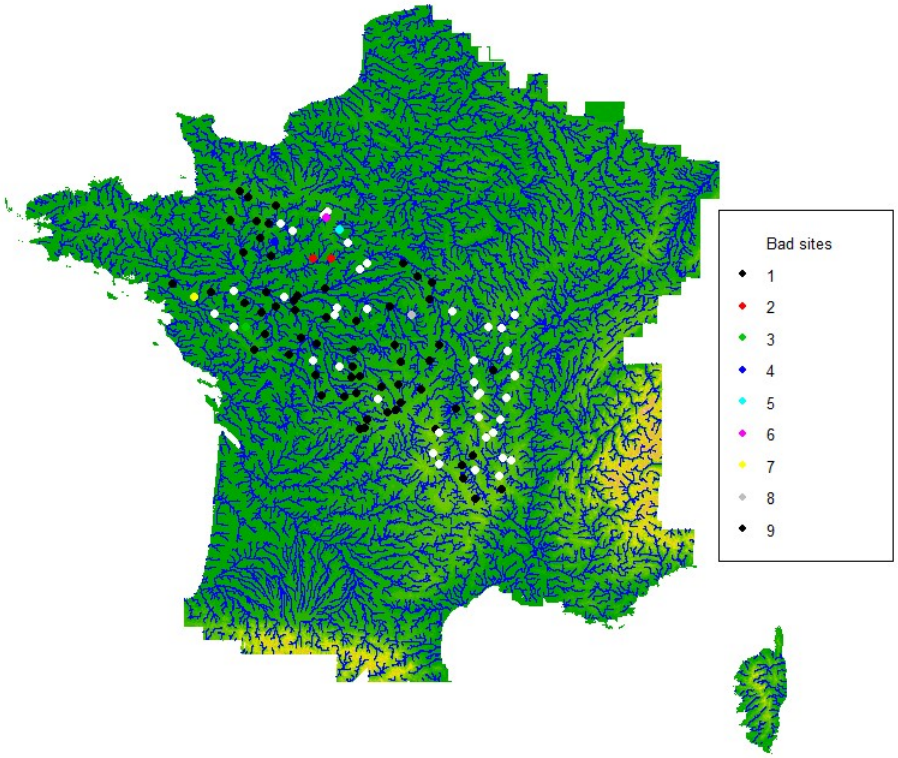



Figure 3.16 – Representation of $\mathbf{P}_{1000}$ of $T_{VAI}$ on the map of France. This representation confirms the analysis made thanks to Figure 3.15. The sites that belong to the same sub-tree of $\mathbf{P}_{1000}$ are colored with the same color. The white sites are the bad sites that were treated as empty sites during the algorithm. The blue lines represent the river network.

We now give the results for $\beta = 280$ which are represented on Figure 3.17. Since the value of $\beta$ is way lower than in the 3 first iterations, the size of the estimated partition is larger.

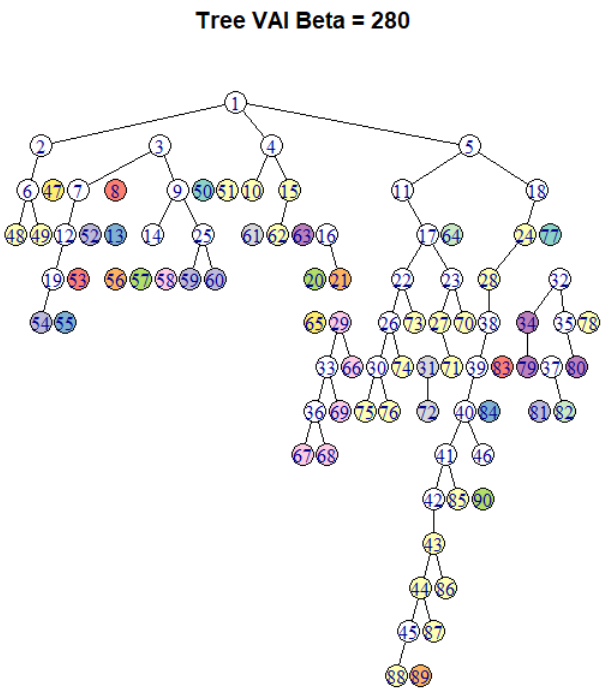



Figure 3.17 – Representation of $\mathbf{P}_{280}$. There are 30 sub-trees. Nodes belonging to the same sub-tree of $\mathbf{P}_{280}$ are represented in the same color. The white nodes are the bad nodes that are treated as empty sites during the algorithm. We observe 24 sub-trees of size 1. We also notice larger sub-trees with several valid nodes.

The partition $\mathbf{P}_{280}$ is made of 24 sub-trees of size 1 which validate the analysis done after the results for $\beta$ equals to 1000. The optimal partition of T is difficult to find with our data and the model 3.44. Yet, there are 4 sub-trees containing more than 3 valid sites. These 4 sub-trees may be a sign that there exist groups of sites on which the abundances of the minnow species are homogeneous. The analysis of the found partition has to be made collaboratively with ecologists who can justify the sub-trees with geographical or ecological information.

It seems that we lack information, and more importantly, we lack observations on the sites left aside because of the quality of the data. If we chose the value of the penalty coefficient $\beta$ to be between 300 and 900, the algorithm does not prune enough partitions at each step, and the memory space is overloaded by the list of possible partitions. With more memory space, it would be possible to compute partitions of different size and maybe find a more accurate partition of the data. Unfortunately, we saw that it is expected that for any value of $\beta$ below 1200, the estimated partition would be made of several small sub-trees containing the leaves of the tree. The results for $\beta = 280$ show that for small values of $\beta$, the estimated partition is made of small sub-trees and that it is rare to find sub-trees of size greater than 1. This can be imputed to different problems. We give three of them in the following.

- The large noise of the data. We expect the data to be very noisy because of the protocol of collection depending on the sites and because of the nature of the observations : time series of abundances. The observations do not only depend on geographical elements but also on the meteorology and natural phenomenon that cannot be translated on a map. The temporal aspect of the data makes the modelization of the data difficult.
- The quality of the data. We saw that even by keeping only the year with the most observations,

several sites do not contain enough observations to be used as valid sites in the algorithm. The valid sites are sometimes geographically far away from one another. Hence, it is not surprising that the means of their distribution are different. From 110 sites we ended up with only 59 valid sites on which we sometimes had to create observations for the missing years.

- The modelling of our data. In our model, we set that the observations are i.i.d. and follow a Poisson distribution. Even if the Poisson distribution is widely used to model abundances of species, it may not be suited to our data. Also, on each site $s$, the parameters $b_{s,j}$ and $b_{s,j'}$ for different years $j$ and $j'$ are chosen completely at random. Yet, it is probable that there exists a correlation between the abundances of a species on the same sites from one year to another.

On the other side, the estimated partition for $\beta = 280$ contains 5 sub-trees with more than 1 valid site. These preliminary results have to be studied and discussed with ecologists but it is possible that there exist groups of sites on which the abundances of minnows are homogeneous. It is encouraging for further work on such data. With more observations and a better modelization of our data, it may be possible to estimate an accurate partition of the river network.

### 3.6.5 Perspectives and Future Work

The results might not be conclusive yet but the full analysis and interpretation of the results with ecologist are ongoing. We only aimed to give a preliminary analysis of the spatial synchrony of a species on the river network.
Due to a lack of time and the bad quality of the data, we did not explore all the possibilities of our algorithm on the fishing data.

As a perspective, we could do the following

. Simulate numerical data on the tree T and run our algorithm on these data. This would confirm that the bad results with the minnow species is due to the bad quality of the data rather than the structure of the tree or the capabilities of our algorithm.

. Perform the same analysis with other species of fish in order to compare the partitions found and the mean of the distributions for each species.

. Implement and test a different method to deal with the missing data. For example as in the gradient descent like method presented in Appendix 3.C. This would allow us to improve slightly the quality of the data and use longer time series of abundances and more sites.

- Run the algorithms on the minnow dataset with a more powerful computer set up to reduce the computational time and add memory space. This way we can run our algorithm 5 with the critical values of $\beta$.

- Change the model of our data. The Poisson distribution and the chosen parameters might not modelize at best the true distribution of the abundances of fishes.

## 3.A Segmentation and break edges

We now prove the equivalence between partitioning the tree into $K$ connected components and removing $K-1$ edges in the tree T. This is equivalent to the following statement: if we remove any $K-1$ edges from a tree, then we obtain a forest with $K$ connected components.

We show it by induction. If we remove one edge to a tree, the resulting graph cannot be connected, otherwise the two endpoints $s$ and $t$ of the removed edge would have belonged to a cycle. Besides, the two connected components are the one containing $s$ and the one containing $t$. Indeed, any node is connected to $s$ in T: the first connected component is made of nodes whose corresponding path to $s$ does not cross the removed edge. The second connected components is made of node whose corresponding path to $s$ crosses this edge.

Now assume that we have removed $K-1$ edges. If we remove one more edge, it will be removed from one of the $K$ sub-trees of $T$ and will there split it into two smaller sub-trees. The result follows.

## 3.B Proofs of Lemma 15, Theorem 2, and Propositions 1 and 2

*Proof of Lemma 15.* Since $\widehat{\mu}_{\mathbf{P}}$ is a least-square estimator on the space $V_{\mathbf{P}}$, its risk satisfies

$$\mathbb{E}\left[\|\widehat{\mu}_{\mathbf{P}} - \mu^*\|_T^2\right] = \|\,\mathbb{E}[\widehat{\mu}_{\mathbf{P}}] - \mu^*\|_T^2 + \sigma^2 \dim(V_{\mathbf{P}})\ .$$

The expectation $\mathbb{E}[\widehat{\mu}_{\mathbf{P}}]$ coincides with the projection $\mu_{\mathbf{P}}$ of $\mu^*$ onto $V_{\mathbf{P}}$. The dimension of $V_{\mathbf{P}}$ equals $p|\mathbf{P}|$. Indeed, any $\mu$ in $V_{\mathbf{P}}$ is uniquely represented by a collection of $|\mathbf{P}|$ vectors $\theta_1,\dots,\theta_{|\mathbf{P}|}$ of size $p$ through the identity $\mu_t = \sum_{i=1}^{|\mathbf{P}|} \theta_i \mathbf{1}_{t\in S_i}$. As a consequence $V_{\mathbf{P}}$ is in bijection with $\mathbb{R}^{p|\mathbf{P}|}$ and the result follows. □

*Proof of Theorem 2.* This result is an application of Birgé Massart model selection theorem in Gaussian models (see Theorem 4.2 in [87]). The estimators $\widehat{\mu}_{\mathbf{P}}$ are least-squares estimators of $\mu^*$ in $V_{\mathbf{P}}$ whereas $\mu_{\mathbf{P}}$ corresponds to the orthogonal projection of $\mu^*$ on $V_{\mathbf{P}}$. Let us first translate this theorem with our notation.

Let $(x_{\mathbf{P}})$, $\mathbf{P}\in\mathcal{P}$ be a collection of positive numbers such that $\sum_{\mathbf{P}\in\mathcal{P}} e^{-x_{\mathbf{P}}} = \Sigma < \infty$. Let $\eta > 1$ and assume that

$$\mathrm{pen}(\mathbf{P}) \geq \eta\sigma^2(\sqrt{\dim(V_{\mathbf{P}})} + \sqrt{2x_{\mathbf{P}}})^2\ . \tag{3.50}$$

then, the penalized least-squares estimator $\widehat{\mu}_{\widehat{\mathbf{P}}}$ satisfies

$$\mathbb{E}[\|\widehat{\mu}_{\widehat{\mathbf{P}}} - \mu^*\|_T^2] \leq C_\eta \left[\inf_{\mathbf{P}\in\mathcal{P}} \left(\|\mu_{\mathbf{P}} - \mu^*\|_T^2 + \mathrm{pen}(\mathbf{P})\right) + (1+\Sigma)\sigma^2\right]\ . \tag{3.51}$$

where $C_\eta$ only depends on $\eta$.

Since (3.51) has the same form as 3.35, we only to find a family $x_{\mathbf{P}}$ such that any penalty satisfying Condition (3.34) in the statement of the theorem also satisfies (3.50).

Fix an integer $1 \leq K \leq n$. We first compute the number of partitions $\mathbf{P}$ of size $K$. As explained in the introduction, a partition $\mathbf{P}$ is one to one with the corresponding set of break edges (the edges that cross two sub-trees of the partition). Besides, a partition $\mathbf{P}$ of size $K$ has exactly $K-1$ break edges and conversely, any set of $K-1$ edges generates a partition $\mathbf{P}$ of size $K$. Since there are exactly $n-1$ edges in a tree of size $n$, the number of partitions of size $K$ is $\binom{n-1}{K-1}$. Fix any $\zeta \leq 1$. Then, for any $|\mathbf{P}| \geq 1$ we set

$$x_{\mathbf{P}} = (|\mathbf{P}|-1)\left[1 + \zeta + \log\left(\frac{n-1}{|\mathbf{P}|-1}\right)\right] \tag{3.52}$$

and we set $x_{\mathbf{P}} = 0$ for the only partition of size 1. Equipped with this choice of $x$, we have

$$\begin{aligned}\Sigma = \sum_{\mathbf{P}\in\mathcal{P}} e^{-x_{\mathbf{P}}} &= 1 + \sum_{K=2}^{n} \binom{n-1}{K-1} e^{-\zeta(K+1)} \left(\frac{(n-1)e}{K-1}\right)^{K-1} \\ &\leq 1 + \sum_{K=2}^{n} e^{-\zeta(K-1)} \leq \frac{e^{\zeta}}{e^{\zeta}-1} ,\end{aligned}$$

where we used in the second that $\log\left(\binom{n-1}{K-1}\right) \leq (K-1)\log\left(\frac{e(n-1)}{K-1}\right)$.

We have already observed that the dimension of $V_{\mathbf{P}}$ equals $p|\mathbf{P}|$. We are now in position to state again the penalty Condition (3.50)

$$\mathrm{pen}(\mathbf{P}) \geq \eta\sigma^2 \left[\sqrt{p|\mathbf{P}|} + \sqrt{2(|\mathbf{P}|-1)\left[1+\zeta+\log\left(\frac{n-1}{|\mathbf{P}|-1}\right)\right]}\right]^2 .$$

Since for any $\theta > 0$, $(a+b)^2 \leq (1+\theta)a^2 + (1+\theta^{-1})b^2$. Condition (3.50) is also satisfied if

$$\mathrm{pen}(\mathbf{P}) \geq \eta\sigma^2 \left[(1+\theta)(p|\mathbf{P}|) + 2(1+\theta^{-1})(|\mathbf{P}|-1)\left((1+\zeta)+\log\left(\frac{n-1}{|\mathbf{P}|-1}\right)\right)\right] .$$

The result follows.

□

*Proof of Proposition 1.* Write $(\widehat{\mathbf{P}}_l)_{l\geq 1}$ for the sequence of partitions in Algorithm 8. By definition of $\widehat{\mathbf{P}}_{l+1}$, we have

$$\mathrm{Cost}_{\mathrm{T}}(|\widehat{\mathbf{P}}_{l+1}|) + |\widehat{\mathbf{P}}_{l+1}| f'(|\widehat{\mathbf{P}}_l|) \leq \mathrm{Cost}_{\mathrm{T}}(\widehat{\mathbf{P}}_l) + |\widehat{\mathbf{P}}_l| f'(|\widehat{\mathbf{P}}_l|) \tag{3.53}$$

By strict concavity of $g$, we have $f'(|\widehat{\mathbf{P}}_l|)(|\widehat{\mathbf{P}}_{l+1}| - |\widehat{\mathbf{P}}_l|) > \mathrm{pen}(\widehat{\mathbf{P}}_{l+1}) - \mathrm{pen}(\widehat{\mathbf{P}}_l)$ as soon as $|\widehat{\mathbf{P}}_{l+1}| \neq |\widehat{\mathbf{P}}_l|$. Gathering the two previous inequalities leads us to

$$\mathrm{Cost}_{\mathrm{T}}(|\widehat{\mathbf{P}}_{l+1}|) + \mathrm{pen}(\widehat{\mathbf{P}}_{l+1}) < \mathrm{Cost}_{\mathrm{T}}(\widehat{\mathbf{P}}_l) + \mathrm{pen}(\widehat{\mathbf{P}}_l) .$$

If $|\widehat{\mathbf{P}}_{l+1}| = |\widehat{\mathbf{P}}_l|$, this implies that we are at the last step of the algorithm. This also implies that $\mathrm{pen}(\widehat{\mathbf{P}}_{l+1}) = \mathrm{pen}(\widehat{\mathbf{P}}_l)$. We conclude from (3.53), that

$$\mathrm{Cost}_{\mathrm{T}}(|\widehat{\mathbf{P}}_{l+1}|) + \mathrm{pen}(\widehat{\mathbf{P}}_{l+1}) \leq \mathrm{Cost}_{\mathrm{T}}(\widehat{\mathbf{P}}_l) + \mathrm{pen}(\widehat{\mathbf{P}}_l) .$$

Let us turn to the second result of the proposition. Consider any minimizer $\mathbf{P}' \in \arg\min_{\mathbf{P}\in\mathcal{P}}[\mathrm{Cost}_{\mathrm{T}}(\mathbf{P}) + f'(|\widehat{\mathbf{P}}_{\mathrm{pen}}|)|\mathbf{P}|]$. If $|\mathbf{P}'| \neq |\widehat{\mathbf{P}}_{\mathrm{pen}}|$, this implies by the previous arguments that the penalized criterion of $\mathbf{P}'$ is strictly smaller than that of $\widehat{\mathbf{P}}_{\mathrm{pen}}$, which contradicts its definition. We have therefore $|\mathbf{P}'| = |\widehat{\mathbf{P}}_{\mathrm{pen}}|$. And $\mathbf{P}'$ belongs to the collection $\arg\min_{\mathbf{P}\in\mathcal{P}:\,|\mathbf{P}|=|\widehat{\mathbf{P}}_{\mathrm{pen}}|} \mathrm{Cost}_{\mathrm{T}}(\mathbf{P})$ that contains $\widehat{\mathbf{P}}_{\mathrm{pen}}$.

□

*Proof of Proposition 2.* Define the cost function $C'_S$ corresponding to the minimum of the negative log-likelihood of $\theta_S$ with respect to the observations $\mathbf{y}_S$

$$C'_S = \min\left\{ -l(\mathbf{y}_S, \theta_S)(\alpha_s :_{s\in S} \in \mathbb{R}^S, (\mathbf{b}_{j,S})_{j=1,\dots,p} \in \mathbb{R}^p, \sum_{j=1}^{p} e^{\mathbf{b}_{S,j}} = 1 \right\} . \tag{3.54}$$

In the above definition, we added the constrain $\sum_{j=1}^{p} e^{\mathbf{b}_{S,j}} = 1$ because the model is over-parametrized. It turns out that $C'_S$ has a simple explicit formula as stated by the following lemma.

**Lemma 16.** *The cost $C'_S$ defined in (3.54) is equal to*

$$C'_S = \sum_{s\in S}\sum_{j=1}^{p} y_{s,j}\log\left(\frac{\sum_{t\in S}\sum_{i=1}^{p} y_{t,i}}{\sum_{t\in S} y_{t,j}}\right) + \sum_{s\in S}\sum_{j=1}^{p}\left(y_{s,j}\left(1-\log\sum_{i=1}^{p} y_{t,i}\right)+\log(y_{s,j}!)\right) \quad . \tag{3.55}$$

In view of (3.55), we have

$$\sum_{S\in\mathbf{P}} C'_S - C_S = \sum_{s\in T}\sum_{j=1}^{p}\left(y_{s,j}\left(1-\log\sum_{i=1}^{p} y_{t,i}\right)+\log(y_{s,j}!)\right) \quad ,$$

which does does not depend on $\mathbf{P}$. The result follows. It remain to prove the lemma.

The negative log-likelihood of $\theta_S$ with respect to the observations $\mathbf{y}_S$ is given by

$$-\,l(\mathbf{y}_S,\theta_S) = \sum_{t\in S}\sum_{j=1}^{p}\left(\log(y_{t,j}!) - y_{t,j}(\alpha_t+\mathbf{b}_j) + e^{\alpha_t+\mathbf{b}_{S,j}}\right). \tag{3.56}$$

The partial derivative of (3.56) with respect to $\alpha_t$ is

$$\partial_{\alpha_s}(-l(\mathbf{y}_S,\theta_S)) = -\sum_{j=1}^{p} y_{s,j} + e^{\alpha_s}\sum_{j=1}^{p} e^{\mathbf{b}_{S,j}}.$$

In particular, for any feasible $(\mathbf{b}_{S,j})_{j=1,\dots,p}$, the negative log-likelihood (3.56) is minimized with respect to $\alpha_s$ for

$$\hat{\alpha}_s = \log\left(\sum_{j=1}^{p} y_{s,j}\right). \tag{3.57}$$

Symmetrically, the partial derivative with respect to $\mathbf{b}_{S,j}$ of (3.56) at $(\hat{\alpha}_s,\mathbf{b}_{S,j})_{s\in S,j=1,\dots,p}$ is

$$\partial_{\mathbf{b}_j}(-l(\mathbf{y}_S,(\hat{\alpha}_t,\mathbf{b}_{S,j})_{s\in S,j=1,\dots,p})) = -\sum_{s\in S} y_{s,j} + e^{\mathbf{b}_{S,j}}\sum_{s\in S}\sum_{i=1}^{p} y_{s,i}.$$

We observe that the above partial derivative is equal to 0 for

$$\hat{\mathbf{b}}_{S,j} = \log\left(\frac{\sum_{s\in S} y_{t,j}}{\sum_{s\in S}\sum_{i=1}^{p} y_{t,i}}\right), \tag{3.58}$$

which also fulfills the condition $\sum_{j=1}^{p} e^{\hat{\mathbf{b}}_{S,j}} = 1$. Hence, the constrained minimum (3.54) is achieved for $(\hat{\alpha}_t,\hat{\mathbf{b}}_{S,j})_{t\in S,j=1,\dots,p}$ given by (3.57) and (3.58). Plugging these two formulas into (3.56) gives (3.55). □

## 3.C Likelihood with NA - descent gradient like algorithm

We present here a method that aims to solve the problme of missing data in a dataset.
This method has not been implemented yet in one of our algorithm but will be in future work.

Below, we write $A$ for the set of artificial nodes. In this case, we compute the likelihood of the parameters only based on the observed data. Denote $Y_s$ the set of all years $j$ such that there is an observation on the site $s$ at year $j$, the negative log-likelihood of the parameters $(\theta_{\mathbf{P}},\mathbf{P})$ in presence of missing data is given by

$$-\,l\left(\mathbf{y},\theta_{\mathbf{P}},\mathbf{P}\right) = \sum_{S\in\mathbf{P}} -l\left(\mathbf{y}_{S\setminus A},\theta_{S\setminus A},S\right) = \sum_{S\in\mathbf{P}}\sum_{s\in S\setminus A}\sum_{j\in Y_s}\left(\log(y_{s,j}!) - y_{s,j}\left(\alpha_s+\mathbf{b}_{S,j}\right) + e^{\alpha_s+\mathbf{b}_{S,j}}\right) \quad . \tag{3.59}$$

For this data set, we use therefore the cost function $C_S = \min_{\theta_{S\setminus A}} -l\left(\mathbf{y}_{S\setminus A}, \theta_{S\setminus A}, S\right)$. Contrary to Section 3.5, we do not have explicit formulas for this minimum. Yet, the function $\theta_{S\setminus A} \to -l\left(\mathbf{y}_{S\setminus A}, \theta_{S\setminus A}, S\right)$ is convex so it is easily amenable by a coordinate descent approach, described below.

Since the model is over-parametrized, we use the following linear constrain on the parameters $\mathbf{b}_{S\setminus A,j}$

$$\sum_{j\in Y_{S\setminus A}} \mathbf{b}_{S\setminus A,j} = 0\ . \tag{3.60}$$

This differs from the constraint used in the proof of Proposition 2, but (3.60) has the nice feature to be linear. Let us denote by $S^{(j)}$ the set of all sites $s$ with an observation at year $j$. Then, for $S \in \mathbf{P}$ and for a fixed vector $(\alpha_s)_{s\in S\cap S^{(j)}}$, the minimum of (3.59) with respect to $\mathbf{b}_{S,j}$, under the constraint (3.60) is achieved for

$$\hat{\mathbf{b}}_{S,j} = \log\left(\frac{\lambda_S + \sum_{s\in S\cap S^{(j)}} y_{s,j}}{\sum_{s\in S\cap S^{(j)}} \mathrm{e}^{\alpha_s}}\right), \tag{3.61}$$

with $\lambda_S$ solution to

$$\sum_{j\in Y_S} \log\left(\frac{\lambda_S + \sum_{s\in S\cap S^{(j)}} y_{s,j}}{\sum_{s\in S\cap S^{(j)}} \mathrm{e}^{\alpha_s}}\right) = 0.$$

Symmetrically, for $S \in \mathbf{P}$, $s \in S \setminus A$ and a fixed vector $(\mathbf{b}_{S\setminus A,j})_{j\in Y_s}$, the minimum of (3.59) with respect to $\alpha_s$ is achieved for

$$\hat{\alpha}_s = \log\left(\frac{\sum_{j\in Y_s} y_{s,j}}{\sum_{j\in Y_s} \mathrm{e}^{\mathbf{b}_{S\setminus A,j}}}\right). \tag{3.62}$$

Minimizing (3.59) under the constraint (3.60) by coordinate gradient descent, simply amounts to alternate the minimizations (3.61) and (3.62).

# CHAPTER 4

# IS INTERSPECIFIC SYNCHRONY CORRELATED TO LONG-TERM ABUNDANCE TRENDS OR TO SOME SPECIES TRAITS?

## 4.1 Introduction

Community stability has been extensively studied both theoretically and experimentally, especially in plant communities to examine the relationship between species diversity and stability of biomass. But the diversity - stability relationship has received less attention in natural communities, particularly in animals. However, in the context of global changes and given the ecosystem services provided by animal communities such as seed dispersal, pollination or pest control, it is of a primary importance to understand the drivers of community, hence ecosystem services, stability and the impact of habitat degradation or climate change.

We focus in this chapter in one of the main driver of community stability: synchrony among species temporal fluctuations. In Ecology, two species are said to be synchronous, if their variations of relative abundance show similar patterns. In mathematical terms, synchronous species are species whose inter-annual abundances are strongly correlated.

We investigate in Section 4.3 and 4.4 two main questions

- is there a link between synchrony of populations and long-term temporal trends?
- is there a link between synchrony of populations and traits similarity?

We investigate both questions at a regional scale, as synchrony among species at the regional scale is predicted to be one of the main mechanism controlling regional stability. This scale also corresponds to a typical scale in management and conservation policies of the ecosystems.

The hypothesis motivating the first question is the following. If species with similar long-term temporal trends are in the same synchronous groups, we can expect the extinction of all the species in some groups of synchrony, which would impact the compensatory dynamics and the stability of communities at a regional scale. As for the second question, our aim is to understand how the synchronous species are structured in terms of traits. The motivation is two-fold. First, as some traits are related to ecosystem services, we want to understand if these traits are spread among different synchronous groups. If some functional traits are well spread among the different groups, it is positive in terms of stability of ecosystem services. Second, we investigate if some traits can be the driver of synchrony, as it may impact conservation policies.

We explore these questions, by analyzing a butterfly abundance data set covering the United Kingdom between 2006 and 2015. This data set and the main statistical tools are described in next section. The link between synchrony and long-term temporal trends is investigated in Section 4.3. The link between synchrony and similarity is explored in Section 4.4. Some appendices then gathers some methodological details complementing Section 4.2.

## 4.2 Material and methods

### 4.2.1 UKBMS Data and temporal dynamics extraction

#### UK Butterfly Monitoring Data

The data set is extracted from the UK Butterfly Monitoring Scheme (UKBMS, http://www.ukbms.org/). This participative science program, consists in butterfly counts in sites located all over the UK. Each site is divided into transects located in different local habitat or management units, for a total length of about 2-4km. The spatial position of transects is fixed across years. Butterflies are recorded every year between the beginning of April and the end of September in a fixed-width band (5m wide) along transects. Transects walks are undertaken between 10.45am and 3.45pm with the following requirements for weather conditions: (1) temperature above 13°C (in northern upland areas this may be reduced to 11°C); (2) between 13-17°C, a transect may be walked providing there is at least 60% sun and (3) wind speed below 5 on the Beaufort scale. We perform our analyses on 53 species in 1,859 sites over the time period 2006–2015. Species were selected on the following basis: each species has been recorded in at least two sites every year between 2006 and 2015, and for each species, at least one site has made records every year from 2006 to 2015.

The UKBMS data are presence-only data, so we have to handle carefully the non-recording of a species a given day at a given site. There can be two reasons for not observing this species during this counting session. Either the species is not present at all on the site (absent), or, it is present, but its presence has not been detected during this session. In the first case, as it is not meaningful to follow the temporal variations of an absent species, the species is recorded as absent on this site and is not considered further. In the second case, it is important to record the non-observation of the species during a session. Hence, when a species was not observed during a monitoring event at a given site and day, but was observed at least once in this site, the count of the species is set to 0 at this day and site.

#### Phenology and inter-annual relative abundance variation

The analysis of the data requires a preprocessing in order to extract inter-annual relative abundance variations. Actually, the counts at each site cannot be directly used as a proxy for relative abundances, for the following reasons. As data is collected by volunteers, each site is visited only a small number of days per year, and the days and number of days may vary from one year to the other. So we cannot directly compare the counts from one year to the other. This inhomogeneity could be easily corrected, if the abundances of butterflies were not mainly driven by strong seasonal fluctuations. The intra-annual seasonal fluctuation of a species is called phenology, and it accounts for fluctuations due to seasonal environmental variations and life cycle events. As the phenology of a species may strongly vary from one year to the other, there is no direct correction that we can apply to data.

In order to extract inter-annual abundance variations from these data, [109] proposes a statistical model taking into account both sources of variations: inter-annual variations and phenology fluctuations. The general recipe of their method is to recover the shapes of the phenologies from (implicit) aggregation of the counts from the multiple sites of the region and to correct the counts with the estimated phenologies in order to recover the inter-annual abundance variations. The assumption underlying this approach is that the phenology for a given species and year do not vary across all the sites of a region. Due to the known effect of meteorological conditions on phenology [29], [97], [106], [112], [113], [126], we partition sites according to bioclimatic regions (see Figure 4.1) as defined by the UK Meteorological Office [1], We therefore estimate one phenology per species, bioclimatic region and year.

The statistical model proposed by [109] is a semi-parametric generalized linear model with log link and Poisson distribution. Writing $N_{syd}^{(i)}$ for the count for the species $i$ at location $s$, the day $d$

Figure 4.1 – Map of the 393 sites with at least two species observed every year between 2006 and 2015 and used to compute interspecific synchrony, with the different climatic regions used to estimate annual species abundances and yearly phenology.

of the year $y$, the count $N_{syd}^{(i)}$ is modeled as a Poisson distribution with mean

$$\mathbb{E}\left[N_{syd}^{(i)}\right] = \exp\left(\gamma_{sy}^{(i)} + f^{(i)}(d)\right). \tag{4.1}$$

The phenology is represented by $f^{(i)}(d)$ and it fulfills the standardisation

$$\sum_{d} \exp(f^{(i)}(d)) = 1.$$

The model is fitted by maximizing the likelihood with $f$ expanded on a cubic-spline basis.

The parameter $\gamma_{sy}^{(i)}$ can be decomposed into two parts

$$\gamma_{sy}^{(i)} = \alpha_{s}^{(i)} + \beta_{sy}^{(i)},$$

where $\alpha_{s}^{(i)}$ is a site effect, constant through time and $\beta_{sy}^{(i)}$ represents the inter-annual abundance variation. The $\beta$ coefficients are required to fulfill the standardisation

$$\sum_{y} \beta_{sy}^{(i)} = 0.$$

In this chapter, we slightly change the modeling of [109] by replacing the Poisson distribution by a Negative Binomial distribution. Actually, there is a strong overdispersion in the observed counts and in his PhD Thesis T. Olivier [95] shows that we get a much better fit with the Negative Binomial distribution rather than the Poisson distribution.

### Regional synchrony

**Regional inter-annual variations.** As we are interested in regional synchrony, we aggregate the local inter-annual abundance variations $\beta_{sy}^{(i)}$ into a regional inter-annual abundance variation $\beta_y^{(i)}$ by taking the median of the indices

$$\beta_y^{(i)} = \text{median}\{\beta_{sy}^{(i)} : s \in \text{UK}\}. \tag{4.2}$$

The median is preferred to the mean, in order to mitigate the impact of possible outliers. We emphasize that the adjective "regional" refer here to the UK and not to the bioclimatic (sub)regions used for computing the phenologies.

**Correction for long-term trends.** Significant long-term temporal trends (strong expansion or decline) can occur both at a local or UK scale [36]. In case of strong long-term temporal trends, the synchrony measured on the basis of inter-annual variations will simply reflect similar trends. Typically, two species will be synchronous because of a similar decline or expansion across time and not because of non-systematic inter-annual variations. We illustrate this phenomenon in Figure 4.8 Appendix 4.A. To avoid the confounding effect of temporal trends, we correct the regional inter-annual variations $\beta_y^{(i)}$ by removing the trends. To do so, we regress linearly the regional inter-annual variations $(\beta_y^{(i)} : y = 2006, \dots, 2015)$ with respect to time $y$ and we retrieve the residuals $\tilde{\beta}_y^{(i)}$ for each species and year, referred to as the trend-corrected regional inter-annual variations below.

**Synchrony indices.** As explained in the introduction, two species are said to be synchronous, if their abundances share some similar patterns of (non-systematic) inter-annual variation. In order to quantify this property, we consider the synchrony index between two species $i$ and $j$, defined as the Spearman-$\rho$ correlation $\rho_{ij}$ between the two time series $(\tilde{\beta}_y^{(i)} : y = 2006, \dots, 2015)$ and $(\tilde{\beta}_y^{(j)} : y = 2006, \dots, 2015)$. We recall that the Spearman-$\rho$ correlation is obtained by ranking the time series in increasing order, then extracting the rank for each year and finally taking the correlation between the ranks of the observations. We prefer Spearman-$\rho$ correlation to the classical Pearson correlation, as it is more robust and as it also better quantifies the ecological notion of "similar patterns" (which corresponds to similar ranking, rather than proportional values).

The matrix of pairwise synchrony indices $\rho_{ij}$ is represented in Figure 4.2.

**Synchronous species.** We define synchronous groups by clustering the species on the basis of their regional synchrony indices computed above. We apply an agglomerative hierarchical clustering algorithm, with dissimilarity matrix $d$ given by the pair-wise synchrony indices

$$d_{ij} = 1 - \rho_{ij} \tag{4.3}$$

and with Ward linkage [132]. We refer to the Appendix 4.B for a reminder on hierarchical clustering algorithms.

We use the `hclust` function from the R package `stats` [115] for the practical implementation.

## 4.3 Structuration of species synchronous groups in terms of long-term trends

This section is based on the technical report "What are the links between long-term temporal trends and regional-scale species synchrony in butterfly communities of Great Britain?", whose authors are

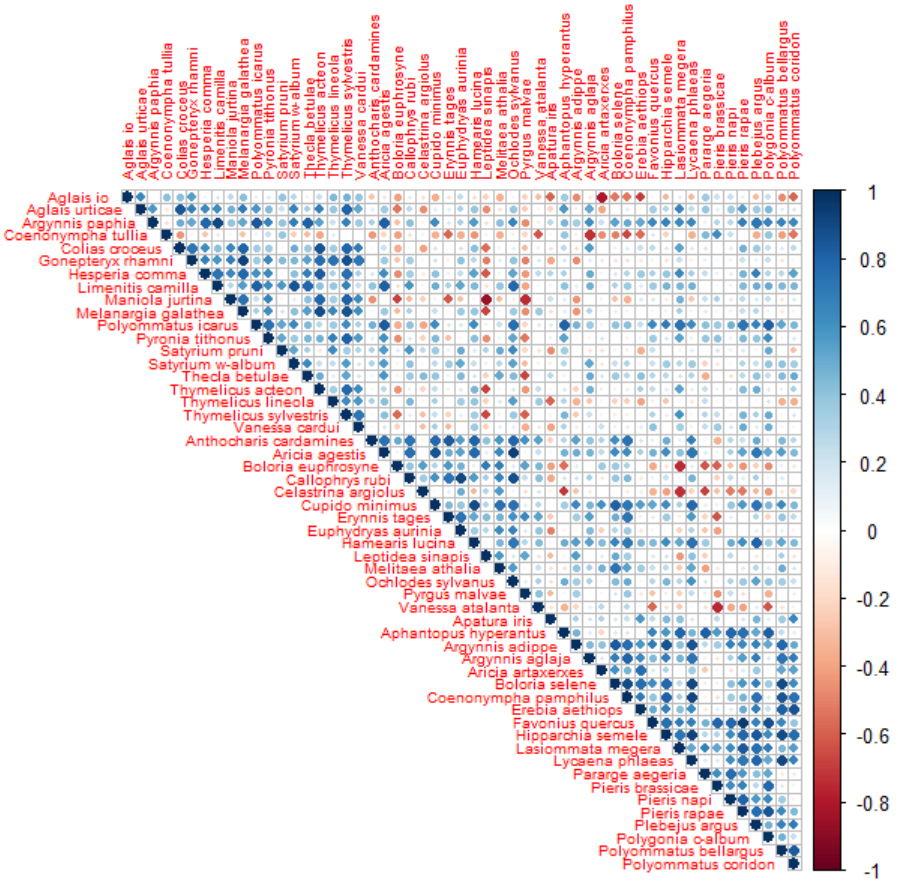



Figure 4.2 – Pairwise synchrony indices. A value of 1 (dark blue) corresponds to perfectly synchronous species, and −1 (dark red) to perfectly asynchronous species. Species are ordered according to their group of synchrony. Species from the same group of synchrony are grouped together in the matrix of correlation.

Théophile Olivier[1], Solène Thépaut[2], Elisa Thébault[3], Emmanuelle Porcher[1], Christophe Giraud[2], David Roy[4], Colin Fontaine[1].

**Overview.** The local scale is the most common in the study and comprehension of the stability of ecological systems. Recently, the stability at regional scale has been studied, especially in the context of global change, to understand the impact of regional extinction on the stability of ecosystems and their operation. This approach at a larger scale facilitates the decision making in management and conservation of the ecosystems.

At a regional scale, we find the same mechanisms that affect the stability as at a local scale. For example, the stability of the components of a system, and the synchrony between fluctuations of these components. Also, we observe temporal tendencies in the evolution of the abundances of the species. These tendencies, when negative, can lead to extinction of species and therefore to the

[1]Centre d'Ecologie et des Sciences de la Conservation, UMR 7204 MNHN-CNRS-Sorbonne Université, Muséum national d'Histoire naturelle de Paris, 43 rue Buffon, 75005 Paris

[2]Institut de Mathématiques d'Orsay, Université Paris-Sud, F-91405 Orsay Cedex

[3]Centre National de la Recherche Scientifique, Sorbonne Université, Institute of Ecology and Environmental Sciences of Paris, 4 Place Jussieu, 75005 Paris, France

[4]Centre for Ecology and Hydrology, Maclean Building, Crowmarsh Gifford, Wallingford OX10 8BB, United Kingdom

disappearance of the impact these species had on the ecosystem.

In this section, we study the link between the synchrony between species at a regional scale and the temporal tendencies of abundances of these species. We work on abundances of butterflies across the United Kingdom. We cluster the different species of butterflies according to their degree of synchrony. Then, we compare the groups of species to their temporal tendencies. The hypothesis of this study is that if the species whose abundances increase are in different groups than the species whose abundances decrease over time, we can expect that the extinction of a decreasing species leads to an extinction of all the other species in its group of synchrony. It would impact the potential of compensatory dynamics and the stability of communities at a regional scale. Studying the link between stability of populations at a regional scale and temporal tendencies of the species helps to understand the impact of global changes on communities at a regional scale.

We provide evidences of the existence of a variation of the value of synchrony between butterflies species at a regional scale. It suggests a potential compensatory dynamics that can stabilize the communities at a regional scale. After grouping the species into synchronous groups of species, we do not observe a specific temporal tendencies division, other than the ones created by random effects. These results suggest that the decline of species induced by global changes does not affect more than randomly expected the potential of compensatory dynamics of the butterfly communities in the United Kingdom at a regional scale.

### 4.3.1 Introduction

In a local community, synchrony reflects the similarity across species of local temporal fluctuations in a given property such as biomass or abundance, and is negatively correlated with stability [116]. A decrease in the compensatory dynamics increases synchrony among species, leading to an increase of the stability of the whole community [78], [79], [116]. Synchrony among species is one of the main drivers of the stability of local community [51], [79], [108], [116].

Recently, theoretical studies have extended our understanding of stability to larger spatial scales [130] , [129] , to better fit with the spatial scale of management and conservation policies. At the regional scale, the stability of metacommunities can be decomposed into two components [129]. First, the average metapopulation stability, that reflects the stability of aggregated local populations of a species. The contribution of each metapopulation stability to the average metapopulation stability is weighted by its relative abundance in the metacommunity. The stability of the metacommunity increases with the average metapopulation stability, and depends more on the abundant than rare metapopulations. Second, the regional-scale species synchrony. Synchrony reflects the correlation between regional-scale species fluctuations. Metapopulations with low synchrony (high asynchrony) will stabilize the metacommunity because of compensatory dynamics. Wang et al. [129] showed that at both local and regional scale, species synchrony decreased when species richness increased. They also showed that species synchrony decreased from local to regional scale, and suggested a potential effect of an increase of species richness from local to regional scale.

In local communities, variations in synchrony among species are related to species responses to variations of environmental conditions [119]. Variations of environmental conditions could also explain synchrony among species at a regional scale. Environmental changes such as landscape degradation or climate change affect environmental conditions but also temporal trends of species abundance, for example butterflies (Fox et al., 2015). Species that are synchronous because of similar responses to environmental condition variations could also have similar temporal trends because of similar sensitivity to environmental changes. Species groups according to regional species synchrony could be correlated with species group according to temporal trends. In the case of decline, temporal trends can lead to the extinction of a species. If declining species are in the same groups of synchronous species, the potential extinction of these groups would decrease the potential of compensatory dynamics at the regional scale, decreasing the regional metacommunity stability.

In this study, we aim to assess if long-term temporal trends of butterfly species computed on the 1976-2004 period are specifically distributed in groups of synchronous species, and if potential species

extinction could affect compensatory dynamics at the regional scale. We work on the UKBMS data set and compute synchrony indices as described in Section 4.2. We then analyze the distribution of long-term temporal trends across groups of synchrony.

### 4.3.2 Testing the link between long-term temporal trends and synchrony

**1976 – 2014 UK butterfly temporal trends.** To get long-term temporal trends, we use data from The State of the UK's Butterflies 2015 [36]. This report estimates the temporal trend of butterfly species in the UK, based on the UKBMS counts from 1976 to 2014. For each species, we extract the percentage of abundance change over the 1976-2014 period as well as the significance of estimated temporal trends, see Section 4.3.5. We only used significant trends to analyse the distribution of temporal trends among groups of synchrony.

**Testing for structuration.** To investigate if some groups of synchrony show particular global temporal trends, we perform some tests based on a resampling strategy. Our null hypothesis is that synchronous groups have no specific long-term trends, i.e. the long term trends are spread at random among the synchronous groups.

For each group, we compute the mean temporal trend for species with significant temporal trend. We then sample 1000 times in the full set of species the same number of species as in the group, and compute the mean temporal trend of species with significant trend. We compare the mean temporal trends of the group to the distributions of mean temporal trend of random samples. If the group mean temporal trend is outside the 95% confidence interval of the distribution, the mean temporal trend is declared "different from randomly expected". We emphasize that we are in an exploratory process, and we do not try to precisely identify which groups have significant long-term trends. Hence, though our setting corresponds to a multiple testing setting, we do not implement any FDR or FEWR control.

### 4.3.3 Results

We observe contrasting Spearman $\rho$ correlation values across species pairs, with both highly synchronous and highly asynchronous species pairs (Figure 4.2). We choose three different dissimilarity thresholds to separate species into groups of synchrony, resulting in three, five or seven groups (Figure 4.3). These three levels of clustering allow us to analyse the distribution of temporal trends inside groups of synchrony according to different level of synchrony among species.

Of the 53 species used in this study, eleven show a significant and positive temporal trend, twenty a significant and negative temporal trend, and twenty-two a non-significant temporal trend on the 1976-2004 period in the UK. There are both declining and increasing species in all groups of synchrony, except in group B.2, which contains only decreasing species and one stable species. In all groups but two, the number of species with significant declining temporal trend is greater than the number of species with significant increasing temporal trend (Figure 4.3). For group A.2, there are more species with significant increase, and for group C.3 there is an equal number of declining and increasing species. However, mean temporal trends inside groups are not different than randomly expected. Our resampling tests did not show any mean temporal trend outside the 95% confidence interval (Figure 4.4), thought the cluster B.2 is close to the boundary.

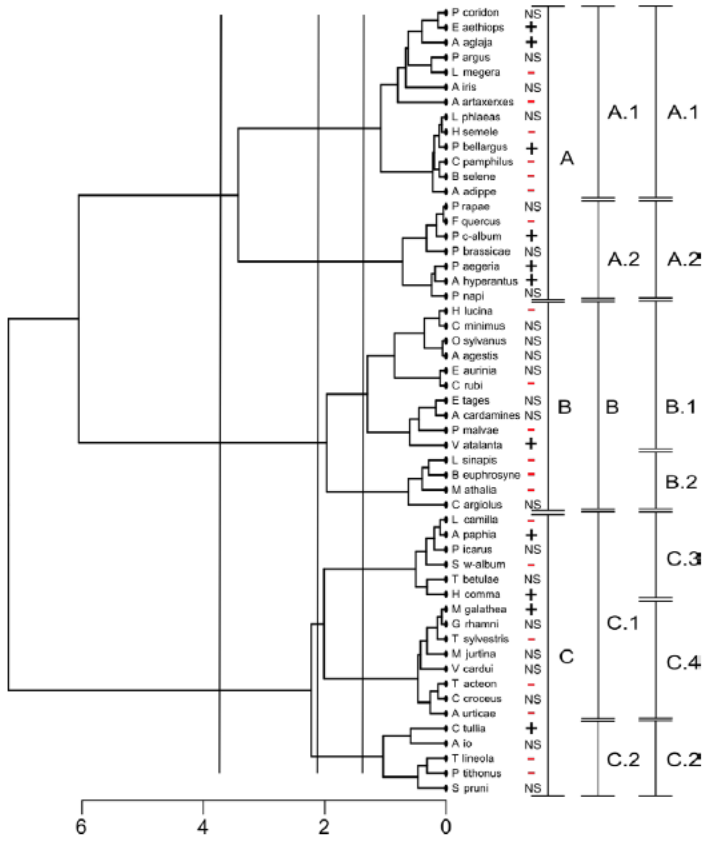



Figure 4.3 – Dendrogram of the hierarchical clustering performed for three levels of synchrony. Vertical black lines are the different thresholds to separate species into groups of synchrony.

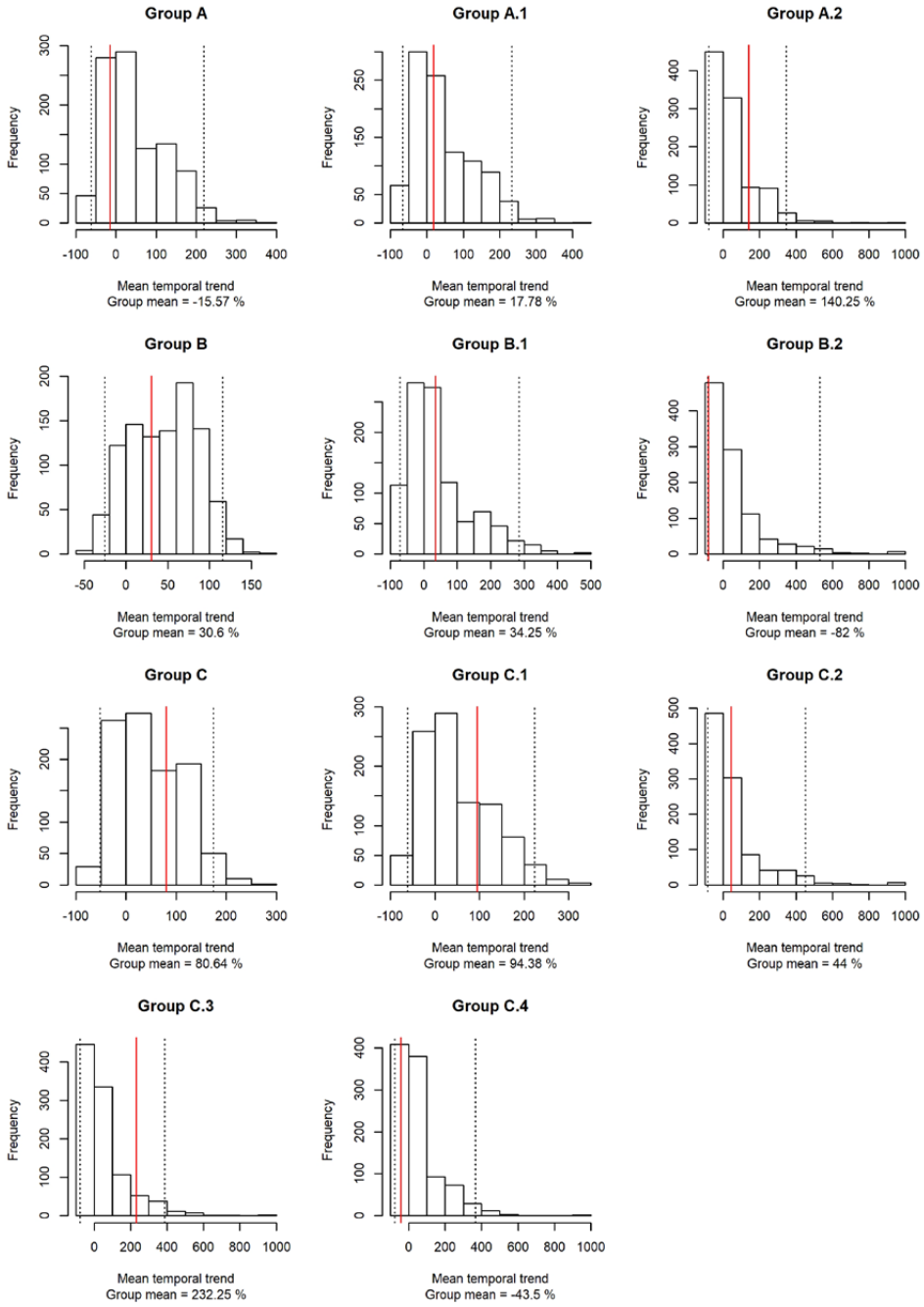



Figure 4.4 – Histograms of the null models by groups of synchrony, with only species with significant trends, for hierarchical clustering based on three, five and seven groups. Dashed black lines are the 95% confidence interval, and red lines are the mean temporal trend of groups.

### 4.3.4 Discussion

In this study, we highlighted variations in regional interspecific synchrony values across species pairs, suggesting that compensatory dynamics can occur to stabilize the abundance of regional butterfly communities. Variations in regional synchrony among species could be related to similarity in species traits as it has been found on local synchrony among species [95], [101]. The regional synchrony among species could also increase with the similarity in latitudinal range edges because of similar sensitivity to the variations of environmental conditions [117].

When analysing the distribution of significant temporal trends across groups of synchrony, we did not highlight groups with significant temporal trends lower or higher than randomly expected (Figure

4.3 and 4.4). The absence of particular distribution of temporal trends among groups of synchronous species suggests that different environmental factors drive temporal trends and synchrony. Temporal trends of butterfly abundance seems to be mainly affected by habitat loss and fragmentation [36]. At the local scale, synchrony among species increases primarily with species trait similarity and climate variability [95]. Species trait based approaches could allow to describe precisely which species traits explain temporal trends or regional interspecific synchrony and help conservation decisions.

Our results suggest that global changes responsible for temporal trends are not likely to affect regional compensatory dynamics more than if declining species were randomly distributed among groups of synchrony. However, temporal trends may affect average metapopulation stability. In addition to the correlation between temporal trends and regional-scale species synchrony, there is the need to investigate the potential correlation between temporal trends and average metapopulation stability.

### 4.3.5 Supplementary material: Species list and long-term temporal trends (Fox et al., 2015)

| Species | % Abundance change (1976-2014) | P-value |
|---|---|---|
| Aglais io | 17 | >0.05 |
| Aglais urticae | -73 | <0.05 |
| Anthocharis cardamines | 10 | >0.05 |
| Apatura iris | 69 | >0.05 |
| Aphantopus hyperantus | 381 | <0.001 |
| Argynnis adippe | -62 | <0.05 |
| Argynnis aglaja | 186 | <0.001 |
| Argynnis paphia | 141 | <0.001 |
| Aricia agestis | -25 | >0.05 |
| Aricia artaxerxes | -52 | <0.05 |
| Boloria euphrosyne | -71 | <0.001 |
| Boloria selene | -58 | <0.001 |
| Callophrys rubi | -41 | <0.01 |
| Celastrina argiolus | 37 | >0.05 |
| Coenonympha pamphilus | -54 | <0.001 |
| Coenonympha tullia | 261 | <0.01 |
| Colias croceus | 734 | >0.05 |
| Cupido minimus | 9 | >0.05 |
| Erebia aethiops | 170 | <0.01 |
| Erynnis tages | -19 | >0.05 |
| Euphydryas aurinia | -10 | >0.05 |
| Favonius quercus | -54 | <0.05 |
| Gonepteryx rhamni | 1 | >0.05 |
| Hamearis lucina | -42 | <0.01 |
| Hesperia comma | 943 | <0.001 |
| Hipparchia semele | -58 | <0.001 |
| Lasiommata megera | -87 | <0.001 |
| Leptidea sinapis | -88 | <0.001 |
| Limenitis camilla | -59 | <0.01 |
| Lycaena phlaeas | -37 | >0.05 |
| Maniola jurtina | 1 | >0.05 |
| Melanargia galathea | 50 | <0.05 |
| Melitaea athalia | -87 | <0.001 |
| Ochlodes sylvanus | -17 | >0.05 |
| Pararge aegeria | 84 | <0.01 |
| Pieris brassicae | -30 | >0.05 |
| Pieris napi | -7 | >0.05 |
| Pieris rapae | -25 | >0.05 |
| Plebejus argus | 19 | >0.05 |
| Polygonia c-album | 150 | <0.001 |
| Polyommatus bellargus | 175 | <0.01 |
| Polyommatus coridon | 20 | >0.05 |
| Polyommatus icarus | -17 | >0.05 |
| Pyrgus malvae | -37 | <0.05 |
| Pyronia tithonus | -41 | <0.05 |
| Satyrium pruni | -54 | >0.05 |
| Satyrium w-album | -96 | <0.001 |
| Thecla betulae | -15 | >0.05 |
| Thymelicus acteon | -76 | <0.01 |
| Thymelicus lineola | -88 | <0.001 |
| Thymelicus sylvestris | -75 | <0.001 |
| Vanessa atalanta | 257 | <0.01 |
| Vanessa cardui | 133 | >0.05 |

## 4.4 Structuration of species synchronous groups in terms of traits

In this section, we explore whether the synchronous groups are, or not, structured in terms of some individual traits. Our motivation is threefold. First, as synchrony is a major factor of stability, understanding the possible structuration of synchronous groups in terms of traits provides key insights on the functional stability of the ecosystem. Second, if we observe a strong structuration of some synchronous groups in terms of some traits, it suggests that these traits may be important factors for structuring these synchronous groups. This could help to better understand the mechanisms of interspecific synchrony. Finally, if species inside a group of synchrony share some common traits, it strengthens our confidence on the biological meaningfulness of the synchronous groups obtained in Section 4.2.

To explore this structuration, we proceed as follows. Our idea is to implement supervised machine learning technics in order to predict, from some traits data, the synchronous groups obtained in Section 4.2. The two algorithms that we implement are Random Forest [52] and Neural Networks [45]. For the latter, we use a two-layer Neural Networks, with the logistic activation function. We refer to the Annex 4.C for a reminder on these algorithms and the details of our implementation. The Random Forest algorithm has the nice feature to predict labels indifferently from categorical or numeric traits. The Neural Networks algorithm is acknowledged for its performance. Yet, the handling of categorial features is less straightforward. We compare the results and the performances of the two algorithms. Then, once we have run both algorithms, we use traits importance methods in order to determine which traits were the most important and useful to predict the groups. We use different methods to compute the importance of variables and then compare the results. These methods are described in Annex 4.C. The traits that are found important by several methods are more likely to be really significant in the determination of the groups.

The notebook gathering all the experiments and results is available online (`http://hebergement.u-psud.fr/solene.thepaut/IMG/html/notebookselectiontraits.html`). We summarize in this section and the Appendix 4.A our most important results, some other analyses and insights can be found in this document.

### 4.4.1 Traits data

We present in this subsection the different types of traits that we consider, and how we group them in different data sets based on their types and on which part of the life of the species they are related to.

To construct the traits dataset, we used the larval hostplant and key nectar plant species list compiled for UK butterfly species in the book of Dennis [30].

#### Initial dataset

Our initial dataset, denoted `Df all`, gathers a large variety of traits.

In `Df all`, most traits are divided into categorical sub traits. For such traits, we create a categorical trait that gathers all the information contained in the sub traits. Instead of considering each sub-trait as a trait, we consider the trait as a vector of dimension the number of its sub-trait. Then, to each vector is assigned a class. We create a categorical trait that contains the class of each vector for the species. If for one trait two species share the same sub-traits, then the created categorical trait will be equal to the same class for both species.

We consider the dataset containing the groups and the categorical traits we created from the original traits, such datasets are called factor datasets. The traits without sub-traits are not modified.

We also consider the non modified dataset where each sub-trait is considered as a trait in our algorithm.

Both datasets have their pros and cons. For the original datasets, we expect the predictors to be more precise since each information about a trait is as important as the others. In this case we are able to identify exactly which sub-trait of a trait is important in the determination of the groups of

synchrony. But, by keeping all the sub-traits, we have 52 species for about a hundred traits. This can lead to overfitting and make the interpretation of the results difficult.

On the other hand, the factor datasets give clear results, especially in the variable importance step. For each trait, we summarized the information into classes. The results can be less precise but they are less prone to overfitting.

From `dfAll`, we create sub-datasets containing the traits related to a same category. The idea to divide the traits into different datasets instead of using only the initial dataset containing all the traits is motivated by the fact that we only have 52 species for 35 traits in the factor datasets, and around a hundred traits in the original datasets. By dividing the traits into different datasets, we limit overfitting issues and we expect that our algorithms are more efficient and more precise in their selection of the most important traits. We present these datasets in the following. Since we have both categorical and numeric traits, we precise the type between parenthesis. The symbol (0/1) means that the trait, or its original sub-traits, is binary.

**Life history :** This dataset contains general information about the way of life of the species. It contains both categorical and numeric traits.

- Hibernation site (0/1)
- Overwintering stage (0/1)
- Ants collaboration (0/1)
- Voltinism (0/1)
- Max number of generations per year (continuous)
- Length of flight period (continuous)
- Adult Appearance (continuous)
- Mobility (continuous)

**Diet:** This dataset contains information about the diet of the species. All the following traits are continuous.

- Adult feeding specialism
- Adult nectar specialism
- Total number of hostplants
- Number of main hostplants
- Number of core hostplants
- Annual / perennial hosplants
- % nectar plants that are hostplants
- % of introduced nectar or hostplants

**Adult environment :** This dataset contains information about the environment in which the species live at an adult stage.

- Roost and rest sites (in adverse weather) (0/1)
- Mate location sites (0/1)
- Basking sites (0/1)

**Adult behavior :** This dataset contains information about the behavior of the species at an adult stage.

- Mate location method (0/1)
- Roosting mode (0/1)
- Basking method (0/1)
- Egg laying mode (0/1)

**Larval environment:** This dataset contains information about the environment of the species at a larval stage.

- Hostplant growth form occupied (0/1)
- Hostplant part used (0/1)
- Plant maturity (0/1)
- Hostplant patchiness (0/1)
- Larval zone occupied (0/1)

**Hostplants :** This dataset contains information about the hostplants of the species. That is to say the plants on which the species lay their eggs.

- Hostplant families used (0/1)
- Larval specialisation (categorical : monophageous / oligphageous / polyphageous)
- Type of hostplant used (categorical : annual, perennial, etc)

**Pupal environment :** This dataset contains information about the environment of the species at a pupal stage. In the pupal stage, a butterfly is not mobile. The only information we have about this stage is the location of the pupal.

- Pupal zone occupied (0/1)

**Egg environment :** This data set contains information about the location and environment where the species lay their eggs.

- Egg laying locality (0/1)
- Egg environment (0/1)
- Egg substrate (0/1)

### Final datasets

We test our supervised machine learning algorithms on all the datasets mentioned previously. Yet, it seems appropriate to create larger datasets containing traits from different categories. We present only the results and performance for the datasets described below. Some of the results for the other datasets can be found online (http://hebergement.u-psud.fr/solene.thepaut/IMG/html/notebookselectiontraits.html). We consider the aggregated datasets

- Adult traits (adultenvironment + adultbehavior)

- Larvae traits (larvalenvironment + larvalbehavior)
- Traits related to environment (adult environment + larval environment + egg environment + pupal environment)
- Traits related to behavior (adult behavior + larval behavior)
- All traits

We also create a dataset called combined dataset which contains the traits from the Life History, Hostplant and Diet datasets. After discussion with ecologists, the traits contained in the combined dataset are the most likely to explain the groups of synchrony.

### 4.4.2 Results

In this subsection we present our results on the different datasets. We describe the Leave-one-out error rates of our models in Tabular 4.5. We refer to the appendix 4.C for the details of the computations of the error rates

| Error rates for RF and NN algorithms | | | |
|---|---|---|---|
| Dataset | RF C $\tilde{\beta}$ | RF C $\tilde{\beta}$ Factor | NN C $\tilde{\beta}$ Factor. |
| All | 0.49 | 0.47 | 0.45 |
| Combined | 0.42 | 0.42 | 0.49 |
| Adult | 0.60 | 0.53 | 0.23 |
| Larval | 0.60 | 0.64 | 0.43 |
| Behavior | 0.53 | 0.47 | 0.28 |
| Environment | 0.66 | 0.58 | 0.57 |

Figure 4.5 – This tabular summarize the performance of the Random Forest (RF) and Neural Networks (NN) on the different datasets. The acronym C $\tilde{\beta}$ stands for corrected $\tilde{\beta}_y^{(i)}$. Factor means that the datasets used are the modified datasets where the sub-traits have been replaced by one categorical trait. If there is no Factor, then the original datasets with the sub-traits were used.

As a comparison, since we have 3 groups of synchrony, if we choose at random the group of each observations, we would have in average a 66% error rate. And the $p$-value of a 40% error rate is less than 5%. Considering the quality of our data, we consider that a 40% error rate is a significant improvement from the random classification.

In Tabular 4.5, we observe that the Neural Networks gives most of the time better results than the Random Forest. We emphasize yet that, as detailed in the Appendix 4.C the error rates are not computed exactly in the same way.

The difference between the original datasets and the factor datasets depends on which dataset we trained our models. When computing the new traits, we noticed that for some traits the number of class created was reasonable, around 10 classes, but could lead to 40 classes. The more classes, the less relevant is the trait, because it does not allow to differentiate the species if each species has it own class. This could explain the high error rate on these datasets.

We observe that the more promising datasets are the Adult, Hostplant and Behavior dataset.

The performance of both Random Forest and Neural Network are disappointing on the dataset containing all the traits and on the combined dataset. This is probably explained by the overfitting due to the high number of traits compared to the low number of observations.

The performances of the models are encouraging on some of the datasets. We now present our results about the importance of variables for the all dataset. We refer to Appendix 4.C for the details on the computation of the variable importance. We consider that the higher the accuracy of our model, the more credit it gives to the importance of variables we compute. Nonetheless, if some traits seems to be important for all algorithms and with several methods for the computation of

importance, it would increase the probability that these traits explain, at least in part, the groups of synchrony.

On the Figures 4.6 and 4.7, we colored the traits according to the phase of the butterfly life it relies to.

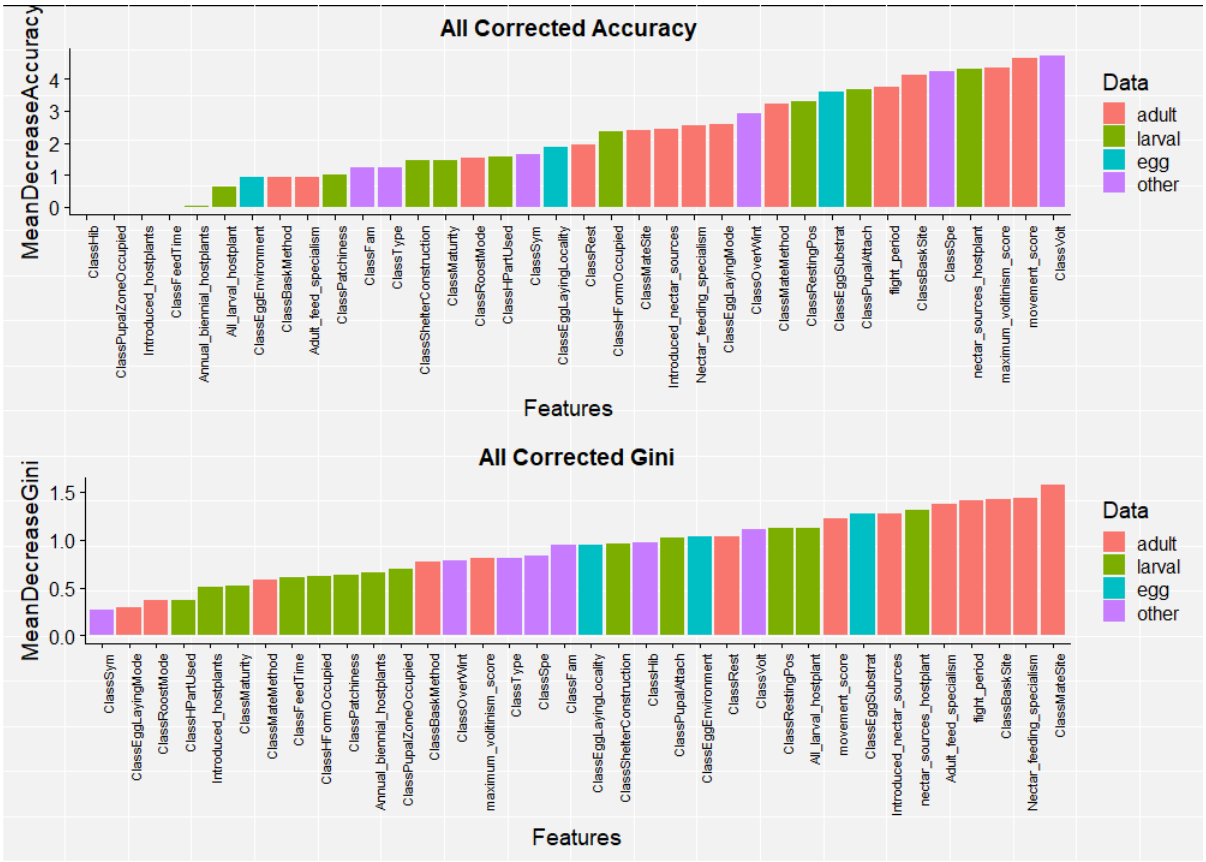



Figure 4.6 – This figure represents the Importance of variables found by the Random Forest algorithm for the all datasets with groups of synchrony found thanks to the corrected $\tilde{\beta}_y^{(i)}$. The method for importance of variables is discussed in 4.C.2. In this plot there is not a clear delimitations between the most important traits and the other. There are notable differences between the two plots. In both plots, the more important traits come from the adult dataset but not exclusively.

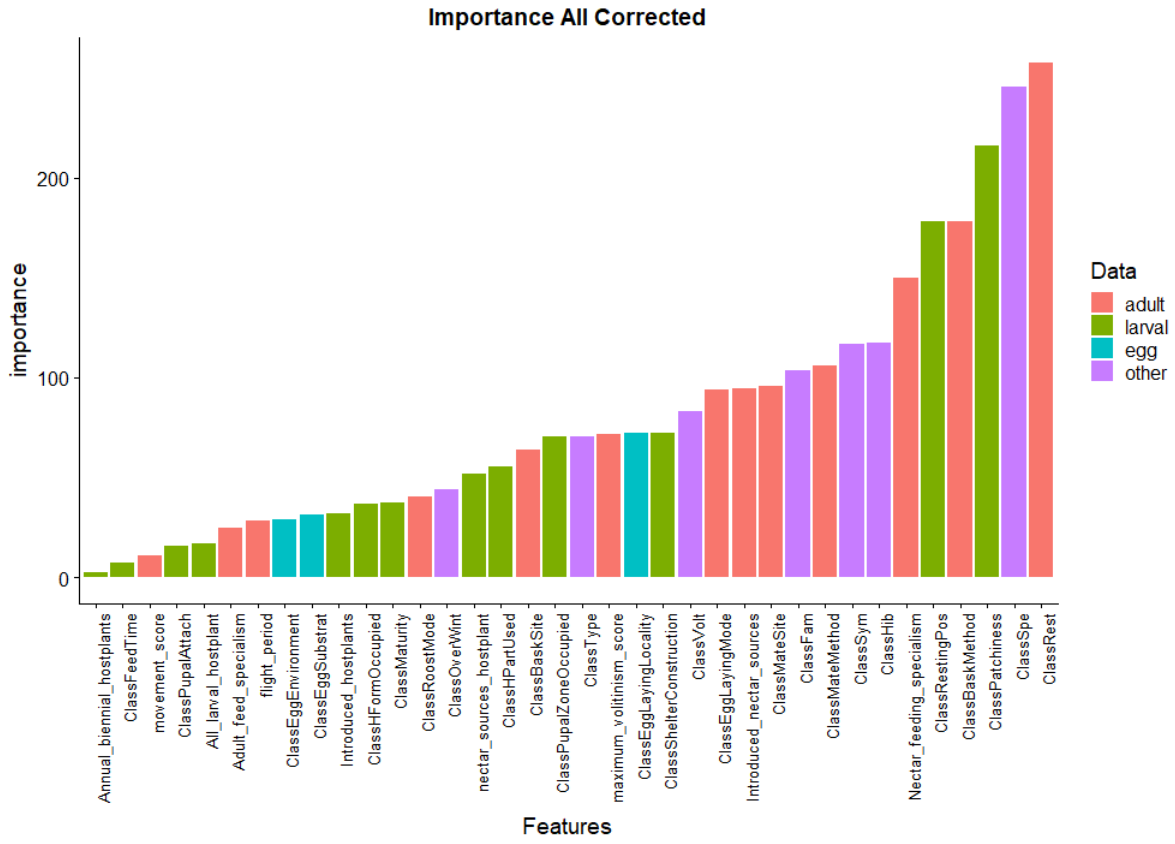



Figure 4.7 – This figure represents the Importance of variables found by the Neural Networks for the all datasets with groups of synchrony found thanks to the corrected $\tilde{\beta}_y^{(i)}$.The method for importance of variables is discussed in 4.C.2 We can see that the most important traits comes from the adult and larval datasets. As for the Random Forest, when the groups of synchrony are computed thanks to the corrected parameters, there is no real gap between the most important traits and the others.

The following traits are important both for the Random Forest algorithm and mean decrease accuracy or Gini methods and for the neural network algorithm and Olden method

- Class Diet Specialization
- Class Resting Position
- Nectar Feeding Specialism

When moving to the other datasets, we obtain varying results. These results are presented in the notebook (http://hebergement.u-psud.fr/solene.thepaut/IMG/html/notebookselectiontraits.html).

It is hard to draw conclusions from the plot of the importance of traits. The results do not seem conclusive. Even if the previous traits are more important in all context, we do not have strong evidences that these traits explain the groups of synchrony. The error rate is rather high for the all datasets for both methods.

### 4.4.3 Discussion

We first discuss the methods used in this section, then we comment on our results.

For Neural Network, we used a neural network with two small hidden layers in order to compute the predictors. We tested to add layers to the networks but the results were the same or less accurate than the simple networks. Once we decided to use only two hidden layers, we tried different number

of neurons on each layers in order to have the best possible results. For optimisation, it is convenient to have an activation function which is everywhere differentiable. Accordingly, we choose the logistic function as activation function. We observe that the accuracy of the Neural Network is way better than the accuracy of Random Forest in most cases.

Also, we use Dummy Variables to replace the categorical traits in our datasets. Since the Dummy Variables are computed thanks to the labels, there exists a correlation between them and the groups of synchrony. With our data, it is complicated to find a better way to use Neural Networks on our datasets.

The Random Forest is most of the time less efficient than the Neural Networks. But we can use all our traits without modification, which is a major advantages since our data consists in all kind of traits.

We did not compare all the datasets, but the error rate seems systematically lower when we use Neural Networks instead of Random Forest. In a way, the fact that the Neural Networks are more successfull than Random Forest to predict the groups of the species can imply that the importance of traits found with Neural Networks are more trustworthy than the one found with random Forest. However, what is really significant in our study is the fact that there exist traits that are found important both by random Forest and Neural Network and with different methods (Mean Accuracy and GIni methods for Random Forest and Olden method for Neural Network). These traits have more chances to really be decisive in the determination of the groups.

In the view of the quality of our data, and the accuracy of our models, we have to be careful about our conclusion on the important traits. The results have to be analyzed and their relevance to be confirmed by ecologists on the basis of their expert knowledge and the literature on the field.

Indeed, we have few observations compared to the number of traits. Moreover, we suspect that our data are quite noisy, since the abundances come from a participative science program and their analysis require to handle the nuisance induced by the phenologies. If the signal is weak compared to the noise, then it is difficult to find a pattern in the groups of synchrony we computed in Section 4.2.

Even if we do not end up with conclusive results, our methodology can be replicated to other settings to study the link between traits of species and interspecific synchrony. To successfully work on this subject, the collaboration between ecologists and statisticians is crucial.

## 4.A Non-corrected synchrony

In this appendix we gather some results on synchrony computed from non-corrected regional inter-annual variation $\beta_y^{(i)}$ instead of corrected regional inter-annual variation $\tilde{\beta}_y^{(i)}$ .

### 4.A.1 Synchrony without trend correction

A explained page 111, synchrony is meant to reflect similar non-systematic inter-annual variations in abundance between two species. When computing correlations directly from the regional inter-annual abundance variations $\beta_y^{(i)}$, strong long-term temporal trends can hide the synchrony between two species. To illustrate this point and the necessity to correct for the trends, we compare in Figure 4.8 the clustering obtained when computing the Spearman-$\rho$ correlations from the regional inter-annual abundance variations $\beta_y^{(i)}$ and from the trend-corrected regional inter-annual abundance variations $\tilde{\beta}_y^{(i)}$. We observe some similarities between the groups found in the two dendogramms, but we can also see that there are not the same and species from the same group on the left dendogram end up in different groups in the right dendogram.

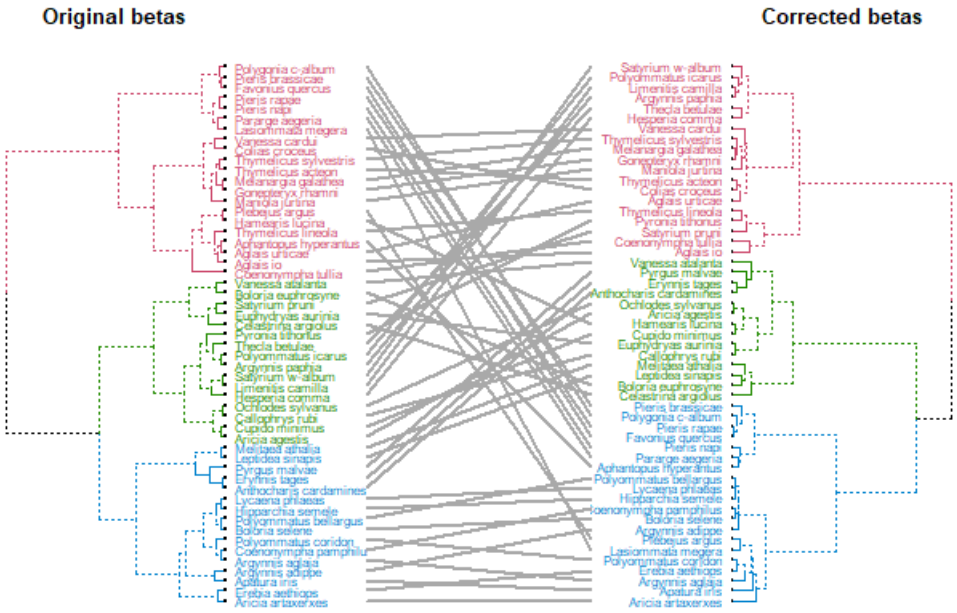



Figure 4.8 – Tanglegram of the dendograms obtained from the non corrected $\beta_y^{(i)}$ (left) and the trend-corrected $\tilde{\beta}_y^{(i)}$ (right).

### 4.A.2 Traits versus non-corrected synchronous groups

In this section, we report the results obtained on the synchrony versus trait structuration, when the synchrony indices are computed from the inter-annual abundance variations $\beta_y^{(i)}$, not corrected for the trends. These results allow us to identify different mechanisms between traits and synchrony, depending if we focus on non-systematic variations only (as in Section 4.4) or if we include long-term temporal trends as in the results presented below.

| Error rates for RF and NN algorithms | | | |
|---|---|---|---|
| Dataset | RF NC $\beta$ | RF NC $\beta$ Factor | NN NC $\beta$ Factor . |
| All | 0.42 | 0.43 | 0.45 |
| Combined | 0.43 | 0.40 | 0.36 |
| Adult | 0.55 | 0.64 | 0.30 |
| Larval | 0.51 | 0.58 | 0.23 |
| Behavior | 0.57 | 0.45 | 0.38 |
| Environment | 0.49 | 0.57 | 0.49 |

Figure 4.9 – This tabular summarize the performance of the Random Forest (RF) and Neural Networks (NN) on the different datasets. The acronym NC $\beta$ means that we focused on the groups of synchrony found with the non corrected $\beta_y^{(i)}$. Factor means that the datasets used are the modified datasets where the sub-traits have been replaced by one categorical trait. If there is no Factor, then the original datasets with the sub-traits were used.

On the Figures 4.10 and 4.11, we colored the traits according to the aspect of the butterfly life it relies to.

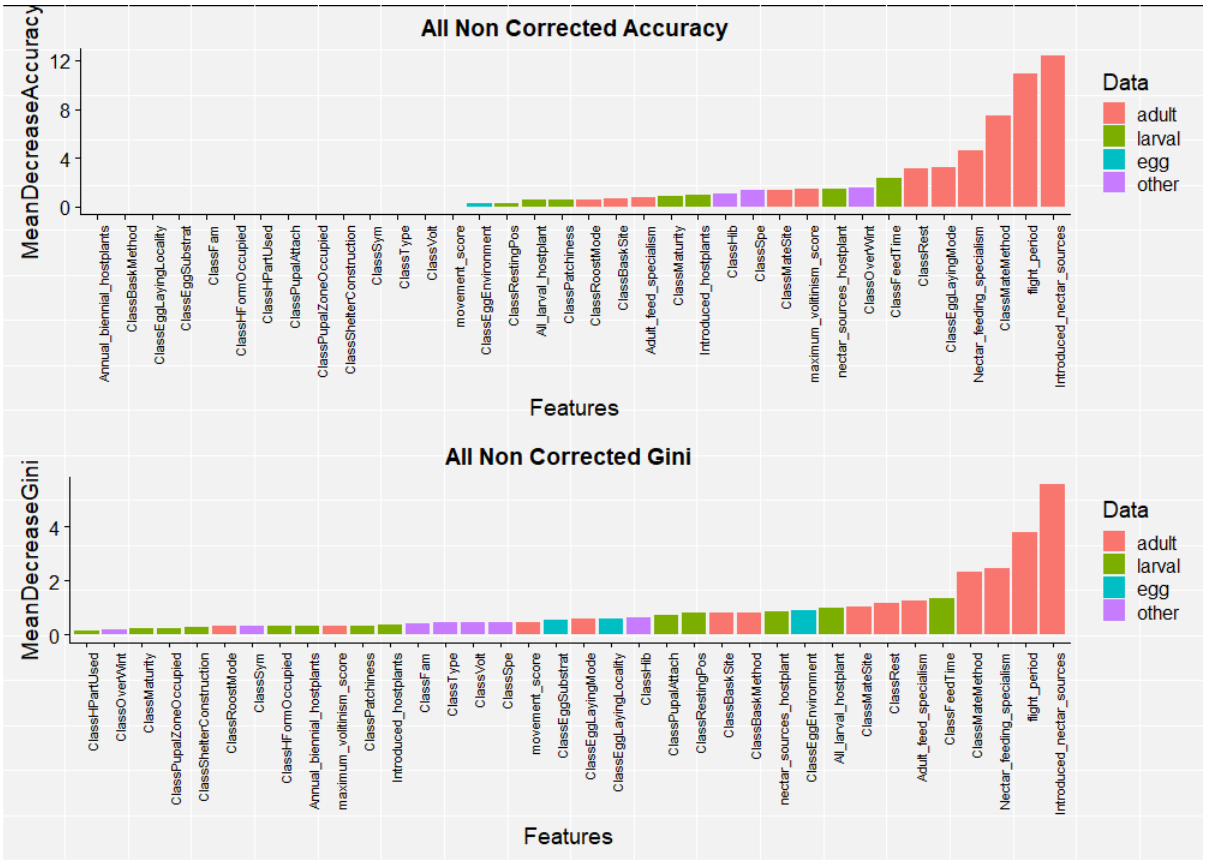



Figure 4.10 – This figure represents the Importance of variables found by the Random Forest algorithm for the all datasets with groups of synchrony found thanks to the non corrected $\beta_y^{(i)}$ 4.C.2. We can see that the most important traits comes from the adult dataset. According to this plot, there exists a gap between the most important traits and the others. Even if there are small differences between the two plots, the Decreasing Accuracy methods and the Gini methods gives similar results.

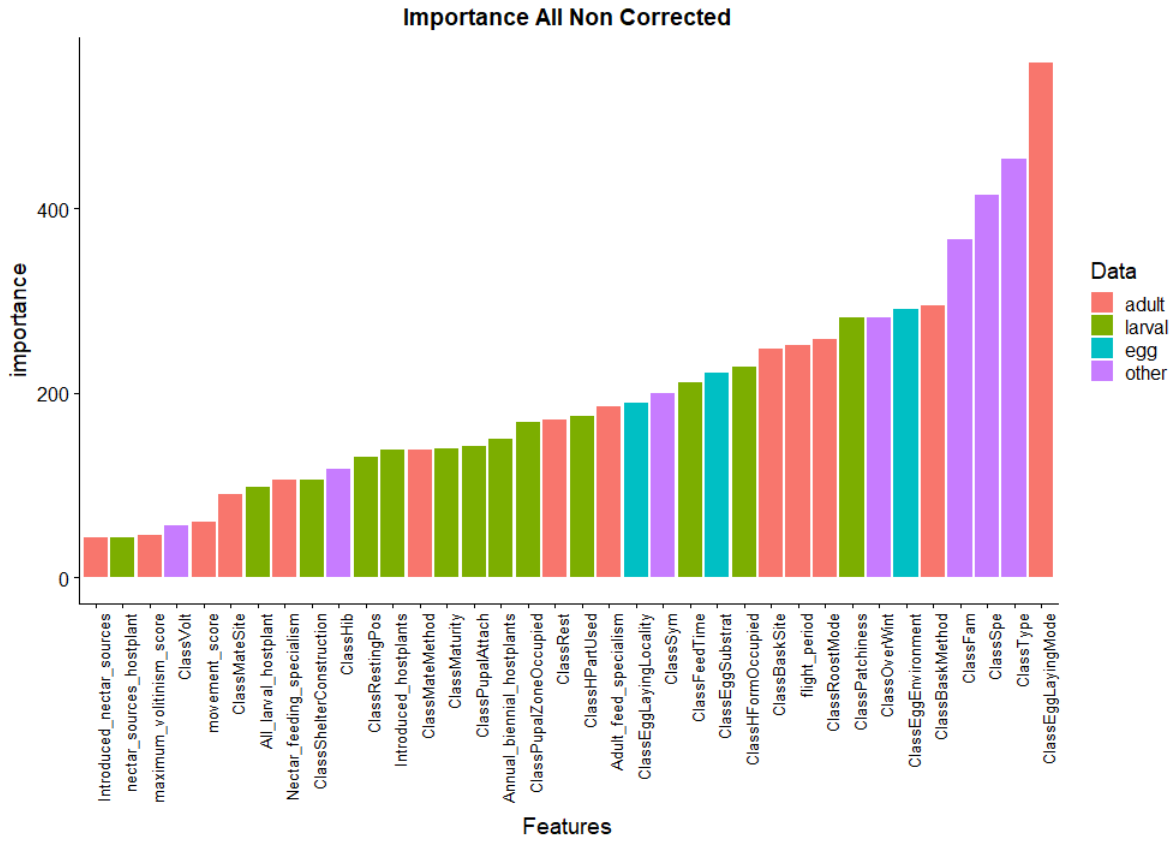



Figure 4.11 – This figure represents the Importance of variables found by the Neural Networks for the all datasets with groups of synchrony found thanks to the non corrected $\beta_y^{(i)}$ 4.C.2. We can see that the most important traits comes from the adult dataset and others, such as Hostplants. According to this plot, there is one trait more important than the other. Then, the importance decrease slowly from one trait from another.

The following traits are important both for the Random Forest algorithm and mean decrease accuracy or Gini methods and for the neural network algorithm and Olden method

- Class Egg laying Method
- Flight Period

The results for the importance of variables are not the same as the one obtained with the corrected $\tilde{\beta}$. We notice that the Class Specialization is important in both case for Neural Networks and its Olden method for computing the importance of variable but does not appear as important for the Random Forest algorithm and both Decreasing Accuracy and Decreasing Gini. Apart for this exception, it is hard to find a common important variable between the results for corrected $\tilde{\beta}$ and the non corrected $\beta$.

## 4.B Hierarchical clustering

Hierarchical clustering algorithms are clustering algorithms producing a sequence of nested clusterings. These algorithms take as input a matrix of pairwise dissimilarities $[d_{ij}]_{i,j}$ between the observations. In our case, $d_{ij}$ is defined by (4.3).

Like most of the popular clustering methods, hierarchical clustering is explained in details in [62].

**Dendogram.** The hierarchy of clusterings produced by a hierarchical clustering algorithm can be represented by a tree, called dendogram. The leafs of the tree correspond to the observations, and an internal node represents a cluster, which is made up of all the observations of its descendant leafs. Then, a clustering at a given level of the hierarchy is represented by the nodes at this level in the tree: the nodes of the tree at this level represent the clusters.

Usually, the length of an edge between a node and his two daughters is proportional to the intergroup dissimilarity between the two daughters clusters. This intergroup dissimilarity, that is used for clustering, is computed according to a linkage function (see Section 4.B.1) and the dissimilarity matrix.

**Agglomerative versus divisive clustering.** Hierarchical clustering algorithms fall into one of the two categories

- agglomerative clustering
- divisive (or partitional) clustering

*Agglomerative clustering.* Agglomerative clustering algorithms build the tree from the leaves and iteratively merge samples step by step and bottom-up. At step 0, we only have singletons. Then, step by step, observations and groups of observations are merged together according to the dissimilarity matrix and a specified criterion, the linkage criterion, to end-up with only one cluster containing all the observations. Typically, the higher the dissimilarity between two samples, the more steps will be needed before the two samples are merged into the same cluster. On the contrary, two very similar observations will be merged very quickly.

*Divisive clustering.* Divisive clustering works backward compared to agglomerative clustering. The starting point is a unique cluster with all the observations from the dataset, and the clusters are split iteratively into smaller one at each step, until having only singletons. This procedure is called top-down. As in agglomerative clustering, the splits are based on the dissimilarity matrix and a specified linkage criterion. If the dissimilarity between two observations is high, these two observations will be typically separated by the algorithm at a low step.

**Compactness and closeness of a cluster.** Two important traits drive the clustering process. The choice of a dissimilarity measure between observations (here we choose 1−synchrony index as dissimilarity) and the choice of a linkage function which specifies how clusters must be merged (or split). These two choices have a strong impact on the properties of the resulting clustering.

There is no universal criterion to decide whether a partition is better than another. Yet, the two following criteria are very popular in the applied literature on clustering.

1. Compactness : In a good cluster, the observations are similar to each other. Therefore, if a cluster is too "wide" in terms of dissimilarity, it is likely to find within it observations that are not as similar as we would wish them to be.

2. Closeness : The observations should be "closer" to the one in the same group than to the one in the other clusters, in term of similarity. If a cluster is too "narrow", it is likely to find observations from other clusters which are closer to an observation inside it than the other observations in the cluster are.

We discuss below some choices of linkage functions and their incidence on the two above properties of the clusters.

### 4.B.1 Linkage criteria

There are many possible choices of linkage function we list below four of the most popular ones.

**Complete linkage.** To compute the dissimilarity between a cluster $i$ and a cluster $i'$ using complete linkage, the algorithm compare the pairwise dissimilarities between the elements of cluster $i$ and the elements of cluster $i'$ and takes the largest of these dissimilarities. This linkage gives compact clusters where the observations inside a cluster are all very similar to each other.

**Single linkage.** The single linkage is the opposite of the complete linkage, as it takes the smallest of the pairwise dissimilarities as the dissimilarity between the two clusters. The clusters obtained from single linkage can lack of compactness but the observations in a cluster tend to be closer to their nearest neighbor in the cluster than to the observations of the other clusters, which is not always true with the complete linkage.

**Average linkage.** Average linkage is a compromise between single linkage and complete linkage. It takes the average of the pairwise dissimilarities between the element of cluster $i$ and the element of the cluster $i'$ as the dissimilarity between clusters $i$ and $i'$. It creates clusters both compact and close but respectively less than with complete and single linkage.

**Ward linkage.** For any cluster $c$, we denote $SS_c$ the summed square distance within $c$.

$$SS_c = \sum_{i \in c} \sum_{j \in c} d_{i,j}^2. \tag{4.4}$$

where $d_{i,j}$ is the measure of dissimilarities chosen for the clustering.
For any pair of cluster $c$ and $c'$, we denote $SS_{cc'}$ the summed square distance within the joint cluster created by $c$ and $c'$.

$$SS_{cc'} = \sum_{i \in c \cup c'} \sum_{j \in c \cup c'} d_{i,j}^2. \tag{4.5}$$

With the Ward linkage at each step, we compute for each pair of existing cluster $c$ and $c'$ the summed square distance within their joint cluster, minus the sum of their summed square distance : $W(c, c') = SS_{cc'} - (SS_c + SS_{c'})$. Then, we choose to merge the two clusters $c$ and $c'$ that minimizes $W(c, c')$.
This is the linkage we choose in our procedure.

### 4.B.2 Selecting the number of clusters

As explained above, once we have obtained the dendogram, we "cut" the tree at some height, and the nodes at this height gives the partition associated to this level. Finding a good level in the dendogram, which is equivalent to finding a good number of clusters, is a delicate task. When the number of clusters is not known, different selection procedures can be applied. Many methods have been proposed, we refer to [62] We discuss here the most popular ones. In practice, it is highly recommended to use more than one of them, in order to compare the choices made by each method.

When trying to select the number of clusters, the user will have to compare different partitions of the data with different number $K$ of clusters. To compare two different partitions, different criteria have been proposed, each of them trying to define how good a partition is.

A good clustering would be one where the observations in a cluster are very close, or similar, to each other, and far away from the observations of the other clusters. The criteria described below quantify this statement. We define two measures : the within-cluster dissimilarity and the between-cluster dissimilarity, based on the dissimilarity $d_{ij}$ defined in (4.3). The within-cluster dissimilarity of a cluster $C_k$ is defined as

$$WCD(C_k) = \sum_{i,j \in C_k} d_{ij}.$$

The total within-cluster dissimilarity of a partition $\mathcal{C} = \{C_1, \ldots, C_K\}$ corresponds to the sum of the within-cluster dissimilarities of its clusters

$$TWCD(\mathcal{C}) = \sum_{k=1}^{K} WCD(C_k).$$

Symmetrically, we can define the between cluster dissimilarity as

$$BCD(\mathcal{C}) = \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \notin C_k} d_{ij} = \|d\|_1 - TWCD(\mathcal{C}).$$

We notice that, in a hierarchy of clusterings, when the number $K$ of groups grows, the total within-cluster dissimilarity $TWCD$ decreases, while the between cluster dissimilarity $BCD$ increases.

**A simple criterion.** A possible way to choose the number of clusters in hierarchical clustering is to cut the dendogram at a height where the edges are the longest. As mentioned before, the length of the edges in the dendogram is related to the dissimilarities between the daughter clusters. Cutting the tree where there is a large gap between a node and his daughters, amounts to cut the tree at a level where the splitting is significant. Hence, a simple rule is to cut the tree at the lowest level (closest to the leafs) where an edge larger than a prescribe level is present.

Yet, this criterion is somewhat rough and informal. More elaborate methods have been proposed.

**Elbow method.** When increasing the number of clusters, the $TWCD$ decreases. In a situation where the data present a clear clustering structure with $K^*$ clusters, the decrease of $TWCD$ with $K$ is typically stronger for values of $K$ smaller than $K^*$ than for values of $K$ larger than $K^*$. The shape of the plot of the $TWCD$ with respect to $K$ then present an "elbow" around $K^*$. This phenomenon can be theoretically explained in some contexts, see [88].

The principle of the "elbow method" is then to plot the $TWCD$ with respect to $K$, and to seek for a drop of the decrease of the $TWCD$ with $K$ ("elbow"). The selected value is the one located at the drop. This heuristic can be theoretically justified in some contexts as in [88].

Many other methods for selecting the number $K$ of clusters are variants of the elbow method.

**Silhouette method.** The silhouette coefficient [105] is a measure of dissimilarity between a point of a cluster and points of the other clusters for a given partition of the data. The silhouette coefficient of an observation is close to 1 if the observation is far away from the other clusters, and close to 0 if the observation is on the frontier between two clusters. To be more specific, let us explain how silhouette coefficients $(s_i : i = 1, \ldots, n)$ are computed.

Let $a_i$ be the average dissimilarities between the observation $i$ and the other observations in its cluster; $d_{i,G_j}$ be the average dissimilarity between $i$ and the observations in the cluster $G_j$; and $b_i = \min_{j: i \notin G_j} d_{i,G_j}$. The silhouette coefficient for the observation $i$ is defined as

$$s_i = \frac{b_i - a_i}{a_i \vee b_i}.$$

The silhouette plot represents the average of the silhouette coefficients in function of the number $K$ of clusters. The selected $K$ is the one giving the higher value in the silhouette plot.

The silhouette method is reported to be useful when one is looking for compact and well separated clusters.

## 4.C Supervised learning methods

### 4.C.1 Supervised classification with Random Forest and Neural Networks

#### Random Forest

Random Forest is a machine learning algorithm based on decision trees.

The decision tree algorithm is a machine learning algorithm which creates a model under the form of a tree. The purpose is to compute a classifier able of predicting the class of a new observation. The construction of the decision tree is done thanks to a train dataset in which every observation is labeled by its true class. At each step, the algorithm constructs a node on which a test is made on one of the features of the dataset. To determine the feature that is used at a certain node, the algorithm tests several of them and choose the one that maximizes a specified criterion.

We find on the leaves of the decision tree, the final decision of the classifier. When we want to classify a new observation, we browse the decision tree from the root to the leaves, and at each nodes, we test the value of the features of the new observation. At the end, we obtain the estimated class of the new observation. A classical reference for decision trees is [17].

The Random Forest algorithm combines bagging with decision trees in order to improve the robustness of decision trees. It was proposed in 1995 by Ho [52] and introduced more formally by Breiman and Cutler [16] in 2001. The Random Forest algorithm divides randomly the dataset into sub-datasets, and constructs a decision tree model on each of this sub-dataset. Then, for the classification of a new observation, the class of a new observation is decided by a majority vote between all the classifiers obtained by the decision trees on each subsets.

We use the train function of the `caret` package in R [38]. This function allows us to choose the algorithm we want to use for the supervised learning. It also tests multiple parameters in order to fit the best possible model to the data. In our case, the decision trees of the random forest are constructed according to the CART algorithm. In order to choose the feature that is used at a certain node, the algorithm chooses the one that maximizes the Gini index of diversity.

The Gini index of diversity of a feature is a measure of the frequency of the mis-classification of a random element among the observations in the train set if the feature is chosen for this node. Let us assume that there exist $n$ classes in which the observations have to be classified, then the Gini index is equal to the sum over all the observations in the train set of the probability to be chosen times the probability to be mis-classified.

One of the parameter for the Random Forest is the number of features among which the algorithm can choose when constructing the nodes of the decision trees. Another one is the number of decision trees to be constructed. The train function automatically tries different values for these parameters and chooses the ones that give the lowest rate of errors.

The error rate can be computed very easily with the Random Forest algorithm. Indeed, the algorithm naturally induces a cross validation. We don't need a test dataset to evaluate the performance of the algorithm on our data, which is an advantage since we only have 52 observations. To compute the rate of error, the algorithm browses all the observations. For each one of them, it computes a prediction of its class using only the decision trees constructed with a subset that does not contains the observation. This procedure is close to Leave-One-Out cross validation.

The Random Forest algorithm seems to be a good choice for our data. First, this algorithm can deal with different type of features. Since we have datasets containing both categorical and numerical features, we can simply use our datasets as they are in the algorithm. As mentioned earlier, we do not need to split our datasets into a train set and a test set. This is an important factor because we do not have a lot of observations compared to the number of features.

#### Neural Networks

Artificial Neural Networks are a supervised machine learning method inspired by the structure of the human cerebral cortex. The original idea comes from a biology article from 1959 [74]. The

book [45] gives a good insight about artificial neural networks and deep learning.
Artificial Neural networks are organized into different layers. Each layers is made of nodes, referred to as neurons. The neurons of successive layers are connected to each other. A Neural Network always have an input layer where each node contains the value of a feature and an output layer. In the case of classification, the output layer consists in as many nodes as classes. There exist many different Artificial Neural Network. We present here only the feed forward Neural Networks (ANN) where the signal travels one way from an input layer to the output layer.

Between the first layer and the last layer, we find one or several layers called hidden layers. The purpose of these layers is to broadcast and transform the signal from features to classes. Once the signal crossed the network, the output is one value for each node of the output layer. The predicted class corresponds to the output-layer node with the highest value.

The connections between the neurons are weighted. Between each layer, we also have a bias parameter represented outside the network. The weights and the bias are the parameters of the network. These parameters have to be optimized in order to train the model and obtain a classifier. When the signal crosses the network, each neuron of a layer receives as an input the weighted sum of the outputs of the neurons from the previous layer that are connected to it plus a bias.

To each neuron is associated an activation function. The activation is applied to the input of the neuron and the result is its output. The signal is then send to the next layer. If a neuron, denoted by $a$ is connected to $n$ neurons of the previous layer and if the $i$-th neuron produces an output denoted by $x_i$ for $i$ between 1 to $n$, then the neuron $a$ will receive as an input

$$z_a = \sum_{i=1}^{n} w_{i,a} x_i + b_a$$

where the $w_{i,a}$ are the weight on the edge between $i$ and $a$, $x_i$ is the output of the $i$-th neuron from the previous layer that is connected to $a$ and $b_a$ is the bias associated to $a$.

The neuron $a$ will produce as an ouput the quantity denoted $x_a$.

$$x_a = f(z_a)$$

where $f$ is the activation function associated to the layer of $a$. There exist many popular activation functions. For example, the ReLU function, tanh, sigmoid or logistic.

$$\begin{aligned} \text{ReLU}(x) &= \max(0, x) \\ \tanh(x) &= \frac{\mathrm{e}^{2x} - 1}{\mathrm{e}\, 2x + 1} \\ \text{sigmoid}(x) &= \frac{\mathrm{e}\, x}{\mathrm{e}^{x} + 1} \\ \text{logistic}(x) &= \frac{1}{1 + \mathrm{e}^{-x}}. \end{aligned}$$

We rather use the logistic function which is differentiable on the contrary of the ReLU function. We noticed empirically that the algorithm computed faster a predictor with the logistic function than with the tanh or sigmoid function.

Once the Artificial Neural Network structure is decided and constructed, the algorithm needs to optimize the parameters in order to fit a model and to predict the class of new observations. As in most supervised learning algorithm, the parameters are optimized thanks to a training dataset where the observations are labeled. We choose a loss function $l(.,.)$. The algorithm then finds the parameters that minimize :

$$\sum_{i=1}^{N} l(y_i, \hat{y}_i), \tag{4.6}$$

where $N$ is the number of observations in the train dataset, $y_i$ is the true label of the $i$-th observation and $\hat{y}_i$ is the predicted label for the $i$-th observation. We use the cross entropy loss. We denote $y_i^g$

the variable that is equal to 1 if the species $i$ is in the group of synchrony $g$ and 0 otherwise, and $\hat{y}_i^g$ the variable that is equal to 1 if the species $i$ is estimated in the group $g$ by the neural network and 0 otherwise. The cross entropy $l(y_i, \hat{y}_i) = -\sum_{g=1}^{G} y_i^g \log(\hat{y}_i^g)$, where $G$ is the number of class, in our case the number of group of synchrony.

In order to minimize (4.6), different optimization methods exist. Among the most popular ones is the simple gradient descent. The gradients are computed thanks to the so-called backpropagation. This method is close to the Gauss Newton algorithm since it aims to minimize a multivariate function and perform until convergence several iteration of a minimization step which gives an estimated minimum closer to the true minimum at each step. The difference is in the execution, which is going backward. At step $i$, the algorithm uses the weight computed at step $i-1$ in the neural network. We then compute the error between the estimated output and the true output. This error is then propagated backward into the neural network, and the weights are corrected according to the error. For more details about the bakcpropagation see [104]. To perform the gradient descent with backpropagation the activation functions need to be differentiable. The backpropagation allows to obtain the gradient of all the neurons in only one pass in the network. Then a gradient descent is performed in order to find the optimal parameters of the networks.

The Artificial Neural Networks are reported to be very efficient in many context. With a powerful computer, the optimization of the multiple parameters are not an issue. This is the reason why we choose to use an Artificial Neural Network to predict the class thanks to the features of the species. Neural Network is not as general as a Random Forest yet, since it does not handle categorical features. We need to transform our categorical and logical features to numeric features. To do so we create Dummy Variables from the data. Moreover, the normalization of the features is crucial for a Neural Network. We need to transform categorical features into numeric features and then normalize all the features before we use the training set to optimize the parameters of our Neural Network.

The way we compute our Dummy Variable is the following. For each class $c$ of our categorical feature, we compute the number $n_1$ of observations such that the feature is equal to $c$. Then, for each group of synchrony $g$, we compute the number $n_2$ of observations such that the feature is equal to $c$ and the group of synchrony is $g$. Finally, we compute a numeric variable to replace the original categorical feature as $\frac{n_2+0.1}{n_1+0.2}$. For a certain feature, a class $c$ of the feature and a group of synchrony $g$, the quantity we compute with the ratio $\frac{n_2}{n_1}$ is the proportion of observations from the group $g$ among all the observations such that the feature is equal to $c$. The 0.1 and 0.2 constants that we add to the numerator and the denominator are useful to deal with the case where $n_1$ and therefore $n_2$ are equal to 0. We avoid dividing by 0 and the value for the concerned feature and the concerned class is equal to 0.5 which is considered as a neutral value. This, way, we transform the categorical features into numeric ones that can be handled by neural network algorithm. We change the information provided by the feature since all the observations such that the feature is equal to $c$ does not share the same value for the new feature. Yet, the new feature gives information about the distribution of each group in the different categories of the feature. The main drawback of this method is overfitting since the Dummy Variables are directly correlated to the group of synchrony of each species. This is not the only methods to create dummy variables but it is the one which gives the best results with our data. Method like one-hot-encoder have the main drawbacks to increase the dimension of the data by replacing one categorical feature by several binary features. Since our dataset already contains a high number of features compared to the number of observations, using methods that increase the dimension of our data is not a good idea.

With Neural Network, we need to find a way to assess the performance of our algorithm. Since the number of observations is small, we decide to adopt a Leave One Out cross validation. That is to say, for each observation $i$, we set the train set as all the observations except $i$ and the test set as $i$. It allows to train our Neural Network on a sufficiently large dataset, but it also implies that we need to run the algorithm as many times as there are observations in the full dataset. To compute the accuracy, we compute for how many observations our model finds the true label.

We use the `neuralnet` function from the `NeuralNet` package in R [37]. The procedure is the following : we run the neuralnet function on each of the dataset. The number of nodes of each layer will depend on the dataset used. At the end of the neural network, a softmax function is applied giving 3 values, one for each groups. The estimated group of a species is the one with the highest output value.

We also stock the error rate and the importance of features along with our neural network algorithm. Then we can compare the results with the one we got with the Random Forest algorithm, even if there are not computed in the same way.

### 4.C.2 Features importance

Once we ran our supervised machine learning algorithms on our different datasets, we need to define the features that are the most important in our models to predict the synchronous groups obtained in Section 4.2. Depending on the algorithm we use, Random Forest or Neural Network, we have access to different methods.

#### For Random Forest

We use the function `importance` from the `caret` package [38], which rank the variables by order of importance.

The importance function for Random Forest models gives two ways to compute importance for features. The first one is an accuracy based method. This method computes for each features by how much the accuracy drops if we remove this feature from the predictors. High values of the MeanDecreaseAccuracy for a feature means that the accuracy dropped at a significant rate when we remove the considered feature. Therefore, it means that this feature is important for the model.

The second method is a Gini based method. The Gini Impurity is an indicator of the probability to missclass a new observation. Then, at each node of a decision tree, in order to decide which feature will be used, the Random Forest algorithm computes for each feature the total Gini impurity on all the existing decision trees, less the Gini impurity left if we remove the node containing the said feature. The mean of all Gini decrease is then taken over all the decision trees of the Random Forest model. It leaves us with one single value for every feature present in a dataset. As in the previous method, the higher is the MeanDecreaseGini for a feature, the more important is the corresponding feature.

In the case of Random Forest, the importance of the features can be computed within the algorithm, while the decision trees are constructed. One of the output of the train function of the caret package is the importance of features. We do not have to use another function since the quantities of interest are directly computed by the train function.

#### For Neural Network

The method we use to compute the importance of features for Neural Networks is different than the two methods used with Random Forest. If the more important variables are the same with the two algorithms and the two methods, there is a good chance for this features to be decisive in the determination of the groups. The method we will use with our neural network algorithm is called Olden method [94]. The principle is the following :

The method is based on the weights of the constructed neural network. For each feature, which corresponds to a node of the input layer, and each possible output, Olden computes the sum, over all the hidden nodes from the first node (feature) to the last node (group), of the product between the input weight of a node and the output weight of this node. Therefore, for each feature, we have three values of importance. Also, this method keeps the sign of the weight, meaning we can have a negative values for importance. In our case, we are only interested in the amplitude of the Olden coefficient which gives an insight about the importance of a feature. Hence, we compute the

Olden importance of the neural network. We take the maximum of the absolute values of the Olden coefficients over the groups of synchrony. At the end, we obtain one value for each features, which allows us to compare the results with the Random Forest importance of variables.

To compute the Importance of variables, we use the `olden` function of the `NeuralNetTools` package in R [9].

# BIBLIOGRAPHY


[1] Meteo office website.

[2] Greg W Anderson, Alice Guionnet, and Ofer Zeitouni. *An introduction to random matrices*, volume 118. Cambridge university press, 2010.

[3] Elena Andreou and Eric Ghysels. Detecting multiple breaks in financial market volatility dynamics. *Journal of Applied Econometrics*, 17(5):579–600, 2002.

[4] Sylvain Arlot, Alain Celisse, and Zaıd Harchaoui. Kernel change-point detection. *arXiv preprint arXiv:1202.3878*, 6, 2012.

[5] Ivan E Auger and Charles E Lawrence. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of mathematical biology*, 51(1):39–54, 1989.

[6] Maria-Florina Balcan, Yi Li, David P Woodruff, and Hongyang Zhang. Testing matrix rank, optimally. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 727–746. Society for Industrial and Applied Mathematics, 2019.

[7] Elita Baldridge, David J Harris, Xiao Xiao, and Ethan P White. An extensive comparison of species-abundance distribution models. *PeerJ*, 4:e2823, 2016.

[8] Yannick Baraud, Christophe Giraud, Sylvie Huet, et al. Gaussian model selection with an unknown variance. *The Annals of Statistics*, 37(2):630–672, 2009.

[9] Marcus W. Beck. NeuralNetTools: Visualization and analysis tools for neural networks. *Journal of Statistical Software*, 85(11):1–20, 2018.

[10] Richard Bellman et al. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 1954.

[11] Jan Bengtsson. Which species? what kind of diversity? which ecosystem function? some problems in studies of relations between biodiversity and ecosystem function. *Applied soil ecology*, 10(3):191–199, 1998.

[12] Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

[13] Lucien Birgé and Pascal Massart. Gaussian model selection. *Journal of the European Mathematical Society*, 3(3):203–268, 2001.

[14] Anthony J Bishara and James B Hittner. Testing the significance of a correlation with nonnormal data: comparison of pearson, spearman, transformation, and resampling approaches. *Psychological methods*, 17(3):399, 2012.

[15] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities*. Oxford University Press, Oxford, 2013. A nonasymptotic theory of independence, With a foreword by Michel Ledoux.

[16] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[17] Leo Breiman, Jerome Friedman, Richard Olshen, and Charles Stone. Classification and regression trees. wadsworth int. *Group*, 37(15):237–251, 1984.

[18] T. Tony Cai and Mark G. Low. Testing composite hypotheses, Hermite polynomials and optimal estimation of a nonsmooth functional. *Ann. Statist.*, 39(2):1012–1041, 2011.

[19] Bradley J Cardinale, J Emmett Duffy, Andrew Gonzalez, David U Hooper, Charles Perrings, Patrick Venail, Anita Narwani, Georgina M Mace, David Tilman, David A Wardle, et al. Biodiversity loss and its impact on humanity. *Nature*, 486(7401):59, 2012.

[20] Mark Chandler, Linda See, Kyle Copas, Astrid MZ Bonde, Bernat Claramunt López, Finn Danielsen, Jan Kristoffer Legind, Siro Masinde, Abraham J Miller-Rushing, Greg Newman, et al. Contribution of citizen science towards international biodiversity monitoring. *Biological Conservation*, 213:280–294, 2017.

[21] Sourav Chatterjee et al. Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214, 2015.

[22] Alice Cleynen, Emilie Lebarbier, et al. Model selection for the segmentation of multiparameter exponential family distributions. *Electronic journal of statistics*, 11(1):800–842, 2017.

[23] Olivier Collier, Laëtitia Comminges, and Alexandre B Tsybakov. Minimax estimation of linear and quadratic functionals on sparsity classes. *arXiv preprint arXiv:1502.00665*, 2015.

[24] Joseph H Connell and Ralph O Slatyer. Mechanisms of succession in natural communities and their role in community stability and organization. *The American Naturalist*, 111(982):1119–1144, 1977.

[25] Romain Couillet and Merouane Debbah. *Random matrix methods for wireless communications*. Cambridge University Press, 2011.

[26] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[27] Srivatsava Daruru, Nena M Marin, Matt Walker, and Joydeep Ghosh. Pervasive parallelism in data mining: dataflow solution to co-clustering large and sparse netflix data. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1115–1124. ACM, 2009.

[28] K. R. Davidson and S. J. Szarek. Local operator theory, random matrices and Banach spaces. In *Handbook of the geometry of Banach spaces, Vol. I*, pages 317–366. North-Holland, Amsterdam, 2001.

[29] W James Davies. Multiple temperature effects on phenology and body size in wild butterflies predict a complex response to climate change. *Ecology*, 100(4):e02612, 2019.

[30] Roger LH Dennis. *A resource-based habitat view for conservation: butterflies in the British landscape*. John Wiley & Sons, 2012.

[31] David Donoho, Matan Gavish, et al. Minimax risk of matrix denoising by singular value thresholding. *The Annals of Statistics*, 42(6):2413–2440, 2014.

[32] David L Donoho and Michael Nussbaum. Minimax quadratic estimation of a quadratic functional. *Journal of Complexity*, 6(3):290–323, 1990.

[33] Miguel Equihua. Fuzzy clustering of ecological data. *The Journal of Ecology*, pages 519–534, 1990.

[34] Zhou Fan, Leying Guan, et al. Approximate $l_0$-penalized estimation of piecewise-constant signals on graphs. *The Annals of Statistics*, 46(6B):3217–3245, 2018.

[35] Marie-josée Fortin, Mark RT Dale, and Jay M Ver Hoef. Spatial analysis in ecology. *Encyclopedia of environmetrics*, 5, 2006.

[36] R Fox, TM Brereton, J Asher, TA August, MS Botham, NAD Bourn, KL Cruickshanks, CR Bulman, S Ellis, CA Harrower, et al. The state of the uk's butterflies 2015. 2015.

[37] Stefan Fritsch, Frauke Guenther, and Marvin N. Wright. *neuralnet: Training of Neural Networks*, 2019. R package version 1.44.2.

[38] Max Kuhn. Contributions from Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, the R Core Team, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, Luca Scrucca, Yuan Tang, Can Candan, and Tyler Hunt. *caret: Classification and Regression Training*, 2018. R package version 6.0-81.

[39] Piotr Fryzlewicz et al. Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6):2243–2281, 2014.

[40] Damien Garreau and Sylvain Arlot. Consistent change-point detection with kernels. *Electronic*

*Journal of Statistics*, 12(2):4440–4486, 2018.

[41] Matan Gavish and David L Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014.

[42] Christophe Giraud. *Introduction to high-dimensional statistics*. Chapman and Hall/CRC, 2014.

[43] Christophe Giraud, Romain Julliard, and Emmanuelle Porcher. Delimiting synchronous populations from monitoring data. *Environmental and ecological statistics*, 20(3):337–352, 2013.

[44] Gene Golub, Virginia Klema, and Gilbert W Stewart. Rank degeneracy and least squares problems. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1976.

[45] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[46] Volker Grimm and Christian Wissel. Babel, or the ecological stability discussions: an inventory and analysis of terminology and a guide for avoiding confusion. *Oecologia*, 109(3):323–334, 1997.

[47] Adityanand Guntuboyina, Donovan Lieu, Sabyasachi Chatterjee, and Bodhisattva Sen. Spatial adaptation in trend filtering. *arXiv preprint arXiv:1702.05113*, 8, 2017.

[48] Yanjun Han, Jiantao Jiao, Rajarshi Mukherjee, and Tsachy Weissman. On Estimation of $L\_\{r\}$-Norms in Gaussian White Noise Models. *arXiv preprint arXiv:1710.03863*, 2017.

[49] Yanjun Han, Jiantao Jiao, and Tsachy Weissman. Local moment matching: A unified methodology for symmetric functional estimation and distribution estimation under Wasserstein distance. *arXiv preprint arXiv:1802.08405*, 2018.

[50] Brage B Hansen, Vidar Grøtan, Ronny Aanes, Bernt-Erik Sæther, Audun Stien, Eva Fuglei, Rolf A Ims, Nigel G Yoccoz, and Åshild Ø Pedersen. Climate events synchronize the dynamics of a resident vertebrate community in the high arctic. *Science*, 339(6117):313–315, 2013.

[51] Andrew Hector, Yann Hautier, Philippe Saner, Lukas Wacker, Robert Bagchi, J Joshi, Michael Scherer-Lorenzen, Eva M Spehn, Ellen Bazeley-White, Markus Weilenmann, et al. General stabilizing effects of plant diversity on grassland productivity through population asynchrony and overyielding. *Ecology*, 91(8):2213–2220, 2010.

[52] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[53] Jan-Christian Hütter and Philippe Rigollet. Optimal rates for total variation denoising. In *Conference on Learning Theory*, pages 1115–1146, 2016.

[54] Sergei Izrailev. *tictoc: Functions for timing R scripts, as well as implementations of Stack and List structures.*, 2014. R package version 1.0.

[55] Brad Jackson, Jeffrey D Scargle, David Barnes, Sundararajan Arabhi, Alina Alt, Peter Gioumousis, Elyus Gwin, Paungkaew Sangtrakulcharoen, Linda Tan, and Tun Tao Tsai. An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters*, 12(2):105–108, 2005.

[56] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[57] Javier Jarillo, Bernt-Erik Sæther, Steinar Engen, and Francisco J Cao. Spatial scales of population synchrony of two competing species: effects of harvesting and strength of competition. *Oikos*, 127(10):1459–1470, 2018.

[58] Jiantao Jiao, Kartik Venkat, Yanjun Han, and Tsachy Weissman. Minimax estimation of functionals of discrete distributions. *IEEE Transactions on Information Theory*, 61(5):2835–2885, 2015.

[59] IM Johnstone. Gaussian estimation: sequence and multiresolution models. *Unpublished manuscript*, 2011.

[60] Lou Jost. Entropy and diversity. *Oikos*, 113(2):363–375, 2006.

[61] Mira Kattwinkel and Eduard Szöcs. *openSTARS: An Open Source Implementation of the 'ArcGIS' Toolbox 'STARS'*, 2018. R package version 1.1.0.

[62] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[63] Ashish Khetan and Sewoong Oh. Spectrum estimation from a few entries. *The Journal of Machine Learning Research*, 20(1):718–772, 2019.

[64] Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

[65] Eriko Koda. Scene-change-point detecting method and moving-picture editing/displaying method, February 15 2000. US Patent 6,025,886.

[66] Vladimir Koltchinskii and Karim Lounici. Concentration inequalities and moment bounds for sample covariance operators. *Bernoulli*, to appear.

[67] Weihao Kong, Gregory Valiant, et al. Spectrum estimation from samples. *The Annals of Statistics*, 45(5):2218–2247, 2017.

[68] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24, 2007.

[69] Serge Lang. *Graduate Texts in Mathematics: Algebra*. Springer, 2002.

[70] B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, 28(5):1302–1338, 2000.

[71] Émilie Lebarbier. Detecting multiple change-points in the mean of Gaussian process by model selection. *Signal processing*, 85(4):717–736, 2005.

[72] Oleg Lepski, Arkady Nemirovski, and Vladimir Spokoiny. On estimation of the L r norm of a regression function. *Probability theory and related fields*, 113(2):221–253, 1999.

[73] Gérard Letac and Hélène Massam. All invariant moments of the Wishart distribution. *Scandinavian Journal of Statistics*, 31(2):295–318, 2004.

[74] Jerome Y Lettvin, Humberto R Maturana, Warren S McCulloch, and Walter H Pitts. What the frog's eye tells the frog's brain. *Proceedings of the IRE*, 47(11):1940–1951, 1959.

[75] Yi Li, Huy L Nguyen, and David P Woodruff. On sketching matrix norms and the top singular vector. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1562–1581. Society for Industrial and Applied Mathematics, 2014.

[76] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

[77] Kevin Lin, James L Sharpnack, Alessandro Rinaldo, and Ryan J Tibshirani. A sharp error analysis for the fused lasso, with application to approximate changepoint screening. In *Advances in Neural Information Processing Systems*, pages 6884–6893, 2017.

[78] Michel Loreau and Claire de Mazancourt. Species synchrony and its drivers: neutral and nonneutral community dynamics in fluctuating environments. *The American Naturalist*, 172(2):E48–E66, 2008.

[79] Michel Loreau and Claire de Mazancourt. Biodiversity and ecosystem stability: a synthesis of underlying mechanisms. *Ecology letters*, 16:106–115, 2013.

[80] Thomas Lux and Michele Marchesi. Volatility clustering in financial markets: a microsimulation of interacting agents. *International journal of theoretical and applied finance*, 3(04):675–702, 2000.

[81] Robert MacArthur. Fluctuations of animal populations and a measure of community stability. *ecology*, 36(3):533–536, 1955.

[82] Anne E Magurran, Stephen R Baillie, Stephen T Buckland, Jan McP Dick, David A Elston, E Marian Scott, Rognvald I Smith, Paul J Somerfield, and Allan D Watt. Long-term datasets in biodiversity research and monitoring: assessing change in ecological communities through time. *Trends in ecology & evolution*, 25(10):574–582, 2010.

[83] Robert Maidstone, Toby Hocking, Guillem Rigaill, and Paul Fearnhead. On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, 27(2):519–533, 2017.

[84] Stéphanie Manel, Michael K Schwartz, Gordon Luikart, and Pierre Taberlet. Landscape ge-

netics: combining landscape ecology and population genetics. *Trends in ecology & evolution*, 18(4):189–197, 2003.

[85] Maurizio Maravalle, Bruno Simeone, and Rosella Naldini. Clustering on trees. *Computational Statistics & Data Analysis*, 24(2):217–234, 1997.

[86] Vladimir Alexandrovich Marchenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Matematicheskii Sbornik*, 114(4):507–536, 1967.

[87] Pascal Massart. Concentration inequalities and model selection. 2007.

[88] Cathy Maugis and Bertrand Michel. Data-driven penalty calibration: a case study for gaussian mixture model selection. *ESAIM: Probability and Statistics*, 15:320–339, 2011.

[89] Vladimir N Minin, Karin S Dorman, Fang Fang, and Marc A Suchard. Dual multiple change-point model leads to more accurate recombination detection. *Bioinformatics*, 21(13):3034–3042, 2005.

[90] Eduard Szöcs Mira Kattwinkel. openSTARS. https://github.com/MiKatt/openSTARS, 2016. [Online].

[91] Takafumi Miyatake, Satoshi Yoshizawa, and Hirotada Ueda. Method for detecting change points in motion picture images, January 28 1992. US Patent 5,083,860.

[92] Vito MR Muggeo and Giada Adelfio. Efficient change point detection for genomic sequences of continuous measurements. *Bioinformatics*, 27(2):161–166, 2010.

[93] Raj Rao Nadakuditi. Optshrink: An algorithm for improved low-rank signal matrix denoising by optimal, data-driven singular value shrinkage. *IEEE Transactions on Information Theory*, 60(5):3002–3018, 2014.

[94] Julian D Olden, Michael K Joy, and Russell G Death. An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecological Modelling*, 178(3-4):389–397, 2004.

[95] Théophile Olivier. *Le rôle de la diversité et des perturbations environnementales sur la stabilité temporelle des communautés animales en milieu naturel.* PhD thesis, Museum d'Histoire Naturelle, 2019.

[96] Robert T Paine. A note on trophic complexity and community stability. *The American Naturalist*, 103(929):91–93, 1969.

[97] Camille Parmesan. Influences of species, latitudes and methodologies on estimates of phenological response to global warming. *Global Change Biology*, 13(9):1860–1872, 2007.

[98] Egon Sharpe Pearson. *Karl Pearson: An appreciation of some aspects of his life and work.* CUP Archive, 1938.

[99] Dan Pelleg, Andrew W Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, volume 1, pages 727–734, 2000.

[100] Simon G Potts, Jacobus C Biesmeijer, Claire Kremen, Peter Neumann, Oliver Schweiger, and William E Kunin. Global pollinator declines: trends, impacts and drivers. *Trends in ecology & evolution*, 25(6):345–353, 2010.

[101] Sandy Raimondo, Marek Turcáni, Jan Patoèka, and Andrew M Liebhold. Interspecific synchrony among foliage-feeding forest lepidoptera species and the potential role of generalist predators as synchronizing agents. *Oikos*, 107(3):462–470, 2004.

[102] Alfréd Rényi et al. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961.

[103] 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(4):180–205, 2015.

[104] Raúl Rojas. *Neural networks: a systematic introduction.* Springer Science & Business Media, 2013.

[105] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[106] DB Roy and TH Sparks. Phenology of british butterflies and climate change. *Global change*

*biology*, 6(4):407–416, 2000.

[107] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European Signal Processing Conference*, pages 606–610. IEEE, 2007.

[108] Takehiro Sasaki, Xiaoming Lu, Mitsuru Hirota, and Yongfei Bai. Species asynchrony and response diversity determine multifunctional stability of natural grasslands. *Journal of Ecology*, 2019.

[109] Reto Schmucki, Guy Pe'Er, David B Roy, Constantí Stefanescu, Chris AM Van Swaay, Tom H Oliver, Mikko Kuussaari, Arco J Van Strien, Leslie Ries, Josef Settele, et al. A regionally informed abundance index for supporting integrative analyses across butterfly monitoring schemes. *Journal of Applied Ecology*, 53(2):501–510, 2016.

[110] Ernst-Detlef Schulze and Harold A Mooney. *Biodiversity and ecosystem function*. Springer Science & Business Media, 2012.

[111] Andrey A Shabalin and Andrew B Nobel. Reconstruction of a low-rank matrix in the presence of Gaussian noise. *Journal of Multivariate Analysis*, 118:67–76, 2013.

[112] TH Sparks and TJ Yates. The effect of spring temperature on the appearance dates of british butterflies 1883–1993. *Ecography*, 20(4):368–374, 1997.

[113] Sandra Stålhandske, Philipp Lehmann, Peter Pruisscher, and Olof Leimar. Effect of winter cold duration on spring phenology of the orange tip butterfly, anthocharis cardamines. *Ecology and evolution*, 5(23):5509–5520, 2015.

[114] Terence Tao. 254a, notes 3a: Eigenvalues and sums of hermitian matrices. *Terence Tao's blog*, 2010.

[115] R Core Team et al. R: A language and environment for statistical computing. 2013.

[116] Loic M Thibaut and Sean R Connolly. Understanding diversity–stability relationships: towards a unified model of portfolio effects. *Ecology Letters*, 16(2):140–150, 2013.

[117] JA Thomas, Dorian Moss, and E Pollard. Increased fluctuations of butterfly populations towards the northern edges of species' ranges. *Ecography*, 17(3):215–220, 1994.

[118] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[119] Andrew T Tredennick, Claire de Mazancourt, Michel Loreau, and Peter B Adler. Environmental responses, not species interactions, determine synchrony of dominant species in semiarid grasslands. *Ecology*, 98(4):971–981, 2017.

[120] Robert R Trippi and Efraim Turban. *Neural networks in finance and investing: Using artificial intelligence to improve real world performance*. McGraw-Hill, Inc., 1992.

[121] Charles Truong, Laurent Oudre, and Nicolas Vayatis. A review of change point detection methods. *arXiv preprint arXiv:1801.00718*, 2018.

[122] Alexandre B. Tsybakov. *Introduction to nonparametric estimation*. Springer Series in Statistics. Springer, New York, 2009. Revised and extended from the 2004 French original, Translated by Vladimir Zaiats.

[123] Pascal Vallet, Philippe Loubaton, and Xavier Mestre. Improved subspace estimation for multivariate observations of high dimension: the deterministic signals case. *IEEE Transactions on Information Theory*, 58(2):1043–1068, 2012.

[124] Lyudmila Yur'evna Vostrikova. Detecting "disorder" in multidimensional random processes. In *Doklady Akademii Nauk*, number 2, pages 270–274. Russian Academy of Sciences, 1981.

[125] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *Icml*, volume 1, pages 577–584, 2001.

[126] Gian-Reto Walther, Eric Post, Peter Convey, Annette Menzel, Camille Parmesan, Trevor JC Beebee, Jean-Marc Fromentin, Ove Hoegh-Guldberg, and Franz Bairlein. Ecological responses to recent climate change. *Nature*, 416(6879):389, 2002.

[127] Daren Wang, Yi Yu, and Alessandro Rinaldo. Optimal Covariance Change Point Localization in High Dimension. *arXiv preprint arXiv:1712.09912*, 2017.

[128] Daren Wang, Yi Yu, and Alessandro Rinaldo. Univariate mean change point detection: Pe-

nalization, cusum and optimality. *arXiv preprint arXiv:1810.09498*, 2018.

[129] Shaopeng Wang, Thomas Lamy, Lauren M Hallett, and Michel Loreau. Stability and synchrony across ecological hierarchies in heterogeneous metacommunities: linking theory to data. *Ecography*, 42(6):1200–1211, 2019.

[130] Shaopeng Wang and Michel Loreau. Ecosystem stability in space: $\alpha$, $\beta$ and $\gamma$ variability. *Ecology letters*, 17(8):891–901, 2014.

[131] Yu-Xiang Wang, James Sharpnack, Alexander J Smola, and Ryan J Tibshirani. Trend filtering on graphs. *The Journal of Machine Learning Research*, 17(1):3651–3691, 2016.

[132] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[133] Hermann Weyl. Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung). *Mathematische Annalen*, 71(4):441–479, 1912.

[134] Yihong Wu and Pengkun Yang. Minimax rates of entropy estimation on large alphabets via best polynomial approximation. *IEEE Transactions on Information Theory*, 62(6):3702–3720, 2016.

[135] Jindřich Žďánskỳ. Detection of acoustic change-points in audio streams and signal segmentation. *Radioengineering*, 2005.

[136] Y-T Zhou, Rama Chellappa, Aseem Vaid, and B Keith Jenkins. Image restoration using a neural network. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1141–1151, 1988.

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Titre : Problèmes de clustering liés à la synchronie en écologie : estimation de rang effectif et détection de ruptures sur les arbres

Mots Clefs : Synchronie, Apprentissage Machine, Regroupement, Classification non supervisée

Résumé : Au vu des changements globaux actuels engendrés en grande partie par l'être humain, il devient nécessaire de comprendre les moteurs de la stabilités des communautés d'êtres vivants. La synchronie des séries temporelles d'abondances fait partie des mécanismes les plus importants. Cette thèse propose trois angles différents permettant de répondre à différentes questions en lien avec la synchronie interspécifique ou spatiale. Les travaux présentés trouvent des applications en dehors du cadre écologique. Un premier chapitre est consacré à l'estimation du rang effectif de matrices à valeurs dans $\mathbb{R}$ ou $\mathbb{C}$. Nous apportons ainsi des outils permettant de mesurer le taux de synchronisation d'une matrice d'observations. Dans le deuxième chapitre, nous nous basons sur les travaux existants sur le problème de détection de ruptures sur les chaînes afin de proposer plusieurs algorithmes permettant d'adapter ce problème au cas des arbres. Les méthodes présentées peuvent être utilisées sur la plupart des données nécessitant d'être représentées sous la forme d'un arbre. Afin d'étudier les liens entre la synchronie interspécifique et les tendances à long termes ou les traits d'espèces de papillons, nous proposons dans le dernier chapitre d'adapter des méthodes de clustering et d'apprentissage supervisé comme les Random Forest ou les Réseaux de Neurones artificiels à des données écologiques.

Title : Clustering problems for synchrony in ecology : estimation of effective rank and change-points detection on trees.

Keys words : Clustering, Synchrony, Machine Learning, Unsupervised Classification

Abstract :In the view of actual global changes widely caused by human activities, it becomes urgent to understand the drivers of communities stability. Synchrony between time series of abundances is one of the most important mecanisms. This thesis offers three different angles in order to answer different questions linked to interspecific and spatial synchrony. The works presented find applications beyond the ecological frame. A first chapter is dedicated to the estimation of effective rank of matrices in $\mathbb{R}$ or $\mathbb{C}$. We offer tools allowing to measure the synchronisation rate of observations matrices. In the second chapter, we base on the existing work on change-points detection problem on chains in order to offer algorithms which detects change-points on trees. The methods can be used with most data that have to be represented as a tree. In order to study the link between interspecific synchrony and long term tendencies or traits of butterflies species, we offer in the last chapter adaptation of clustering and supervised machine learning methods, such as Random Forest or Artificial Neural Networks to ecological data.