THÈSE DE DOCTORAT

NNT : 2024UPASM043

université PARIS-SACLAY

Pattern detection and shape analysis for physiological time series

Détection et analyse de formes pour les séries temporelles physiologiques

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 574, mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques aux interfaces
Graduate School : Mathématiques
Référent : ENS Paris-Saclay

Thèse préparée dans l'unité de recherche **Centre Borelli**
(Université Paris-Saclay, CNRS, ENS Paris-Saclay)
sous la direction de **Laurent OUDRE**, Professeur des Universités
et le co-encadrement de **Charles TRUONG**, Chercheur

Thèse soutenue à Paris-Saclay, le 6 décembre 2024, par

Thibaut GERMAIN

Composition du Jury

Membres du jury avec voix délibérative

Alain TROUVE Professeur des Universités, ENS-Paris-Saclay	Président
Jessica LIN Professeure adjointe, George Mason University	Rapporteure
Romain TAVENARD Professeur des Universités, Université Rennes 2	Rapporteur
Laurent YOUNES Professeur, Johns Hopkins University	Examinateur
Guillermo SAPIRO Professeur, Princeton University	Examinateur

université PARIS-SACLAY | ÉCOLE DOCTORALE de mathématiques Hadamard (EDMH)

Titre : Détection et analyse de formes pour les séries temporelles physiologiques

Mots clés : apprentissage non supervisé, reconnaissance de formes, analyse de formes, séries temporelles

Résumé : Les données temporelles sont courantes en recherche biomédicale et les formes récurrentes ou anormales qu'elles comportent constituent des variables essentielles pour mener des analyses statistiques. Par exemple, dans les électrocardiogrammes, la forme des battements de cœur peut varier selon l'état physiologique des sujets, ce qui en fait une variable décisive pour diagnostiquer des maladies cardiaques. En revanche, la comparaison de telles formes nécessite des outils mathématiques spécifiques situés entre l'apprentissage automatique pour les séries temporelles et de l'analyse des formes.

Alors que la communauté d'analyse de formes s'est partiellement penchée sur les séries temporelles, les méthodes d'apprentissage en séries temporelles s'appuyant sur la notion de forme ont connu un succès dans diverses applications. Cette thèse vise à combiner les forces de ces deux domaines pour proposer des méthodes adaptées à l'analyse de données temporelles en recherche biomédicale. Une attention particulière sera accordée à l'interprétabilité des méthodes par le biais de rendus visuels des formes et des déformations, essentielles dans le dialogue entre données et chercheurs.

La thèse est structurée en deux parties : la première se concentre sur la recherche et la découverte de formes dans des séries temporelles, tandis que la seconde se concentre sur la comparaison de formes.

La première partie aborde le problème de la recherche ou de la découverte de formes dans de longues séries temporelles avec des distances indépendantes de certaines sources de variabilité modélisées par un groupe de déformations. Pour ce faire, une méthode générale de construction de distances invariantes par rapport au groupe de déformation est introduite. Ce cadre étend la distance Z-normalisée en permettant la personnalisation du groupe de déformations. Ces distances peuvent être intégrées dans des algorithmes pour la recherche ou la découverte de motifs sans perte d'efficacité. Un algorithme pour la découverte de motifs a aussi été développé. Il transforme une série temporelle en un graphique permettant l'identification de motifs récurrents. En tirant parti de l'interprétabilité et de l'efficacité de cet algorithme, une application interactive a été conçue pour faciliter la découverte de motifs.

La deuxième partie se concentre sur la comparaison des formes à l'aide de déformations élastiques qui tiennent compte des paramétrisations temporelles. Les méthodes proposées s'inspirent de l'analyse des cycles respiratoires de souris afin d'identifier des modalités de ventilation et d'évaluer les changements respiratoires chez des souris de génotypes différents après exposition à une molécule affectant la respiration. Une première méthode compare les cycles respiratoires à l'aide d'un algorithme de clustering s'appuyant sur la distance Dynamic Time Warping. Conçus comme référence, les résultats expérimentaux montrent que les clusters reflètent des modalités de ventilation liées à des génotypes et à des réponses à l'exposition. La seconde méthode crée des représentations vectorielles de séries temporelles échantillonnées de manière irrégulière et de longueur variable par le vecteur paramétrant les déformations qui font correspondre une série temporelle de référence aux autres. Cette approche s'appuie sur une méthode d'analyse de formes nommée Large Deformation Diffeomorphic Metric Mapping (LDDMM). Celle-ci est modifiée pour maintenir la structure spatio-temporelle des données tout en garantissant la bijectivité des représentations. Cette nouvelle méthode fournit des informations statistiques et visuelles sur les formes et les déformations. Une simple analyse statistique révèle que les déformations les plus importantes ont une signification physiologique permettant de mieux comprendre les modalités de ventilation selon le génotype et l'exposition.

Title : Pattern detection and shape analysis for physiological time series

Keywords : unsupervised learning, pattern detection, shape analysis, time series

Abstract : Time series are prevalent in biomedical applications where they frequently display recurring or abnormal patterns that hold significant information for statistical analysis. A notable example is the heartbeat in electrocardiograms, a recurring pattern whose shape can vary depending on the underlying condition, making it an important feature for diagnosing heart-related diseases. However, comparing such patterns requires specialized mathematical tools lying at the intersection between machine learning for time series and shape analysis.

While the shape analysis community has partially addressed the case of time series, shape-related approaches from machine learning for time series have achieved great success in various applications. This thesis aims to combine the strengths of both fields to propose methods suitable for biomedical research depending on temporal data. Particular attention will be given to methods' interpretability through visual interpretation of patterns and deformations, as it is key for meaningful interaction between the data and biomedical researchers.

The thesis is structured into two parts: the first focuses on searching for and discovering valuable patterns in time series, while the second concentrates on pattern comparison.

The first part tackles the challenge of searching or discovering patterns in long time series with distances independent of some irrelevant sources of variability modeled with a group of deformations. To that end, a general framework for constructing deformation-invariant distances is introduced. This framework extends the well-known Z-normalized Euclidean distance, invariant to amplitude scaling and offset shifts, by allowing customization of the group of deformations. The custom distances can be integrated into state-of-the-art algorithms for similarity search and motif discovery without efficiency loss. Additionally, an interpretable and interactive algorithm for motif discovery has been developed. This algorithm maps a time series onto a graph which is then summarized into a diagram providing a visual interpretation that facilitates the identification of recurring patterns. Furthermore, an interactive application has been designed for biomedical researchers, leveraging the algorithm's interpretability and efficiency for effective motif discovery.

The second part focuses on comparing temporal patterns using elastic deformations that notably account for time warping. The proposed methods are driven by the analysis of mice respiratory cycles recorded via plethysmography to identify ventilation modalities and assess the respiratory changes in mice with different genotypes after exposure to a drug affecting respiration. The first method compares respiratory cycles with a clustering algorithm based on the Dynamic Time Warping distance. Designed as a baseline, experimental results show that clusters have physiological relevance, reflecting genotype-specific ventilation modalities and responses to drug exposure. The second method creates fixed-size vector representations of irregularly sampled and variable-length time series by the vector parametrizing the deformations that map a reference time series to the observed ones. This approach draws on the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework from shape analysis, which is refined to maintain the spatiotemporal structure of the deformed time series while ensuring the bijectivity of the embedding. This method provides both statistical insights and visual interpretations of shapes and deformations. A simple statistical analysis reveals that the deformations responsible for most variability carry physiological significance, offering insights into ventilation modalities with respect to genotype and drug exposure effects.

Remerciements (en français)

Je n'aurais pu mener cette thèse sans le soutien, les échanges et les collaborations qui l'ont enrichie. Je tiens à exprimer ma gratitude envers toutes les personnes qui, de près ou de loin, ont contribué à sa réalisation et son aboutissement.

Je souhaite tout d'abord exprimer ma profonde reconnaissance à mes encadrants de thèse, Charles Truong et Laurent Oudre. Au-delà de mon parcours atypique, ils m'ont accordé toute leur confiance pour entreprendre ce doctorat. Je mesure pleinement leur engagement et leur investissement tout au long de ces trois années, j'en suis profondément reconnaissant. A leurs côtés, j'ai découvert les rouages de la recherche scientifique et acquis des compétences qui me guideront tout au long de ma carrière. Qui aurait cru que l'écriture, ma "bête noire", deviendrait un vrai plaisir.

Mes travaux, principalement ancrés dans des problématiques biomédicales, n'auraient pas eu la même portée sans la collaboration précieuse d'Éric Krejci. Bien que nos univers scientifiques soient très différents, il m'a toujours partagé avec enthousiasme et dans la plus grande patience ses connaissances en neurophysiologie. Cette complémentarité nourrie par nos échanges, a été une source de richesse et un moteur essentiel dans nos travaux. Je suis reconnaissant et fier du travail que nous avons accompli ensemble.

Je souhaite également remercier les membres de mon jury de thèse: le président Alain Trouvé, les rapporteurs Jessica Lin et Romain Tavenard ainsi que les examinateurs Guillermo Sapiro et Laurent Younes. Leurs travaux ont été pour moi une source d'inspiration majeure, et ce fut à la fois un plaisir et un honneur de leur présenter les résultats de ma recherche.

Je suis aussi reconnaissant envers le Centre Borelli, mon laboratoire d'accueil, pour la qualité du cadre de travail qu'il m'a offert tout au long de mon doctorat. Je souhaite remercier son directeur, Nicolas Vayatis, ainsi qu'à l'ensemble du secrétariat — Véronique Almadovar, Alina Muller, Annabelle Azan, Annabelle Bruneau, Gwladys Stouvenel et Pascal Renouf — pour toutes les dispositions prises à l'égard des doctorants et de leur quotidien.

Je souhaite tout particulièrement remercier Samuel Gruffaz, non seulement collègue doctorant, mais surtout un ami précieux, sans qui toute cette aventure n'aurait pu avoir lieu. De nos tempéraments pourtant bien différents est né une complicité qui nous a valu de belles réussites et qui s'étend bien au-delà du cadre professionnel. J'espère que nous pourrons continuer à enrichir notre amitié et nos collaborations.

J'adresse une mention spéciale à Sylvain Combette et Sam Perochon pour tous leurs "tips and tricks" qui m'ont aidé bien au-delà de la thèse, à Alexandre Bois, de quelques jours mon aîné et maître incontestable de la TDA, et à Donovan Morel, le psy de la bande, pour qui la meilleure des thérapies est un plat de frites le vendredi. Je tiens également à remercier Paul Boniol, dont l'arrivée au Centre Borelli a marqué un tournant structurant

dans ma thèse, et avec qui collaborer fut un véritable plaisir. De même, je souhaite remercier Chrysoula Kosma, partenaire de café mais surtout experte en deep learning, dont le dévouement et l'expertise ont été un atout précieux dans nos travaux communs.

Je tiens aussi à remercier tous les autres membres de mon équipe, l'admirable équipe Signal: Antoine Mazarguil, Bastien Lhospitalier, Lucas Haubert, Lucas Zoroddu, Marion Chauveau, Mona Michaud, Nicolas Cecchi, Quentin Laborde, Sylvain Young et Valerio Guerrini.

Mes remerciements s'étendent à tous mes collègues du Centre Borelli pour leur convivialité, la richesse de nos échanges et le plaisir de les côtoyer. Pour n'en citer que quelques-uns: Agnès Desolneux, Antoine de Mathelin, Argyris Kalogeratos, Axel Roques, Brian Tervil, Chloé Berland, Christophe Labourdette, Gaspard Abel, Harry Sevi, Ioannis Bargiotas, Josua Sassen, Julien Ballbé, Marie Garin, Margaux Calice, et Miguel Colom.

Au cours de ces trois années, il m'a aussi été donné la chance de pouvoir enseigner à l'ENS Paris-Saclay, je remercie Frédéric Pascal, Laure Quivy et Alain Trouvé pour cette belle opportunité qui m'a conforté dans la voie professionnelle sur laquelle je m'engage aujourd'hui. Pouvoir participer à la construction du cours de statistique et apprentissage fut un réel plaisir.

Je ne peux oublier mes parents qui ont toujours été présents et impliqués tout au long de mon parcours académique. Je suis profondément reconnaissant des valeurs qu'ils m'ont transmises et de leur soutien qui m'a permis d'avancer dans mes études et dans la vie.

Je ne saurais conclure sans exprimer toute ma gratitude envers ma chère et tendre Nadège. Présente à chaque étape de cette aventure, elle m'a accompagné tout au long de cette montagne russe qu'est une thèse sans jamais cesser de me soutenir et de m'encourager à suivre ma passion malgré toutes ces implications. Je suis profondément reconnaissant de l'avoir à mes côtés, et je me réjouis des prochaines aventures que la vie nous réserve.

En vous remerciant tous.

Résumé (en français)

Les données temporelles sont courantes en recherche biomédicale et les formes récurrentes ou anormales qu'elles comportent constituent des variables essentielles pour mener des analyses statistiques. Par exemple, dans les électrocardiogrammes, la forme des battements de cœur peut varier selon l'état physiologique des sujets, ce qui en fait une variable décisive pour diagnostiquer des maladies cardiaques. En revanche, la comparaison de telles formes nécessite des outils mathématiques spécifiques situés entre l'apprentissage automatique pour les séries temporelles et de l'analyse des formes.

Alors que la communauté d'analyse de formes s'est partiellement penchée sur les séries temporelles, les méthodes d'apprentissage en séries temporelles s'appuyant sur la notion de forme ont connu un succès dans diverses applications. Cette thèse vise à combiner les forces de ces deux domaines pour proposer des méthodes adaptées à l'analyse de données temporelles en recherche biomédicale. Une attention particulière sera accordée à l'interprétabilité des méthodes par le biais de rendus visuels des formes et des déformations, essentielles dans le dialogue entre données et chercheurs.

La thèse est structurée en deux parties : la première se concentre sur la recherche et la découverte de formes dans des séries temporelles, tandis que la seconde se concentre sur la comparaison de formes.

La première partie aborde le problème de la recherche ou de la découverte de formes dans de longues séries temporelles avec des distances indépendantes de certaines sources de variabilité modélisées par un groupe de déformations. Pour ce faire, une méthode générale de construction de distances invariantes par rapport au groupe de déformation est introduite. Ce cadre étend la distance Z-normalisée en permettant la personnalisation du groupe de déformations. Ces distances peuvent être intégrées dans des algorithmes pour la recherche ou la découverte de motifs sans perte d'efficacité. Un algorithme pour la découverte de motifs a aussi été développé. Il transforme une série temporelle en un graphique permettant l'identification de motifs récurrents. En tirant parti de l'interprétabilité et de l'efficacité de cet algorithme, une application interactive a été conçue pour faciliter la découverte de motifs.

La deuxième partie se concentre sur la comparaison des formes à l'aide de déformations élastiques qui tiennent compte des paramétrisations temporelles. Les méthodes proposées s'inspirent de l'analyse des cycles respiratoires de souris afin d'identifier des modalités de ventilation et d'évaluer les changements respiratoires chez des souris de génotypes différents après exposition à une molécule affectant la respiration. Une première méthode compare les cycles respiratoires à l'aide d'un algorithme de clustering s'appuyant sur la distance Dynamic Time Warping. Conçus comme référence, les résultats expérimentaux montrent que les clusters reflètent des modalités de ventilation liées à des génotypes et à des réponses à l'exposition. La seconde méthode crée des représentations vectorielles de séries

temporelles échantillonnées de manière irrégulière et de longueur variable par le vecteur paramétrant les déformations qui font correspondre une série temporelle de référence aux autres. Cette approche s'appuie sur une méthode d'analyse de formes nommée Large Deformation Diffeomorphic Metric Mapping (LDDMM). Celle-ci est modifiée pour maintenir la structure spatio-temporelle des données tout en garantissant la bijectivité des représentations. Cette nouvelle méthode fournit des informations statistiques et visuelles sur les formes et les déformations. Une simple analyse statistique révèle que les déformations les plus importantes ont une signification physiologique permettant de mieux comprendre les modalités de ventilation selon le génotype et l'exposition.

Abstract

Time series are prevalent in biomedical applications where they frequently display recurring or abnormal patterns that hold significant information for statistical analysis. A notable example is the heartbeat in electrocardiograms, a recurring pattern whose shape can vary depending on the underlying condition, making it an important feature for diagnosing heart-related diseases. However, comparing such patterns requires specialized mathematical tools lying at the intersection between machine learning for time series and shape analysis. While the shape analysis community has partially addressed the case of time series, shape-related approaches from machine learning for time series have achieved great success in various applications. This thesis aims to combine the strengths of both fields to propose methods suitable for biomedical research depending on temporal data. Particular attention will be given to methods' interpretability through visual interpretation of patterns and deformations, as it is key for meaningful interaction between the data and biomedical researchers.

The thesis is structured into two parts: the first focuses on searching for and discovering valuable patterns in time series, while the second concentrates on pattern comparison.

The first part tackles the challenge of searching or discovering patterns in long time series with distances independent of some irrelevant sources of variability modeled with a group of deformations. To that end, a general framework for constructing deformation-invariant distances is introduced. This framework extends the well-known Z-normalized Euclidean distance, invariant to amplitude scaling and offset shifts, by allowing customization of the group of deformations. The custom distances can be integrated into state-of-the-art algorithms for similarity search and motif discovery without efficiency loss. Additionally, an interpretable and interactive algorithm for motif discovery has been developed. This algorithm maps a time series onto a graph which is then summarized into a diagram providing a visual interpretation that facilitates the identification of recurring patterns. Furthermore, an interactive application has been designed for biomedical researchers, leveraging the algorithm's interpretability and efficiency for effective motif discovery.

The second part focuses on comparing temporal patterns using elastic deformations that notably account for time warping. The proposed methods are driven by the analysis of mice respiratory cycles recorded via plethysmography to identify ventilation modalities and assess the respiratory changes in mice with different genotypes after exposure to a drug affecting respiration. The first method compares respiratory cycles with a clustering algorithm based on the Dynamic Time Warping distance. Designed as a baseline, experimental results show that clusters have physiological relevance, reflecting genotype-specific ventilation modalities and responses to drug exposure. The second method creates fixed-size vector representations of irregularly sampled and variable-length

time series by the vector parametrizing the deformations that map a reference time series to the observed ones. This approach draws on the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework from shape analysis, which is refined to maintain the spatiotemporal structure of the deformed time series while ensuring the bijectivity of the embedding. This method provides both statistical insights and visual interpretations of shapes and deformations. A simple statistical analysis reveals that the deformations responsible for most variability carry physiological significance, offering insights into ventilation modalities with respect to genotype and drug exposure effects.

Glossary

$f \in \mathsf{C}^m(\mathsf{U}, \mathbb{R}^d)$	A m-continously differentiable function from the open $\mathsf{U} \subset \mathbb{R}^n$ to $\mathbb{R}^d$.
$\phi \in \mathsf{D}(\mathbb{R}^d)$	A diffeomorphism from $\mathbb{R}^d$ to $\mathbb{R}^d$ (see Chapter 6).
$f \in \mathsf{L}^2(\mathsf{I}, \mathbb{R}^d, \mu)$	Square integrable function from $\mathsf{I} \subset \mathbb{R}$ to $\mathbb{R}^d$ for the measure μ.
$f \in \mathsf{M}(\mathsf{I}, \mathbb{R}^d)$	Borel measurable function from $\mathsf{I} \subset \mathbb{R}$ to $\mathbb{R}^d$.
$\mathbf{s} \in \mathbb{R}^{n \times d}$	Discrete time series of dimension d and length n.
$\mathbf{s}_i^l \in \mathbb{R}^{l \times d}$	subsequence of length l starting at index i of the time series $\mathbf{s} \in \mathbb{R}^{n \times d}$.
A-PEPA	Adaptive PersistentPattern algorithm (see Chapter 3).
DTW	Dynamic Time Warping. A distance between discrete time series invariant to time parametrization (see Section 1.2.1).
ACh	Acetylcholine. A neurotransmitter that is notably key for the mediation of muscle contraction and breathing regulation (see Section 4.2).
AChE	Acetylcholinesterase. An enzyme that terminates a signal transmission by destroying ACh by hydrolysis (see Section 4.2).
BChE	Butyrylcholinesterase. An enzyme closely related to AChE that is also capable of hydrolyzing ACh (see Section 4.2).
CNS	Central Nervous System.
DCP	Double Chamber Plethysmogram (see Section 4.1).
ECG	Electrocardiogram.
EEG	Electroencephalogram.
FFT	Fast Fourier Transform. An algorithm to compute the Discrete Fourier Transform (DFT).
K-NN	A graph that connects each vertice to its K Nearest Neighbors (see Section 2.1.3).
LDDMM	Large Deformation Diffeomorphic Metric Mapping. A framework from shape analysis (see Chapter 6)
LT	Linear Trend (see Section 2.3).
NMJ	Neuromuscular Junction (see Section 4.2)
PEPA	PersistentPattern algorithm (see Chapter 3).
PNS	Peripheral Nervous System.
PPG	Photoplethysmogram.
PVC	Heartbeat anomaly: Premature Ventricular Contraction.
RKHS	Reproducible Kernel Hilbert Space.
TS-LDDMM	Acronym of the unsupervised representation algorithm for irregularly sampled and variable length time series presented in Chapter 6.

Contents

Chapter 1

Introduction

Key points:

1. Time series data, common in biomedical applications, often exhibit recurring or abnormal deterministic patterns that are valuable for statistical analysis due to their consistent appearance across different subjects. However, effectively utilizing these patterns requires appropriate mathematical tools for precise comparison.

2. The comparison of temporal patterns lies at the intersection of machine learning for time series and shape analysis, where sources of variability are modeled as deformations of a reference shape. However, the case of time series has been partially dealt with shape analysis. In contrast, shape-related works from machine learning for time series have shown great success, suggesting that both communities could benefit from each other.

3. This thesis aims to leverage machine learning for time series and shape analysis to propose methods suitable for biomedical research relying on temporal data. Special emphasis is also placed on the visual interpretation of patterns and deformations, which is crucial to many biomedical research.

Contributions:

1. A general framework for shape analysis is proposed, forming the foundation for the subsequent chapters. This framework defines time series, the group of deformations that can act on them, and how these deformations affect the time series.

Contents

1.1 Motivation

Analyzing experimental data to validate or refute hypotheses is a core principle of modern science. This approach is especially prominent in biomedical research, where it drives advancements in understanding biological structures and functions, developing novel treatments, or improving diagnostics and medical practices to enhance human health. Recent technological innovations have greatly facilitated the non-invasive acquisition of biomedical data, with time series playing a particularly significant role [GKK20; Fer17].

For instance, electroencephalograms (EEG), as shown in Figure 1.1a, record the electrical brain activity by placing electrodes around the skull. Among many other applications, EEGs play an important role in diagnosing neurological disorders like epilepsy or narcolepsy and in studying brain function in response to external stimuli [SBH74]. Similarity Electrocardiogram (ECG), depicted in Figure 1.1b, measures the heart's electrical activity via electrodes placed on the body, aiding in the diagnosis of several pathological heart conditions like arrhythmia or assessing the heart's response to various clinical treatments [Vic+19]. In contrast, gait signals, illustrated in Figure 1.1c, monitor footsteps' angular velocity with inertial measurement units, offering valuable insights for the rehabilitation of patients with reduced mobility due to conditions such as Parkinson's disease or stroke [Bar+15].

With human health at stake, biomedical data analysis requires mathematically founded and statistically grounded tools to facilitate meaningful interactions between the data and biomedical researchers or practitioners.

Structured data. Biomedical time series often exhibit deterministic patterns that reflect the subject's physiological state. For example, specific EEG waveforms, such as K-complexes (sharp peaks) and sleep spindles (sinusoidal patterns), are characteristic of the second stage of sleep, as shown in Figure 1.2a. Similarly, the shape of heartbeats recorded by ECG can be altered by conditions like premature ventricular contractions (PVC), Figure 1.2b. Gait signals also vary between healthy individuals and those affected by neurological disorders Figure 1.2c.

These recurring patterns are valuable in biomedical research as they are consistently observed across different subjects, making them robust features for statistical analysis.

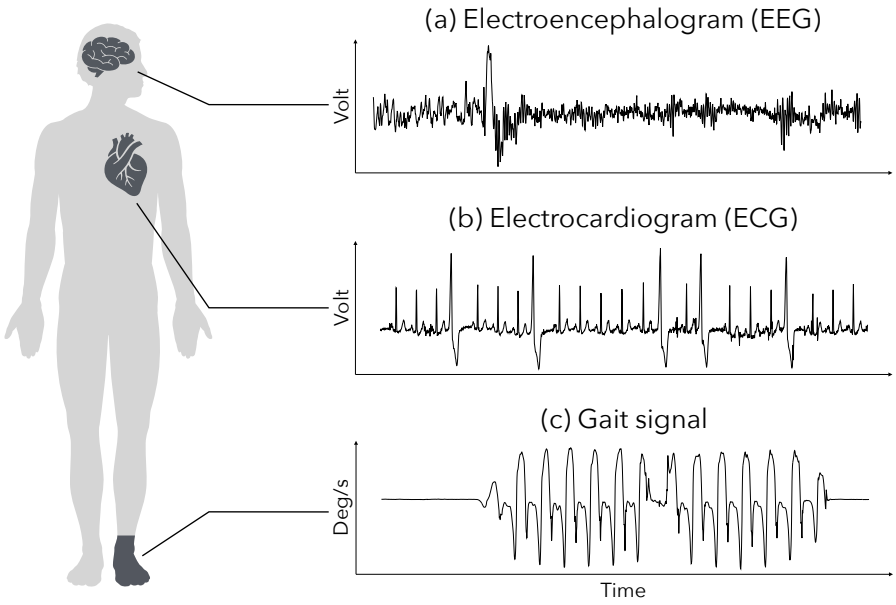


Figure 1.1 Illustrations of non-invasive biomedical time series. **(a)** Electroencephalogram (EEG) measuring brain's electrical activity with electrodes, **(b)** Electrocardiogram (ECG) measuring heart's electrical activity, and **(c)** Gait signal measuring footsteps' angular speed with inertial measurement unit.

However, effectively leveraging these patterns requires appropriate mathematical tools for accurate comparison.

Features on shapes. Interestingly, comparing such deterministic patterns boils down to comparing their shape. Historically, it has been done by comparing handcrafted features extracted from patterns, as illustrated in the case of heartbeats in Figure 1.3. However, such features tend to be precise and localized, which may lead to the loss of discriminative information. More recently, machine learning and deep learning algorithms have been used to learn features directly from the data. However, these methods often require large datasets, an unaffordable luxury in some biomedical contexts. Additionally, ensuring the reliability and interpretability of learned features is an active area of research that is essential in biomedical research.

While the first approach risks being overly reductionist, and the second tends to over-parameterize, a third approach leveraging the notion of shape in patterns could be investigated to address both limitations.

Shape & time series. The shape analysis community has laid out a mathematical framework for studying the shapes of geometric objects, where the concepts of shape and deformation are deeply interconnected. For example, a piece of paper can be deformed by folding or unfolding it. Separating the object from its deformations leads to two analytical approaches:

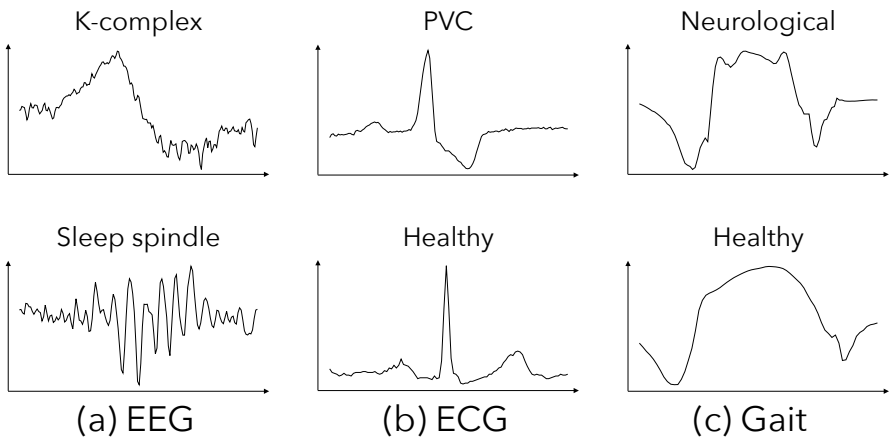


Figure 1.2 Illustrations of deterministic patterns. **(a)** In EEG, K-complex and sleep spindles indicate a stage 2 sleep. **(b)** In ECG, heartbeats of subjects suffering from premature ventricular contraction (PVC) have a different profile compared to healthy subjects. **(c)** In gait signal, the footstep of a subject with a neurological pathology differs from that of a healthy subject.

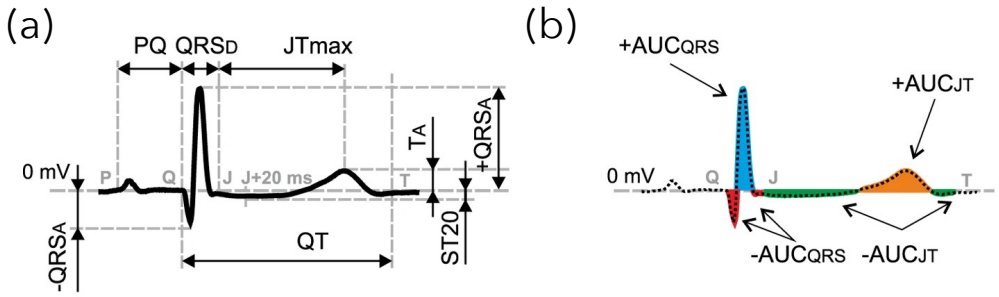


Figure 1.3 From [Mar+17]. Illustrations of standard features for describing heartbeat's shape from ECG toward automatic classification of ventricular premature and ischemic heartbeats. It includes features accounting for **(a)** duration of specific intervals and amplitude of some peaks, **(b)** area under the curve on specific intervals.

1. **Comparing objects independently of the deformations.** For instance, independently of folding, a plane paper and a folded paper are considered the same object.

2. **Comparing objects by quantifying the deformations.** For instance, a paper folded 4 times and a plane paper differ by 4 folds.

Compared to earlier approaches, shape analysis adjusts the complexity of the problem by incorporating expert knowledge into the design of the deformation set. Deformations are carefully selected to account for meaningful sources of variability, ensuring that deformation-invariant features or deformations have biological significance.

While shape analysis primarily focuses on medical imaging to compare organs and tissues under spatial deformations, its application to time series (spatiotemporal data) remains relatively unexplored. Despite this, shape-based methods have demonstrated

significant success in tasks such as classification and clustering for time series. These methods typically rely on distances invariant to common time series deformations, such as amplitude scaling, offset shifts, and time warping, as illustrated in Figure 1.4.

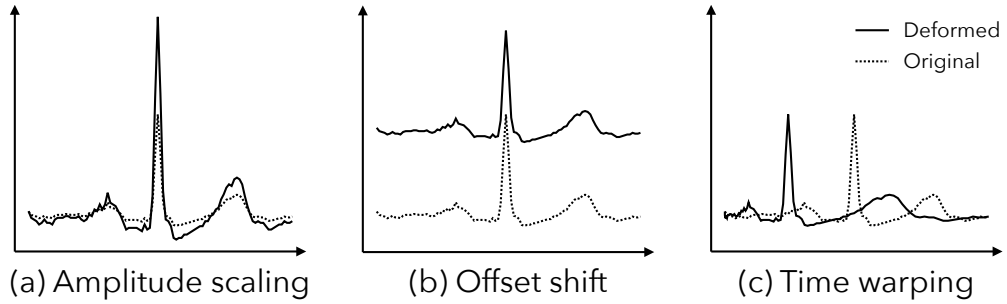


Figure 1.4 Illustrated on a heartbeat from ECG of common time series deformations including **(a)** amplitude scaling, **(b)** offset shift, and **(c)** time warping.

Thesis positioning. This thesis has been conducted at the Centre Borelli[1], a multidisciplinary research laboratory that brings together experts from various fields, including mathematics, computer science, neuroscience, biology, medicine, and clinical practice. This thesis aims to leverage machine learning for time series and shape analysis to propose methods suitable for biomedical research relying on temporal data. Special emphasis is also placed on the visual interpretation of patterns and deformations, which is crucial to many biomedical research.

Usecases. This thesis is divided into two parts. The first part focuses on querying specific patterns or discovering recurrent patterns in a long time series, independent of some predefined deformations. Intended as handy methods, they are tested on several biomedical temporal data, including ECG and EEG.

The second part focuses on unsupervised shape-based embedding methods for time series datasets. The development of such methods is motivated by research conducted at Centre Borelli to understand better the role of an enzyme in the regulation of respiration [Ner+19]. A detailed description of this research is provided in Chapter 4.

Situated at the intersection of machine learning for time series and shape analysis, the following section provides an overview of both research communities and the mathematical foundations necessary for applying shape analysis to time series.

1.2 At a crossroads

1.2.1 Machine learning for time series

Time series are everywhere. Times series appears in numerous fields of applications and raises diverse challenges. Among many examples outside of the biomedical field,

[1] `https://centreborelli.ens-paris-saclay.fr/en`

astronomers are interested in classifying billions of astronomical objects from massive datasets of photometric time series [JB20; Lin+12]. Seismologists aim at predicting upcoming earthquakes from real-time seismograms [BAM23]. Economists wish to detect fraudulent market manipulation from time series of financial transactions [KG22; GZ15]. Industrials wish to streamline their supply chain by predicting sales [RLM21], controlling their stock level [Avi03], or performing preventive maintenance [RBP11].

Facing such diversity of context and problems, researchers in machine learning for time series have organized their work around transversal tasks like classification or forecasting, as well as core criteria to evaluate methods.

Core criteria. Looking at the literature, algorithms are commonly evaluated with three criteria that englobe challenges encountered in most applications:

- **Efficiency:** Dealing with potentially large datasets of long time series, algorithms should be efficient both in computational time and memory usage.
- **Interpretability:** Steamed by applications in fields like industry or medicine, where algorithmic decisions may alter human health and well-being, algorithms should come with guarantees, notably in interpretability where the algorithm's output can be explained from the input data.
- **Performances:** To motivate the creation of algorithms that perform well across multiple application fields, researchers have established task-dependent metrics [SR24; JPJ24; Tat+18] and datasets [Pap+22a; God+21; Dau+19; Bag+18].

The proposed algorithms will be evaluated throughout the thesis in light of these criteria.

Transversal tasks. In several applications, the same tasks must be resolved, and many researchers in the time series community have focused their work around them [EA12b; Fu11]. In what follows, short descriptions of the most prevalent tasks are given:

- **Anomaly detection [SWP22]:** Detecting abnormal parts in a time series. The normal/abnormal behavior can be learned with or without supervision.
- **Classification [Bag+17]:** Predicting a time series class by training an algorithm with a paired label/time series dataset.
- **Clustering [ASW15]:** Grouping time series in homogeneous sets according to a similarity measure and without supervision.
- **Embedding [Li+17]:** Reducing the dimension of time series in time or space to gain performance and efficiency on downstream tasks.
- **Forecasting [LZ21]:** Predicting the future from past observations and relying on statistical properties of the underlying process.
- **Motif discovery [TL17]:** Detecting and locating local patterns that repeat themselves in a time series.

- **Segmentation [TOV20]:** Dividing a time series into homogeneous segments based on a measure of similarity or supervised training.
- **Similarity search [Pat+02]:** Querying a single time series or a dataset in search for occurrences of a given pattern in a time series.

In this thesis, contributions have been made to the task of similarity search in Chapter 2, motif discovery in Chapter 3, and embedding in Chapters 5 and 6.

Two scales. Most tasks related to time series are on one of two different scales. Some tasks like clustering focus on the global scale; they compare time series belonging to a dataset. Others, like motif discovery, focus on the local scale; they search for local events in a single long time series. In some situations, the task refers to both scales. For instance, anomaly detection refers to the detection of anomalous time series in a dataset and the detection of local anomalous events in a time series. Table 1.1 details the scale at which each task operates.

In addition, it is possible to pass from a local scale task to a global one with a proper segmentation algorithm. For instance, with an algorithm for segmenting heartbeats (Figure 1.2b), an ECG (Figure 1.1b) can be decomposed in a dataset of individual heartbeats, and thus comparable with global scale methods.

In this thesis, Part I focuses on local scale tasks, and Part II focuses on global scale tasks.

Table 1.1 Operating scale of common tasks on time series

Task	Local	Global
Anomaly detection	✓	✓
Classification		✓
Clustering		✓
Embedding		✓
Forecasting		✓
Motif Discovery	✓	
Segmentation	✓	
Similarity search	✓	✓

The distance jungle. Most algorithms that address the abovementioned tasks rely on distances between time series. As they are easily interchangeable, numerous distances have been proposed to improve performance in various contexts. Facing the distance jungle, researchers have led several experimental evaluations over the years [HMB24; Pap+20; AML19; Din+08]. For instance, a recent study compares 71 distances over 128 datasets [Pap+20]. The majority of the distances fall into two families:

- **Lock-step distances:** They compare time series of equal length and assume a one-to-one pairing between samples. They are known for their computational efficiency.

- **Elastic distances:** They can compare time series of different lengths, and given a distance between samples, they find the optimal pairing that minimizes the overall distance. They are known for their robustness to time warping.

In this thesis, contributions to both families have been made and are presented in Chapter 2 for lock-step distances and Chapter 6 for elastic ones.

Shape-based distances stand out. Among all distances, two are well established and considered as the baseline distance by many algorithms: The Z-normalized Euclidean distance [GK95] and the dynamic time warping distance (DTW) [SC78]. Both distances are, in fact, shape-based distances, meaning that they compare time series independently of some deformations.

Belonging to the lock-step family, the Z-normalized Euclidean distance is invariant to scale and offset deformations Figure 1.4ab. While basic, these deformations are pervasive in time series, and invariance becomes crucial in many applications. The Z-normalized Euclidean distance between $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^n$ is defined by:

$$d_Z(\mathbf{x}, \mathbf{y}) = \left\| \frac{\mathbf{x} - \mu_x \mathbf{1}}{\sigma_x} - \frac{\mathbf{y} - \mu_y \mathbf{1}}{\sigma_y} \right\| , \tag{1.1}$$

with $\mu_x = \frac{1}{n}\sum_{i=1}^{n} x_i$, $\sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_x)^2$, and $\mathbf{1} = (1, \dots, 1) \in \mathbb{R}^n$. Treated as Gaussian samples, mean and standard deviation are withdrawn from the samples so that the time series becomes invariant to the offset and scale deformations. The Z-normalized distance benefits from an efficient computation [ZM24] and has shown great success, especially in similarity search and motif discovery [ZM24; Yeh+16].

Belonging to the elastic family, the DTW is invariant to a common source of inter-individual variability: the time parametrization of the time series Figure 1.4c. In its original form [SC78], the DTW between $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^n$ is defined by:

$$dtw(\mathbf{x}, \mathbf{y}) = \min_{A \in \mathsf{A}_{m,n}} \langle A, \Delta \rangle_F, \quad \text{where:} \quad \Delta_{ij} = \|x_i - y_j\|^2 , \tag{1.2}$$

with $\mathsf{A}_{m,n} \subset \{0,1\}^{m \times n}$ the set of path matrices that connect the top-left corner to the bottom-right corner [CB17]. Many variants of the DTW have been proposed over the years, among which some intend to improve its robustness to noise [ZI18; CB17]. Recently, the DTW has been combined with optimal transport for comparing time series with heterogeneous output spaces to address the issue of domain adaptation while ensuring time warping invariance [Pai+23; Coh+21; JCG20]. Similarly, a recent work [Vay+20] has proposed a DTW-based distance also invariant to global deformations belonging to Stiefel manifolds. For instance, this distance is well suited to compare time series of motion recordings where the camera angle may differ between recordings. Note that it is not a metric as it does not guarantee the triangular inequality, and it is also quadratic in computation time. However, DTW-based distances are performant in many tasks on short time series datasets [Wan+13].

Showing great success, shape-based distances are a primer choice for many applications. However, the notion of shape in time series has not been explored to its fullness, and further improvements are still possible by taking on the shape analysis literature.

1.2.2 Shape analysis

Comparing shape. Shape analysis refers to methods that compare geometrical objects like surfaces or curves with special attention to modeling the inter-object variability. As illustrated in Figure 1.5, first applications were in biology [DM16], where researchers were interested in anatomical differences between species independently to some source of variability modeled by dilations, translations, or rotations. Such analysis is known as the Ordinary Procruste analysis [HC62].

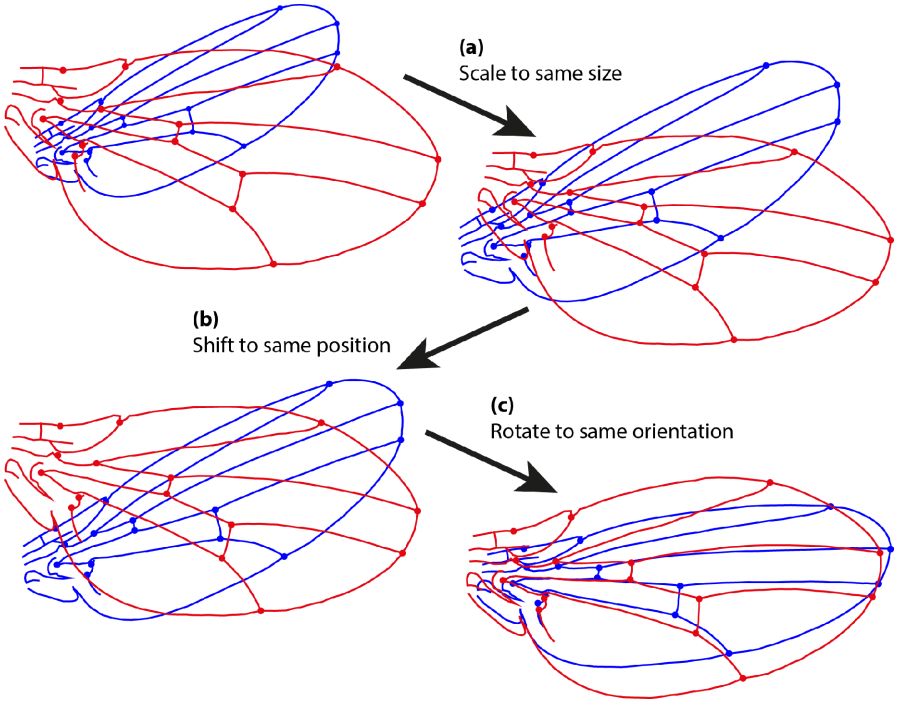


Figure 1.5 From [Kli15], the shapes of insect wings are compared through Ordinary Procrustes analysis. **(a)** both wings are scaled to the same scale, removing the dilatation variability, **(b)** barycenter of both wings is translated to the neutral, removing translation variability, and **(c)** both wings are rotated to the same orientation, removing the rotation variability. Finally, the shape distance between wings is the sum of Euclidean distance paired landmarks.

Toward statistical metohds. More recently, shape analysis has been applied in fields like computer vision [Wei18; WM18; You12], medical imaging [Sto+24; Dub+18; Mor+08], or computational anatomy [Gas+22; MTY02; GM98] where statistical methods play a central role in the scientific process. For instance, several studies have focused on the relationship between the shape of the Hippocampus shape and Alzheimer's disease [Wen+20; Chu+09; Wan+07], and others to the relation between the heart shape and some malfunctions [Gua+24; Man+11; Hel+05].

Unfortunately, classical statistical methods are unsuited to shape spaces as shape spaces are generally not endowed with a vectorial structure. For instance, the pixel-wise sum of two brain MRI does not results in a brain MRI. The development of statistical

methods dedicated to shape spaces has become an active research topic over the past two decades [Fey20].

Metric on shape space. While defining a suitable vectorial structure on a shape space is challenging, quantifying the difference between shapes is easier. A large body of work has been focusing on defining metric structure on shape spaces which are evaluated around three criteria:

- **Relevance to the application field:** Encompass source of variability according to their effect on shapes.
- **Mathematically founded:** Inherit mathematical properties relevant for downstream methods, notably statistical ones.
- **Computationally efficient:** Scalable to large dataset.

Deformation and group action. A conceptual approach to define metric on shape space has been introduced [Gre94]. Specifically, sources of variabilities are modeled as deformations of the ambient space in which the geometrical objects belong. The set of deformations is endowed with a group structure, and its action on the geometrical objects is described with a group action.

Definition 1 (Group action)**.** *A group* G *with neutral* e *acts on the left on a set* M*, if there exists a map* $a : \mathsf{G} \times \mathsf{M} \mapsto \mathsf{M}$ *that verifies:*

$$1)\ a(e, m) = m, \quad \forall m \in \mathsf{M}$$
$$2)\ a(g, a(h, m)) = a(gh, m), \quad \forall (g, h) \in \mathsf{G}^2, \forall m \in \mathsf{M}.$$

Remark 1. *The right action can also be defined; it suffices to replace the second property with* $a(g, a(h, m)) = a(hg, m)$*. To simplify notations, left (resp. right) actions are denoted* $g \times m \mapsto g \cdot m$ *(resp.* $g \times m \mapsto m \cdot g$*).*

For a group G that acts on the left on a set M, the orbit of an element $m \in \mathsf{M}$ is the set $[m] = \{g \cdot m \mid g \in \mathsf{G}\}$. The action of G on M is said transitive if for any $m \in \mathsf{M}$ its orbit is the whole set: $[m] = \mathsf{M}$. Different strategies to define metrics should be considered depending on whether or not the transitivity property holds.

Nontransitive action. For nontransitive actions, distances are designed to compare shapes independently to the set of deformations. Formally, the set of independent orbits, denoted M/G, called quotient space, is not reduced to a singleton. Each orbit represents a shape, and the quotient space M/G must be equipped with a metric structure.

Theorem 1. *Let* (M, d) *be a metric space and* G *a group that acts nontransitively on the left on* M*. The function* $\tilde{d}$ *defined by:*

$$\tilde{d}([m], [m']) = \inf_{(g,g') \in \mathsf{G}^2} d(g \cdot m, g' \cdot m')$$

is a metric on M/G*, if the orbits are closed subset of* M *for the topology induced by* d*.*

In addition, if d *is* G*-equivariant, ie* $d(g \cdot m, g \cdot m') = d(m, m')$*,* $\tilde{d}$ *also verifies:*

$$\tilde{d}([m],[m']) = \inf_{g \in \mathsf{G}} d(m, g \cdot m')$$

Proof. See chapter 12 from *Shapes and diffeomorphisms*, [You10]. □

Example 1 (Rotation and translation invariance)**.** *A distance invariant to rotation and translation is a straightforward application of Theorem 1.*

Formally, suppose two set of paired landmarks $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ *and* $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_N)$ *living in the ambient space* $\mathbb{R}^d$*. The objects* $\mathbf{x}$ *and* $\mathbf{y}$ *belong to the same orbit if there exists a rotation* $R \in \mathsf{SO}(d)$ *and a translation* $\tau \in \mathbb{R}^d$ *such that:*

$$\mathbf{y} = R\mathbf{x} + \tau \qquad i.e. \qquad \forall i \in [\![1, N]\!],\ \mathbf{y}_i = R\mathbf{x}_i + \tau. \tag{1.3}$$

If $\mathbf{x}$ *and* $\mathbf{y}$ *are not collapsed to a single point, by translation and rotation equivariance of the Euclidean distance, the invariant distance is defined as:*

$$d_{RT}(\mathbf{x}, \mathbf{y}) = \inf_{(R_1, \tau_1, R_2, \tau_2)} \|(R_1\mathbf{x} + \tau_1) - (R_2\mathbf{y} + \tau_2)\| = \inf_{(R, \tau)} \|(R\mathbf{x} + \tau) - \mathbf{y}\| \tag{1.4}$$

Example 2 (Time parametrization invariance: the Square Root Velocity (SRV) framework)**.** *Originating from shape analysis, the Square Root Velocity framework [Sri+10] aims to compare curves independently of their time parametrization. It proposes a distance invariant to time parametrization built through the strategy of Theorem 1.*

Formally, let $\mathsf{M} \subset \mathsf{L}^2([0,1], \mathbb{R}^d)$ *be the set of integrable open curves that are differentiable, with a first derivative also integrable, and such that for any* $c \in \mathsf{M}$*,* $c(0) = 0$*. The goal is to define a distance between curves that is invariant to the action of the group* $\mathsf{G} = \{\gamma \in C^1([0,1],[0,1]) \mid \gamma(0) = 0,\ \gamma(1) = 1,\ \gamma'(t) > 0\ \forall t\}$*.*

To that end, let consider the bijective embedding map F *such that for any curve* $c \in \mathsf{M}$*,* $F(c)$ *is the curve defined as:*

$$F(c) : t \mapsto \begin{cases} c'(t)/\sqrt{\|c'(t)\|},\ if\ c'(t) \neq 0 \\ 0,\ else \end{cases}, \tag{1.5}$$

and the distance d *on* M*:*

$$d :\ (c_1, c_2) \in \mathsf{M} \times \mathsf{M} \mapsto \int_0^1 \|F(c_1)(t) - F(c_2)(t)\|^2 dt\ , \tag{1.6}$$

the distance d *is* G*-equivariant, meaning that for any curves* c_1, c_2 *and time parametrization* γ*,* $d(c_1 \circ \gamma, c_2 \circ \gamma) = d(c_1, c_2)$*. According to Theorem 1, the application:*

$$\tilde{d} :\ ([c_1],[c_2]) \in \mathsf{M}/\mathsf{G} \times \mathsf{M}/\mathsf{G} \mapsto \inf_{\gamma \in \mathsf{G}} \int_0^1 \|F(c_1 \circ \gamma)(t) - F(c_2)(t)\|^2 dt\ , \tag{1.7}$$

is a pseudo-distance that compares curves up to their time parametrization, and with some technical considerations [Sri+10], it defines a distance on M/G*.*

Transitive action. With transitive action, it is always possible to find a deformation that maps one geometrical object to the other. The deformation deforms the ambient space of the first object to map it onto the second. The interest of transitive action lies in the possibility of describing the transformation of one object to another at a global or a local scale and for any point of the ambient space. Unfortunately, the strategy described in the nontransitive case is not transferable to the current case. However, defining a distance on the shape space M is still possible if the group G can be endowed with a metric structure. Intuitively, distances defined through the following theorem quantify "how much" the source object has to be deformed to be mapped on the target one.

Theorem 2. *Let (G, e) be a group that acts transitively and on the left on the set M. If d_{G} is a right-equivariant metric on G, ie $d_{\mathsf{G}}(gh, g'h) = d_{\mathsf{G}}(g, g')$, then $\tilde{d}$ defined by:*

$$\tilde{d}(m, m') = \inf_{g \in \mathsf{G}} \{d_{\mathsf{G}}(e, g) \mid g \cdot m = m'\}$$

is a metric on M if $\{g \in \mathsf{G} \mid g \cdot m_0 = m_0\}$ is closed for the topology induced by d_{G} and for a fixed $m_0 \in \mathsf{M}$.

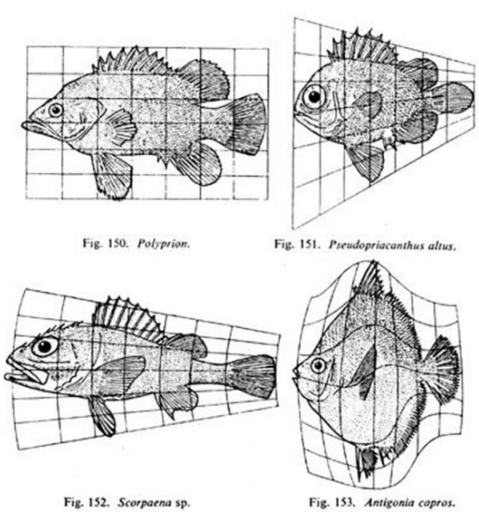
Proof. See chapter 12 from *Shapes and diffeomorphisms*, [You10]. □

Rooted in the work of the biomathematician D'Arcy Thompson [Tho17] who first described the mapping from one species to another through a geometrical deformation, see Figure 1.6a, the group of diffeomorphisms has received particular attention in shape analysis for transitive actions. Intuitively, diffeomorphisms are smooth bijective applications with a smooth inverse. Such deformations can be generated through ordinary differential equations, making the distances induced by this group relevant for any biomedical application where the deformation of a shape evolves smoothly across time. For instance, a child's brain forms smoothly during pregnancy, see Figure 1.6b; the brain shape evolution can be compared at a population level to provide valuable information to clinicians and counseling to the parents [GBA21].

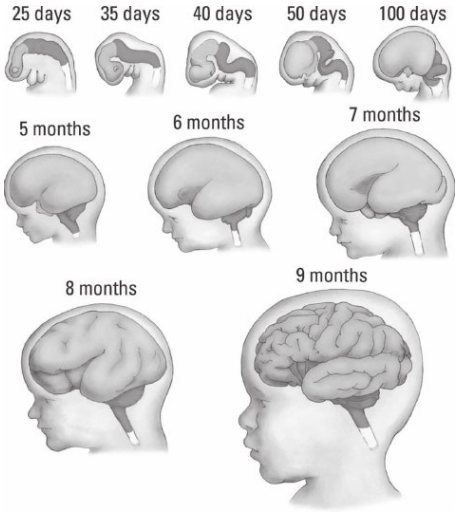
Time series are not curves. Time series and curve refer to the same mathematical object: an application from a closed interval $\mathsf{I} \subset \mathbb{R}$ taking value in a space E. However, the difference comes from the challenges raised by diverse application fields that are dealt with by two different communities.

Steamed by applications in computer vision or medical imaging, curves often refer to detoured objects, see Figure 1.7a, and have been widely studied in shape analysis [Bau+21; You10]. Here, the temporal aspect of curves simply refers to the parametrization of the detoured object and does not carry any meaningful information. Therefore, all curves are defined on the same closed interval I, and the focus is on comparing the shape of curves independently of any time parametrization.

To oppose time series to curves, let us consider a practical example. Bradycardia is a disease where subjects have an abnormally low heartbeat rate, causing oxygen deficiency. The difference between healthy and unhealthy subjects can be seen from electrocardiograms (ECG), see Figure 1.7b. Compared to healthy subjects, the heart cycle of subjects suffering from bradycardia presents a long pause at the end of the



(a) Fish mapping, from [Tho17].



(b) Child brain development, from [KF09].

Figure 1.6 (a) Fish on the top-left corner is deformed to map other fishes. The underlying assumption of D'Arcy Thompson is that deformations between closely related species should be "small". **(b)** A schematic description of a child's brain development during pregnancy. Failure to develop certain parts of the child's brain during pregnancy can lead to cognitive malfunction. Detecting such abnormalities and differentiating them from potential developmental delays is crucial to clinicians [GBA21].

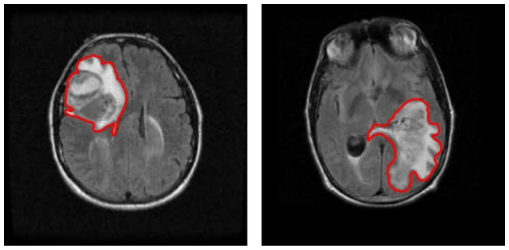
heart contraction. Surprisingly, healthy and unhealthy patient cycles are identical when compared independently of any time parametrization. The discriminative information resides in the time parametrization of the heart's cycle, which should be included in the notion of shape for time series.

The previous example shows that a straightforward application of methods designed for curves to time series is restrictive in some situations. On the other hand, the large and fruitful body of work around the notion of shape needs to be added to the time series community. This remark motivates the positioning of this thesis to extend some notion of shape analysis to the context of time series.

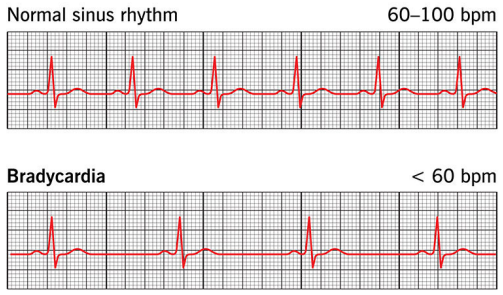
1.2.3 A general framework for shape analysis on time series

According to the previous section, three things must be defined in order to establish a notion of shape: a set of geometrical objects, a group of deformations and the action of the group on the set. The following paragraphs present these sets and the action in the context of time series and in the most generic way possible. The definition of shape-based metrics will be the focus of the next chapters where the generic framework will be declined to more specific cases.

Time series representation. Looking at the literature [Bau+21; Wil17], time series are commonly represented in two ways:



(a) Brain MRIs, from [Bha+18]



(b) ECGs, from Cleveland clinic website

Figure 1.7 (a) Curves representing detoured brain tumors from MRIs. **(b)** Electrocardiogram (ECG) difference between a healthy subject and a subject suffering from bradycardia, a disease where the heart has a slow contraction rate. While hearts' contractions are identical, heartbeats' cycles have long pauses after contraction in bradycardia. At the scale of individual heartbeat cycles, methods for comparing curves independently of time parametrization won't be able to differentiate healthy subjects from subjects suffering from bradycardia. From the website: `https://my.clevelandclinic.org/health/diseases/17841-bradycardia`.

- **The functional representation:** A time series is a function f from a closed interval $\mathsf{I} \subset \mathbb{R}$ taking value in $\mathbb{R}^d$.
- **The discrete representation:** A time series is a sequence $(f(t_1), \dots, f(t_n)) \in \mathbb{R}^{n \times d}$ sampled at time $t_1 < \dots < t_n \in \mathsf{I}$.

While the functional representation of a time series is exact, its discrete counterpart is an approximation whose error depends on the sampling. Going toward shape analysis, the functional representation of time series is more appealing to the mathematician as it is exact and prevents the trouble of dealing with sampling. However, we only have access to discrete time series representations in practice. It motivates the need for bridges between functional and discrete representation to merge the gap between theory and applications.

This thesis will mainly consider the functional representation to define shape-based distances between time series. Additionally, efforts will be made to decline such distances to the discrete case and to provide some convergence guarantees to the exact distance as the discretization gets refined.

Admissible set of time series. Compared to the set of curves that is the set of all continuous functions from the same closed interval $\mathsf{I} \subset \mathbb{R}$ to $\mathbb{R}^d$, the admissible set of time series differs in two ways. The continuity assumption should be revoked as it does not hold in several application fields. For instance, the electrical consumption of appliances often behaves like a binarized signal.

More importantly, the restriction of defining functions on the same closed interval I should also be revoked. Indeed, going back to the bradycardia disease example, when comparing heart cycles, the discriminative information resides in the time parametrization and especially the length of the interval on which the function is defined.

The set of admissible time series should encompass such differences and can be defined in the most general sense as the following union:

$$\mathsf{F} = \{(\mathsf{I}, f) \mid \mathsf{I} \in \mathcal{I} \text{ and } f \in \mathsf{M}(\mathsf{I}, \mathbb{R}^d)\} \ , \tag{1.8}$$

where $\mathcal{I}$ is the set of closed intervals of $\mathbb{R}$ and $\mathsf{M}(\mathsf{I}, \mathbb{R}^d)$ is the set of Borel measurable functions from I to $\mathbb{R}^d$. Note that this large set encompasses most of the time series encounters in applications. However, this set has little structure and the next chapters will focus on subsets that present more structure to ease the definition of metrics.

Admissible group action for time series. Group actions have been introduced in shape analysis to model the action of a deformation on a geometrical object. Looking at a time series $f : \mathsf{I} \mapsto \mathbb{R}^d$, a deformation that would make sense is a combination of a distortion $h : \mathsf{I} \mapsto \mathbb{R}^d$ and a time parametrization $\gamma : \mathsf{I} \mapsto \mathsf{J}$ that would lead to the deformed time series: $g = (f + h) \circ \gamma^{-1}$.

To properly define a group action on the admissible set of time series F, let us model the distortions by the set of Borel measurable functions $\mathsf{M}(\mathbb{R}, \mathbb{R}^d)$ and the time parametrization by the set of strictly increasing homeomorphisms $\mathsf{H}^+(\mathbb{R})$. The set $\mathsf{M}(\mathbb{R}, \mathbb{R}^d) \rtimes \mathsf{H}^+(\mathbb{R})$ with the composition rule: $(h_2, \gamma_2) \times (h_1, \gamma_1) = (h_1 + h_2 \circ \gamma_1, \gamma_2 \circ \gamma_1)$ forms a group which can act on F by the left action:

$$(h, \gamma) \cdot (\mathsf{I}, f) = \big(\gamma(\mathsf{I}), (f + h) \circ \gamma^{-1}\big) \ . \tag{1.9}$$

Note that this action is transitive, meaning that for any admissible time series (I, f) and (J, g), there exists a deformation (h, γ) such that $(h, \gamma) \cdot (\mathsf{I}, f) = (\mathsf{J}, g)$. Even more so, there is a multitude of deformations that map (I, f) to (J, g) as for any time parametrization γ, the distortion whose restriction on I is equal to $g \circ \gamma - f$ ensures the mapping. In summary, this action on time series is very expressive and offers many ways to model deformations relevant to applications.

In terms of notations, the group $\mathsf{M}(\mathbb{R}, \mathbb{R}^d)$ alone refers to **rigid deformations**, while the groups $\mathsf{H}^+(\mathbb{R})$ and $\mathsf{M}(\mathbb{R}, \mathbb{R}^d) \rtimes \mathsf{H}^+(\mathbb{R})$ refers to **elastic deformations**.

Going back to the shape analysis literature, the admissible group action falls into the notion of functional shape [CCT17; CT14], an emerging problem in computational anatomy [MQ09].

Applications based simplification. Simplifying the admissible group action comes at the cost of restricting the set of time series and deformations for the gain of additional structure, which in turn allows the definition of shape-based distances. Depending on the application, simplifications can be made following the two strategies:

- **Invariance to a set of deformations:** The goal is to compare time series independently to some deformations like amplitude, offset, time parametrization, and more. The simplification of the group action leads to a nontransitive action which, according to Theorem 1, leads to a deformation-invariant metric if the set of time series can be endowed with a metric equivariant to the group of deformations. Such metrics will be explored in Chapter 2. Notably, It will be shown that this

framework includes the well-established Z-normalized Euclidean distance (eq. (1.1)) as well as other appealing distances.

- **Quantification of some meaningful deformations:** The goal is to compare time series by the deformation that maps one time series to the other. The group of deformations is picked to carry meaning to the application, and the resulting group action satisfies the transitivity property. According to Theorem 2, a deformation-related metric between time series can be established if the group of deformations can be endowed with a metric equivariant to itself. Such strategy will be explored in Chapter 6 by deriving well-established methods from shape analysis to the case of time series.

Conclusion. The previous paragraphs present a general framework for the shape analysis on time series by leveraging an expressive group action on time series. Going to the application side, two strategies have been presented to create shape-based metrics on time series. This framework forms the foundations on top of which this thesis is conducted.

1.3 Thesis outline

The thesis is organized as follows:

Part I focuses on querying specific patterns or discovering recurrent patterns in a long time series, independent of some predefined deformations. Specifically, it deals with local scale tasks on time series, including similarity search and motif discovery, and addresses invariance to predefined groups of rigid deformations.

- **Chapter 2** tackles the problem of searching for occurrences (repetitions) of predefined patterns within a single long time series. The first section reviews related work on time series similarity search, with a particular focus on exact methods that use lock-step distances. The algorithmic properties contributing to their efficiency are detailed. Building on these properties, the second section introduces a general framework for constructing distances invariant to user-defined sets of deformations while ensuring equivalent computation time complexity as state-of-the-art methods. The final section applies this framework to develop a distance invariant to amplitude scaling, offset shift, and linear trend. This distance proves valuable in cases where time series are affected by trend-induced deformations, as demonstrated experimentally.

- **Chapter 3** introduces a novel algorithm for motif discovery. Following a comprehensive review on motif discovery, the second section presents the proposed algorithm, called PEPA, which enables the discovery of variable-length motifs. This algorithm embeds a time series into a graph and uses persistent homology to summarize the graph in a diagram. Motifs are then identified through a visual interpretation of the diagram. Although PEPA requires the user to specify the number of motifs to discover, an adaptive version, A-PEPA, is also presented, which uses a simple heuristic to infer this number. The subsequent section evaluates the algorithm's

performance on a benchmark dataset developed during this thesis alongside ablation studies. A web application is also presented, demonstrating how the algorithm's efficiency and visual interpretation can be used for interactive motif discovery.

Part II focuses on unsupervised shape-based embedding methods for time series datasets. Specifically, this global scale task is addressed by considering groups of elastic deformations that can either be restricted or quantified and embedded.

- **Chapter 4** presents the biomedical application motivating the development of the methods proposed in the subsequent chapters. Shortly, an enzyme plays an important role in regulating muscle activity and signal transmission within the nervous system. Some drugs inhibit the enzyme's action, which has severe consequences, notably on respiration, which are not yet fully understood. To investigate inhibition consequences, the respiration of mice with different genotypes exposed to inhibitors is monitored by plethysmography. The first section presents the monitoring equipment material (plethysmogram), and highlights the limitations of existing methods for analyzing such signals. Additionally, an algorithm for segmenting plethysmography signals into datasets of respiratory cycles (inspiration and expiration) is presented. The second section outlines the biological context and experimental protocol.

- **Chapter 5** presents, in the first section, a novel unsupervised baseline method for analyzing plethysmography signals. This method uses a DTW-based clustering algorithm to learn a symbolic embedding of respiratory cycles. The symbolization of plethysmography signals results in sequences of shape-based symbols. The following result and discussion illustrate, in particular, the interpretability of the method by presenting correspondences between symbols and physiological functions. Among several discoveries, the symbolic representations have highlighted genotype-dependent respiration modalities and a heterogeneous physiological response to inhibitor exposures.

- **Chapter 6** presents an embedding method called TS-LDDMM, which represents a time series by the vector that parameterizes the deformation mapping a reference time series to the target one. This method is built on the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework from shape analysis, introduced in the first section. LDDMM learns diffeomorphic deformations by solving specific differential equations. The second section adapts LDDMM to time series by establishing sufficient conditions on the differential system to ensure that the learned deformations preserve the spatiotemporal structure of the time series. While several benchmark comparisons and ablation studies are included in the appendices for conciseness, the experimental section focuses on the mice ventilation study. This section demonstrates how TS-LDDMM embeddings capture physiologically meaningful deformations with improved interpretability by analyzing associated statistical results. Specifically, TS-LDDMM embeddings helped to characterize mouse genotypes, ventilation modalities, and the effects of inhibitor exposure.

1.4 Contributions

Chapter 2:

1. A general framework is introduced for constructing deformation-invariant distances that can be integrated into state-of-the-art similarity search algorithms without compromising efficiency. Specifically, when sources of variability can be modeled as a group of deformations acting on time series as a vector subspace, a deformation-invariant embedding can be created, where the distance between embeddings is simply the Euclidean distance. This framework extends the well-known Z-normalized Euclidean distance.

2. As an example, the LT-normalized Euclidean distance is proposed, invariant to amplitude scaling, offset shift, and linear trend. This distance is locally robust deformations caused by a trend, and it has shown great success in several biomedical use cases.

Chapter 3:

1. This chapter introduces an algorithm called PersistentPattern (PEPA) for discovering variable-length motifs without requiring prior knowledge of the similarity between motif occurrences. PEPA works by embedding a time series into a graph and summarizing it through persistent homology, a tool from topological data analysis, which then allows the identification of relevant motifs from the graph's summary.

2. An adaptive version of the algorithm that infers the number of motifs to discover from the graph summary is also presented.

3. A benchmark of 9 labeled datasets, including 6 real-world datasets, is introduced for motif discovery. Empirical evaluations show that PEPA significantly outperforms state-of-the-art algorithms.

Chapter 4:

1. This chapter introduces a new algorithm for segmenting mice respiratory cycles (inspiration and expiration) from plethysmography signals. By incorporating physiological constraints, the method accurately detects the start of inspiration and expiration, offering greater robustness to respiratory variations compared to previous approaches.

Chapter 5:

1. This chapter introduces a baseline method that compares respiratory cycles using a DTW-based clustering algorithm, resulting in a shape-based symbolic representation where each symbol represents a cluster. Tracking these symbols over time results in a symbolic representation of plethysmography signals.

2. This approach facilitates the discovery of various ventilation modalities that are not captured by conventional descriptors. Notably, the symbolic representation helps identify genotype-specific adaptations to enzyme deficiency and reveals diverse responses to drug exposure.

Chapter 6:

1. Section 6.3 describes a class of deformations preserving the graph structure of time series while ensuring a transitive action (Theorem 3). Lemma 1 describe suitable Reproducible Kernel Hilbert spaces for encoding such deformations.

2. Appendix B.5 demonstrates the identifiability of the model by estimating the true generating parameter of synthetic data, and we highlight the sensitivity of our method concerning its hyperparameters.

3. Appendices B.6 and B.7 illustrate the quantitative interest of such representation on classification tasks on real shape-based datasets with regular and irregular sampling.

4. Section 6.5.2 showcases the interpretability of TS-LDDMM embedding on the analysis of mice ventilation.

1.5 Published papers

Chapter 2:

- Thibaut Germain, Charles Truong, and Laurent Oudre. "Linear-trend normalization for multivariate subsequence similarity search". In: *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*. IEEE. 2024, pp. 167–175

Chapter 3:

- Thibaut Germain, Charles Truong, and Laurent Oudre. "Persistence-based motif discovery in time series". In: *IEEE Transactions on Knowledge and Data Engineering* (2024)

- Thibaut Germain, Charles Truong, and Laurent Oudre. "Interactive motif discovery in time series with persistent homology". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2024, pp. 383–387

Chapter 5:

- Thibaut Germain et al. "Unsupervised classification of plethysmography signals with advanced visual representations". In: *Frontiers in Physiology* 14 (2023), p. 781

- Thibaut Germain et al. "Unsupervised study of plethysmography signals through DTW clustering". In: *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE. 2022, pp. 3396–3400

Chapter 6:

- Thibaut Germain et al. "Shape analysis for time series". In: *Advances in neural information processing systems* (2024)

Part I

Local scale tasks
&
Rigid deformations

Chapter 2

Similarity search

Key points:

1. Similarity search involves identifying occurrences of a query time series within a dataset or a single time series. This chapter specifically addresses the search for occurrences within a single time series while accounting for invariance to certain deformations.

2. Two exact and efficient search algorithms, known as the distance profile and matrix profile, are designed for the Z-normalized Euclidean distance, invariant to amplitude scaling and offset shift deformations. These algorithms achieve efficiency through the Fast Fourier Transform and a recursive distance formulation. These properties are emphasized, as they provide the foundation for a more general framework for designing invariant distances.

Contributions:

1. A general framework is introduced for constructing deformation-invariant distances that can be integrated into state-of-the-art similarity search algorithms without compromising efficiency. Specifically, when sources of variability can be modeled as a group of deformations acting on time series as a vector subspace, a deformation-invariant embedding can be created, where the distance between embeddings is simply the Euclidean distance. This framework extends the well-known Z-normalized Euclidean distance.

2. As an example, the LT-normalized Euclidean distance is proposed, invariant to amplitude scaling, offset shift, and linear trend. This distance is locally robust deformations caused by a trend, and it has shown great success in several biomedical use cases.

Associated paper:

- Thibaut Germain, Charles Truong, and Laurent Oudre. "Linear-trend normalization for multivariate subsequence similarity search". In: *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*. IEEE. 2024, pp. 167–175

Contents

2.1 Background

2.1.1 General context.

A computer science problem. Similarity search has received much attention since the early 90s, and it mainly addresses two problems:

1. Searching for time series that match a query time series.

2. Searching for subsequences in time series that match a query time series.

In other words, the first problem focuses on the global scale, while the second focuses on the local scale. In both cases, memory constraints and slow disk I/O have driven the development of efficient approximate algorithms, as querying becomes impractical with increasing dataset sizes [Lin+07]. When such limitations are present, the similarity search problem is often referred to in the literature as the time series indexing problem [ZM24]. Although indexing is not the primary focus of this thesis, the general principles of indexing methods are outlined in the following paragraph. When memory limitations do not apply, the problem of searching for subsequences within a single time series is known as distance profiling [ZM24]. Interestingly, this problem can be solved exactly for certain distances using efficient algorithms [ZM24]. Distance profiling is central to many tasks, such as motif discovery and anomaly detection [Rak+12b], for which a specific data structure, known as the matrix profile [Yeh+16], exists. The concept of distance profiling and the matrix profile structure are discussed in more detail in the following sections.

Indexing time series datasets. The foundational work in time series indexing has been laid out in early 90s [FRM94; AFS93]. The main idea is to create a relevant approximation of the time series so that the approximated dataset can fit in the main memory. Then, queries can be performed efficiently on the approximate dataset. Lastly, the exact time series are retrieved from the disk [Lin+07]. These methods are a fine combination between an algorithm for dimension reduction and an efficient data structure for querying. For instance, in [FRM94], they combined Direct Fourier Transform (DFT) coefficients with an R-tree data structure [Bec+90]. More generally, the querying data structure is a tree learned on the approximate dataset and specifically designed to handle the sequential nature of time series [Wan+24; Wan+13; AFS93].

Regarding the dimension reduction algorithms, early works have leveraged non-adaptive representations like wavelet, Fourier, or cosine coefficients, while more recent works leverage adaptive representations known as symbolic representations. The foundations of such representation have been laid out with the Symbolic Aggregate approXimation (SAX) [Lin+07], and they are built in three steps: segmentation of a time series, extraction of some features, and quantization of these features. Many variants of SAX have been proposed over the years, and in particular, the Indexing SAX family [Pal20] has been designed specifically for indexing time series datasets. Readers interested in symbolic representation can refer to the recent manuscript [Com24]. Finally, early works in time series indexing [FRM94; Keo+01] have established evaluation criteria focusing on time, space, query performances, and adaptiveness. For further readings on indexing, readers can refer to [EA12b; Fu11].

2.1.2 Distance profiling

An exact problem. Distance profiling refers to the problem of identifying, in a single time series, the subsequences similar to a query time series with the additional assumption that the memory is not overloaded [ZM24]. Indexing methods are still applicable in this context, but some other methods take advantage of the memory to efficiently and exactly solve the similarity search problem [ZM24; Rak+12b]. Such methods are preferable when the time to compute the indexing and query is longer than the time to compute similarities between the query and all subsequences. Distance profiling algorithms have been applied for various problems like the prediction of the electricity price [GCS20], the comparison of thermal signature in metal additive manufacturing [Cha+22], or the detection of chicken behaviors [Abd+20].

After a formal presentation of the distance profiling problem, the related methods are presented, and special attention is given to the case of the Euclidean distance, as it is key for several tasks like motif discovery.

A formal definition. Distance profiling requires the computation of all distances between a query and the subsequences of a time series. The sequence of the distances is known as the distance profile, and its formal definition is as follows:

Definition 2 (Distance profile)**.** *The distance profile between a time series* $\mathbf{s} = (s_1, \ldots, s_n) \in \mathbb{R}^n$ *and a query* $\mathbf{q} \in \mathbb{R}^l$*, such that* $l << n$ *and for a distance measure* $d : \mathbb{R}^l \times \mathbb{R}^l \mapsto \mathbb{R}_+$ *is*

the time series:

$$\left(d(\mathbf{q}, \mathbf{s}_i^l)\right)_{i \in [\![1, n-l+1]\!]} \tag{2.1}$$

where $\mathbf{s}_i^l = (s_i, \ldots, s_{i+l-1})$ *is the subsequence of* S *starting at index* i *and of length* l*.*

The overlap between subsequences should be handled carefully when querying from distance profiles. Unfortunately, there is no consensus on the definition of overlapping in similarity search [SL22]. In this work, two subsequences are considered as overlapping whenever they share some timestamps:

Definition 3 (Overlapping subsequences)**.** *The subsequences* $\mathbf{s}_i^l$ *and* $\mathbf{s}_j^{l'}$ *of* $\mathbf{s} \in \mathbb{R}^n$ *and with* $i < j$ *overlap if:* $j < i + l$*.*

Finally, two types of query are possible:

- **K-NN query:** Searching by recursion for the K nearest non-overlapping subsequences, i.e, the i^{th} selected subsequence has the smallest distance to the query and does not overlap with the $i-1$ previously selected subsequences. Algorithm 1 describes the K-NN query from a distance profile.

- **ϵ-range query:** Searching by recursion for the non-overlapping subsequences with the smallest distance to the query and stops whenever the distance is over a threshold $\epsilon > 0$.

The computation bottleneck. A brute force approach for K-NN distance profiling requires a computation time of $\mathcal{O}(Cn + Kn)$ where C is the computation time of the distance between two subsequences. Potentially limiting for long time series, more efficient algorithms have been proposed for several distances by taking advantage of the time series structure and some properties of the distances [ZM24; Yeh+16]. Specifically, it has been done for both elastic and lock-step distances but with different strategies.

Algorithm 1 NNQuery

Require: D a distance profile, K the number of similarities, l the subsequence length
1: $count \leftarrow 0$, $query \leftarrow ()$
2: **while** $count < K$ & $any(D) < +\infty$ **do**
3: $\quad i \leftarrow \text{argmin}(D)$
4: $\quad d \leftarrow \min(D)$
5: $\quad D[\max(i-l+1, 0), \min(i+l-1, n)] \leftarrow +\infty$
6: $\quad query.append((i, d))$
7: $\quad count = count + 1$
8: **return** *query*

Elastic distance profiling. While better suited for many applications, elastic distances are slow compared to the Euclidean distance. To avoid such computational burden, the algorithms take advantage of pruning and early stopping strategies to compute as few exact elastic distance values as possible. Regarding the DTW distance, one of the first works [Rak+12b] makes use of a DTW lower bound [KR05] for pruning or stopping computations. In fact, all pruning and early stopping strategies are derived from Euclidean-like lower bounds of elastic distances, and the efficiency of a strategy depends on the lower bound tightness. Starting with the DTW, several lower bounds have been proposed for different distances over the years [WP21; TPW19; LR13; KPC01]. They have been compared in a recent study [Pap+23] where they also propose a generic framework to define a lower bound of an elastic distance. Conjointly, in [HW21], they propose a general framework to derive pruning/early stopping strategies from lower bounds.

Lock-step distance profiling. Regarding lock-step distances, few algorithms have been proposed, and they are only concerned with the Z-normalized Euclidean distance [ZM24] or one of its variants that better deals with the noise [DAV19]. In all cases, the algorithms first compute the whole distance profile before performing a K-NN or ϵ-range query. To compute the distance profile, they take advantage of the memory to temporarily store some coefficients that are fast to compute, and they also make use of the Fast Fourier Transform (FFT) to compute the dot products between the query and all subsequences. Algorithm 2 describes the rolling dot product computation between a query and all subsequences using the FFT. Remarkably, the time complexity for computing the distance profile does not depend on the length of the query; it is in $\mathcal{O}(n \log(n))$ where n is the length of the time series.

Computing the Z-normalized Euclidean distance profile. It is worth taking a closer look at the MASS algorithm, which computes the Z-normalized Euclidean distance profile [ZM24]. Indeed, its computational tricks are the backbones of many algorithms searching for efficiency.

As a reminder, the Z-normalized Euclidean distance between $x \in \mathbb{R}^l$ and $y \in \mathbb{R}^l$ is defined by:

$$d_Z(\mathbf{x}, \mathbf{y}) = \left\| \frac{\mathbf{x} - \mu_x \mathbf{1}}{\sigma_x} - \frac{\mathbf{y} - \mu_y \mathbf{1}}{\sigma_y} \right\|$$

with $\mu_x = l^{-1} \sum_{i=1}^{l} x_i$, $\sigma_x^2 = l^{-1} \sum_{i=1}^{l} (x_i - \mu_x)^2$, and $\mathbf{1} = (1, \ldots, 1) \in \mathbb{R}^l$. Interestingly, the distance can be formulated such that it becomes better suited for efficient computation of the distance profile:

$$d_Z(\mathbf{x}, \mathbf{y}) = \sqrt{2 \left(l - \frac{\langle \mathbf{x}, \mathbf{y} \rangle - l \mu_x \mu_y}{\sigma_x \sigma_y} \right)} \tag{2.2}$$

Indeed, means and standard deviations of subsequences can be computed in linear time using cumulative sum. Additionally, all dot products between the query and the subsequences can be computed by convolving the query along the time series, which can be efficiently done using the FFT. The MASS algorithm is clearly described in [ZM24] and Algorithm 3 provides a pseudo-code of the algorithm.

Algorithm 2 FFTRollingDotProduct

Require: T a time series, Q a query time series
1: $n \leftarrow length(T),\ l \leftarrow length(Q)$
2: $Q \leftarrow reverse(Q)$
3: $Q[l+1:n] \leftarrow 0$ ▷ *Pad with zeros, assuming $l < n$*
4: $iT \leftarrow FFT(T),\ iQ \leftarrow FFT(Q)$
5: $TQ \leftarrow iFFT(iT * iQ)$
6: **return** $TQ[l:n]$

Algorithm 3 Z-NormalizedEuclideanDistanceProfile

Require: T a time series, Q a query time series
1: $n \leftarrow length(T),\ l \leftarrow length(Q)$
2: $Q \leftarrow Znorm(Q)$ ▷ *Z-normalization of the query*
3: $TQ \leftarrow FFTRollingDotProduct(T, Q)$
4: $\sigma \leftarrow RollingSTD(T, l)$ ▷ *see [ZM24]*
5: $D \leftarrow \sqrt{2(l - TQ/\sigma)}$
6: **return** D

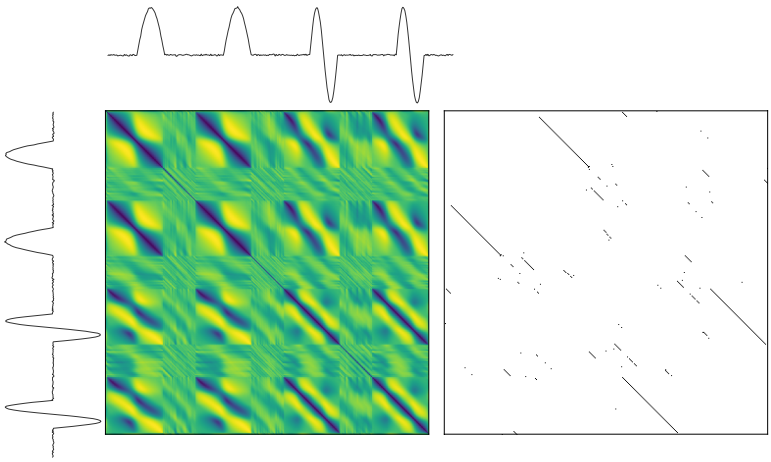


Figure 2.1 Left: Cross-distance matrix between subsequences of the displayed time series in the case of the Z-normalized Euclidean distance. The subsequence length is equal to the pattern length. **Right:** The matrix profile: it is the graph of the nearest non-overlapping neighbor. The continuous straight lines indicate the location of the repeated patterns.

2.1.3 The Matrix Profile

From one to many queries. Until now, the presented algorithms were all designed to perform a single query simultaneously. However, for some unsupervised tasks like motif discovery or anomaly detection, algorithms must perform many queries on the same time series to identify subsequences of interest. In fact, such algorithms end up computing the cross-distance matrix between all subsequences or an approximation of it with its K-NN graph, i.e., the graph that connects any subsequence to its K nearest non-overlapping

neighbors, see Figure 2.1. When considering the Z-normalized Euclidean distance, a straightforward application of distance profiling would lead to the construction of the K-NN graph in a time complexity of $\mathcal{O}(n^2 \log(n) + Kn^2)$. Potentially limiting for long time series, more efficient algorithms have been proposed, and they can reduce the time complexity up to $\mathcal{O}(Kn^2)$. In the literature, such algorithms are referred to as Matrix Profile [Yeh+16], and they have been applied to various tasks like recovering a song from a cover [Sil+18], detecting abnormalities in internet communication protocols [Sco+24] or discovering meaningful patterns in ECGs toward better diseases' diagnosis [WJ21].

Related work. The original work [Yeh+16] defines the matrix profile as the 1-NN graph of a time series weighted by the distances. Formally, given a time series $\mathbf{s} \in \mathbb{R}^n$, a window length $l > 0$ and a distance measure d, the matrix profile is the sequence:

$$\left(\mathbf{s}_i^l, \mathbf{n}_i, d_i\right)_{i \in [\![1, n-l+1]\!]} \tag{2.3}$$

where $\mathbf{n}_i$ is the nearest non-overlapping neighbors of the subsequence $\mathbf{s}_i^l$ and d_i the distance between both subsequences. While the matrix profile is defined for any distance measure, most algorithms focus on the Z-normalized Euclidean distance to benefit from its fast computation.

The first proposed algorithm, called STAMP [Yeh+16], has a greedy approach by looping over all distance profiling, leading to a time complexity of $\mathcal{O}(n^2 \log(n))$. Soon after, a second algorithm, called STOMP [Zhu+16], improves over STAMP by taking advantage of dynamic programming and a recursive formulation of the Z-normalized Euclidean distance, dropping the time complexity to $\mathcal{O}(n^2)$. Compared to STAMP, STOMP also have the advantage of being computable with GPUs [Zhu+16]. Both STAMP and STOMP are offline algorithms and they can be limiting for long time series. This issue has been leveraged with SCRIMP++ [Zhu+18], an anytime algorithm that has a time complexity of $\mathcal{O}(n^2 \log(n))$ in the worst-case scenario by combining both offline approaches.

These three algorithms are the foundational work around the matrix profile, and several variations have been proposed depending on the context. For instance, VALMOD [Lin+18] computes the matrix profile for a range of window lengths. mSTAMP [YKK17] computes the matrix profile of multidimensional time series. SWAMP [AKK20] approximates the matrix profile with the DTW by taking advantage of its lower bound [KR05]. Additionally, an extension of the matrix profile to the case of the K-NN graph has been proposed in [MAM23]. This extension is further described in the next paragraph as it is particularly interesting for motif discovery.

Computation of the K-NN graph with STOMP algortihm. Taking on STOMP, the algorithm loops over the subsequences of a time series, and at each step, it computes the distance profile between the time series and the subsequence before performing a K-NN query. The algorithm's efficiency comes from the recursive formulation of the dot product between subsequences. Indeed, given a time series $\mathbf{s} = (s_1, \ldots, s_n) \in \mathbb{R}^n$ and a window length $l > 0$, $\langle \mathbf{s}_{i+1}^l, \mathbf{s}_{j+1}^l \rangle$ can be computed in $\mathcal{O}(1)$ from $\langle \mathbf{s}_i^l, \mathbf{s}_j^l \rangle$ with the recursion:

$$\langle \mathbf{s}_{i+1}^l, \mathbf{s}_{j+1}^l \rangle = \langle \mathbf{s}_i^l, \mathbf{s}_j^l \rangle + s_{i+1}s_{j+1} - s_i s_j \quad \forall (i,j) \in [\![1, n-l+1]\!]^2 \tag{2.4}$$

In order to take advantage of the distance recursion formula (eq. (2.2)), the algorithm first computes the mean and standard deviation of all subsequences as well as the inner products between the first subsequence and all the others. Then, the distance profiles can be computed successively in $\mathcal{O}(n)$ thanks to the recursive property of the inner product (eq. (2.4)). A K-NN query is performed at each iteration, leading to a time complexity of $\mathcal{O}(Kn^2)$. Algorithm 4 describes the computation of the K-NN graph considering the Z-normalized Euclidean distance.

Algorithm 4 Compute K-NN Graph (Z-normalized Euclidean distance)

Require: S a time series, l the subsequence length, K the number of neighbor

$n \leftarrow Length(S),\ Graph \leftarrow ()$
$\mu, \sigma \leftarrow AdditionalCoefficients(S, l)$
$I_1 \leftarrow InnerProductFFT(S_1^l, S)$
$I \leftarrow I_1$
$D \leftarrow DistanceProfile(I, \mu, \sigma)$
$D[1:l] \leftarrow +\infty$
$Graph.add\_neighbors(1, NNQuery(D, K, l))$ ▷ *See Algorithm 1*
for i=2,. . . ,n-l+1 **do**
    for j=n-l,. . . ,1 **do**
        $I[j+1] = I[j] + S[i+w]S[j+w] - S[j]S[i]$
    $I[1] \leftarrow I_1[i]$
    $D \leftarrow DistanceProfile(I, \mu, \sigma)$
    $D[\max(i-l+1, 0), \min(i+l-1, n)] \leftarrow +\infty$
    $Graph.add\_neighbors(i, NNQuery(D, K, l))$ ▷ *See Algorithm 1*
return *Graph*

Conclusion. A large family of algorithms has been built around the Z-normalized Euclidean distance in similarity search. Thanks to their simplicity and efficiency, they have been applied in numerous contexts with great success. Going further, the following section shows that these algorithms are not restricted to this single distance but can be extended to a large family of distances invariant to more complex deformations.

2.2 On distances invariant to rigid deformations.

2.2.1 A limitation of the Z-normalized Euclidean distance

A trend sensitive distance. One concern for tasks like motif discovery is the trend sensitivity of the Z-normalized Euclidean distance. Indeed, in several applications, the patterns of interest do not depend on the trend, yet its presence modifies the shape of the patterns. For instance, as depicted in Figure 2.2, the pattern of interest in an ECG is the heartbeat cycle, but due to the subject's movements, a trend appears and deforms the shape of heartbeats. When performing a similarity search from one heartbeat, some occurrences are missed. Similarity search under the Z-normalized Euclidean distance is less reliable in the presence of a trend.

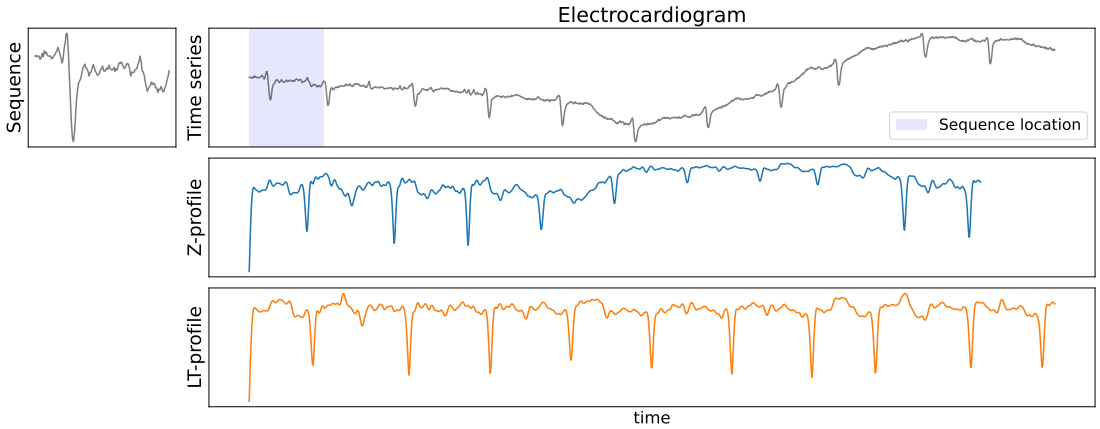


Figure 2.2 Similarity search performed on an electrocardiogram. The top left figure is the query subsequence. The top right figure is the time series with the query subsequence location in blue. The middle is the Z-normalized distance profile. Due to the trend, some occurrences of the query subsequences are missed with the Z-normalized distance profile. The bottom is also a distance profile where, for each subsequence, the linear trend is removed before applying the Z-normalized Euclidean distance. Here, all heartbeats are identifiable.

A closer look. Assuming that the trend is linear at the scale of the subsequences, we compare, with the Z-normalized Euclidean distance, a sequence $\mathbf{x} \in \mathbb{R}^l$ and one of its linearly deformed version: $\mathbf{x} + a\mathbf{t}$. Here, $a > 0$, $\mathbf{t} = (0, \ldots, l-1)$ and there is no offset as the distance is invariant to offset shift. The Z-normalized Euclidean distance verifies:

$$d_Z^2(\mathbf{x}, \mathbf{x} + a\mathbf{t}) = 2l\left(1 - \frac{1 + (\frac{a\sigma_t}{\sigma_x})\rho_{x,t}}{\sqrt{1 + 2(\frac{a\sigma_t}{\sigma_x})\rho_{x,t} + (\frac{a\sigma_t}{\sigma_x})^2}}\right) \tag{2.5}$$

where $\rho_{x,t} = cov(\mathbf{x}, \mathbf{t})/(\sigma_x\sigma_t)$ is the Pearson correlation between $\mathbf{x}$ and $\mathbf{t}$.

Here, the distance depends on two parameters: the variance ratio $(a\sigma_t)^2/\sigma_x^2$ and the Pearson correlation between $\mathbf{x}$ and $\mathbf{t}$, namely $\rho_{x,t}$. The distance is null when the sequence $\mathbf{x}$ is linear, i.e., $\rho_{x,t} = 1$. On the other hand, the distance increases and converges to the limit $\sqrt{2l(1-\rho_{x,t})}$ as the ratio $(a\sigma_t)^2/\sigma_x^2$ converges to infinity. In fact, the ability of the Z-normalized Euclidean distance to detect similarities depends on the nature of the sequence. When the sequence is close to being linear ($\rho_{x,t} \to 1$), the distance remains low regardless of the variance ratio. On the contrary, when the sequence is far from being linear ($\rho_{x,t} \to 0$), the similarity can only be detected if the variance ratio is low, meaning there is almost no linear trend. Figure 2.3-left illustrates the experiment, and Figure 2.3-right illustrates the Z-normalized Euclidean distance between a sequence and its linearly deformed version as a function of the variance ratio. The curves' steepness at low variance ratios indicates that the Z-normalized Euclidean distance is not robust to linear trend deformations.

Potential solutions. A first solution could be to remove the trend before searching for patterns. Unfortunately, state-of-the-art algorithms for detrending do not achieve perfect

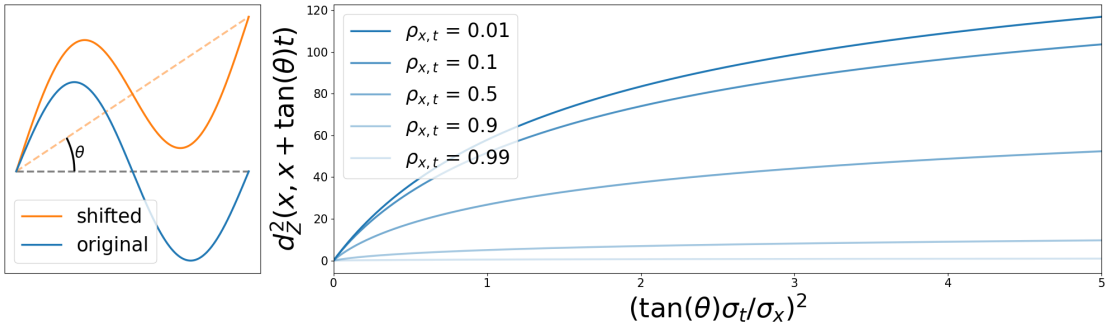


Figure 2.3 Left: Experiment illustrations, the original signal is linearly shifted with an angle θ. Right: Z-normalized Euclidean distance value as a function of the ratio between the linear shift variance and the original sequence variance for different values of the Pearson correlation between $\mathbf{x}$ and $\mathbf{t}$.

results, are time-consuming, or require fine-tuning [Sin+20; Hip+19].

Another solution could be to perform a similarity search with a distance that is invariant to the effect of a trend at the scale of the subsequences. Going back to the example of the trended ECG Figure 2.2, the last row displays a distance profile where the linear trend in the query and all subsequences has been removed before comparing them with the Z-normalized Euclidean distance. Remarkably, all heartbeat occurrences are well identified with this distance profile. This observation motivates the work toward distances invariant to more complex deformations than the amplitude scaling and the offset shift. Additionally, such distances should inherit the recursive property of the Z-normalized Euclidean distance for efficient computation of distance and matrix profiles.

2.2.2 Simplifications of the framework for time series shape analysis

As depicted in Section 1.2.2, comparing time series by their shape requires a proper definition of the action of a group of deformations on a set of time series. In this section, we refine the general group action, eq. (1.9), to subsets of rigid deformations, and we derive distances invariant to such deformations.

A fixed time interval. The efficiency of the distance and matrix profile algorithms under the Z-normalized Euclidean distance is built upon the fast computation of the dot product between subsequences. Such efficiency is achieved by taking advantage of the Fast Fourier Transform or the recursive formula in Equation (2.4). In both cases, the computation relies on the one-to-one Euclidean pairing between subsequences' samples. Such pairing imposes subsequences to be defined on the same time interval for regularly sampled time series. In order to perform shape comparison between subsequences, the set of admissible time series, eq. (1.8), must be reduced to $\mathsf{L}^2(\mathsf{I}, \mathbb{R}^d, \mu)$, i.e., the set of functions defined on the closed interval $\mathsf{I} \subset \mathbb{R}$ taking value in $\mathbb{R}^d$ and that are square integrable for the Borel measure μ. In addition, $\mathsf{L}^2(\mathsf{I}, \mathbb{R}^d, \mu)$ combined with the inner product:

$$\langle f, g \rangle_L = \int_{\mathsf{I}} \langle f(t), g(t) \rangle d\mu(t) \tag{2.6}$$

is a Hilbert space. Note that in comparison to the admissible set of time series, the definition interval is dropped off as it is the same for all time series in this set. Also, the group of deformations does not need to be defined on the ambient space. It is sufficient to define the deformations on the interval I.

No time parametrization invariance. Continuing to focus on efficient distances to compare subsequences, deformations induced by temporal parametrization are not taken into account. Indeed, the invariance to such deformations requires a time-consuming optimization. As a drawback, the distances become sensitive to time warping. More precisely, following the work [AM14; Mal12], the distances' sensitivity depends on the length of the displacement induced by the time parametrization.

Action of a finite dimension vector subspace. Going back to the example of the Z-normalized Euclidean distance, its invariance to offset shifts can be understood as the invariance to the set of constant functions $\{t \in \mathsf{I} \mapsto c \mid c \in \mathbb{R}^d\}$. As well, the invariance to linear trend can be understood as the invariance to the set of functions $\{t \in \mathsf{I} \mapsto at + b \mid (a, b) \in \mathbb{R}^d \times \mathbb{R}^d\}$. In both cases, the group of deformations is, in fact, a vectorial subspace of $\mathsf{L}^2(\mathsf{I}, \mathbb{R}^d, \mu)$.

Going in that direction, we suppose that we wish to be invariant to a finite dimensional vector subspace H of $\mathsf{L}^2(\mathsf{I}, \mathbb{R}^d, \mu)$ and the amplitude scaling deformations. Formally, we wish to be invariant to the group $\mathbb{R}_+^* \ltimes \mathsf{H}$ with the composition rule $(\lambda_2, h_2) \times (\lambda_1, h_1) = (\lambda_2\lambda_1, h_2 + \lambda_2 h_1)$ and which acts on the left on $\mathsf{L}^2(\mathsf{I}, \mathbb{R}^d, \mu)$ by the action:

$$(\lambda, h) \cdot f = \lambda f + h \tag{2.7}$$

Note that the action is not transitive as H is a finite dimensional subspace of an infinite dimension space. In other words, the quotient space, $\mathsf{L}^2(\mathsf{I}, \mathbb{R}^d, \mu)/\mathbb{R}_+^* \ltimes \mathsf{H}$ is not reduced to a single orbit. Therefore, it makes sense to search for a metric between orbits, meaning that the metric is invariant to the action of the group $\mathbb{R}_+^* \ltimes \mathsf{H}$, i.e., with some abuse of notation the metric verifies: $d(\lambda f + h, \lambda' g + h') = d(f, g)$.

2.2.3 Construction of distances invariant to rigid deformations

This section aims to construct efficient distances on time series invariant to deformations whose action is governed by Equation (2.7). In a general sense, we are interested in defining a distance on the quotient space of the non-transitive action:

$$((\lambda, h), m) \in (\mathbb{R}_+^* \ltimes \mathsf{H}) \times \mathsf{M} \mapsto \lambda m + h \in \mathsf{M} \tag{2.8}$$

where M is a Hilbert space, and H one of its finite dimensional vector subspace. Following the strategy presented in Section 1.2.2, we could construct such a metric by defining an equivariant Riemannian metric on M and find an explicit formulation of the geodesic distance. An illustration of this strategy can be found in Chapter 12 of [You10]. However, we can employ a simpler strategy that relies on the vector structure of the group action. Going in that direction, we first present a way to define invariant distances from invariant embeddings. We then show that such embedding exists for the action of H on M, which

can be extended to the action of $\mathbb{R}_+^* \ltimes \mathsf{H}$. We conclude by showing that these distances inherit from the same recursive property as the Z-normalized Euclidean distance, making them relevant for similarity search.

Invariant distances based on embedding. Let us consider a generic left action $(g, m) \in \mathsf{G} \times \mathsf{M} \mapsto g \cdot m \in \mathsf{M}$ that is not transitive and an embedding map $L : \mathsf{M} \mapsto \mathsf{N}$. The following definition and proposition detail the sufficient conditions on the embedding to define a distance on the quotient space M/G.

Definition 4 (Invariant & orbit-injective embedding)**.** *An embedding map $L : \mathsf{M} \mapsto \mathsf{N}$ is said to be G-invariant, if for any $(g, m) \in \mathsf{G} \times \mathsf{M}$, $L(g \cdot m) = L(m)$. Additionally, L is said to be orbit-injective if the application $\tilde{L} : [m] \in \mathsf{M}/\mathsf{G} \mapsto L(m) \in \mathsf{N}$ is injective.*

Proposition 1. *Suppose an embedding map $L : \mathsf{M} \mapsto \mathsf{N}$ that is G-invariant and orbit-injective. If (N, d) is a metric space, then the application:*

$$\tilde{d} : ([m], [m']) \in \mathsf{M}/\mathsf{G} \times \mathsf{M}/\mathsf{G} \mapsto d(L(m), L(m')) \in \mathbb{R}_+$$

is a metric on the quotient space M/G.

Proof. The symmetry and the triangular inequality of $\tilde{d}$ are inherited from d. The separation of $\tilde{d}$ comes from the separation d and the orbit-injectivity of L. □

Note that in the previous proposition and definition, we did not enforce the set M to be a Hilbert space. However, this assumption holds for the remainder of this section.

Embedding invariant to a vector subspace. For now, we disregard the action of amplitude scaling, and we solely focus on the action of the finite dimensional subspace H of M by the usual vector addition: $(h, m) \in \mathsf{H} \times \mathsf{M} \mapsto m + h \in \mathsf{M}$. The following proposition exhibits a H-invariant embedding that is also orbit-injective.

Proposition 2. *Let P_{H} be the orthogonal projector on H, and I_d be the identity map on M, the embedding, $L = I_d - P_{\mathsf{H}}$ (the projector on $\mathsf{H}^\perp$) is H-invariant and orbit-injective.*

Proof. **Existence of L**: As H is a finite dimension vector space, it is a closed and convex subset of the Hilbert space M; the orthogonal projector on H, denoted P_{H}, exists. Therefore, $L : \ m \in \mathsf{M} \mapsto m - P_{\mathsf{H}}(m) \in \mathsf{H}$ is well defined.
H-invariance of L: Since H is closed, $\mathsf{M} = \mathsf{H} \oplus \mathsf{H}^\perp$, and for any $x \in \mathsf{M}$, we decompose $m = m_{\mathsf{H}} + f_{\mathsf{H}^\perp}$. Thus, for any $m \in \mathsf{M}$, and $h \in \mathsf{H}$:

$$\begin{aligned} L(m+h) &= m + h - P_{\mathsf{H}}(m+h) \\ &= m + h - P_{\mathsf{H}}(m_{\mathsf{H}^\perp} + m_{\mathsf{H}} + h) \\ &= m + h - (m_{\mathsf{H}} + h) \quad \textbf{(projector on a closed vectorial subspace)} \\ &= m - m_{\mathsf{H}} \\ &= L(m) \end{aligned}$$

which proves the H-invariance of L.

Orbit-injectivity of L: For any $m \in \mathsf{M}$, its orbits corresponds to:

$$\begin{aligned}[m] &= \{m + h \mid h \in \mathsf{H}\} \\ &= \{L(m) + h' \mid h \in \mathsf{H},\ h' = P_{\mathsf{H}}(m) + h \in \mathsf{M}\} \\ &= L(m) + \mathsf{H}\end{aligned}$$

Therefore, for any $([m], [m']) \in \mathsf{M}/\mathsf{H} \times \mathsf{M}/\mathsf{H}$, such that $[m] \cap [m'] = \emptyset$ implies that $L(m) \neq L(m')$ proving the orbit-injectivity of L. □

Remark 2. *Some properties are note worthy:*

- *If $(h_i)_{i\in[\![1,N]\!]}$ is an orthonormal basis of the finite dimensional vector subspace H, then the orthogonal projector on H as an explicit formulation:*

$$P_{\mathsf{H}} : m \in \mathsf{M} \mapsto \sum_{i=1}^{N} \langle m, h_i \rangle h_i \in \mathsf{H} \tag{2.9}$$

- *The H-invariant embedding map L is a linear and bounded operator with $Ker(L) = \mathsf{H}$.*

Including invariance to amplitude scaling. Fortunately, we can easily define an embedding invariant to the action of rigid deformations, eq. (2.8), from an embedding invariant to the action of the vector subspace:

Proposition 3. *Let $L : \mathsf{M} \mapsto \mathsf{M}$ be the H-invariant and orbit-injective embedding map induced by the orthogonal projector on H as defined in proposition 2. The embedding map:*

$$\hat{L} :\ m \in \mathsf{M} \mapsto \begin{cases} L(m)/\|L(m)\|_{\mathsf{M}} & \text{if } m \in \mathsf{M}\backslash\mathsf{H} \\ 0_{\mathsf{M}} & \text{else} \end{cases}, \tag{2.10}$$

is $(\mathbb{R}_+^* \ltimes \mathsf{H})$-invariant and orbit-injective.

Proof. $(\mathbb{R}_+^* \ltimes \mathsf{H})$-invariance is due to the linearity and H-invariance of L, and the orbit-injectivity in induced by the linearity and orbit-injectivity of L. □

Invariant metric and fast computing. Thanks to Propositions 3 and 1, we can define a metric invariant to rigid deformations by the application:

$$\tilde{d} :\ ([m], [m']) \in \mathsf{M}/(\mathbb{R}_+^* \ltimes \mathsf{H}) \times \mathsf{M}/(\mathbb{R}_+^* \ltimes \mathsf{H}) \mapsto \|\hat{L}(m) - \hat{L}(m')\|_{\mathsf{M}} \in \mathbb{R}_+. \tag{2.11}$$

Formally, it is a metric on the quotient space $\mathsf{M}/(\mathbb{R}_+^* \ltimes \mathsf{H})$, and if $(h_i)_{i\in[\![1,N]\!]}$ is an orthonormal basis of H, then for any $(m, m') \in \mathsf{M}\backslash\mathsf{H} \times \mathsf{M}\backslash\mathsf{H}$:

$$\tilde{d}([m], [m']) = \sqrt{2\left(1 - \frac{\langle m, m' \rangle - \sum_{i=1}^{N} \langle m, h_i \rangle \langle m', h_i \rangle}{\|L(m)\| \|L(m)\|}\right)} \tag{2.12}$$

with $\|L(m)\| = \sqrt{\langle m, m \rangle - \sum_{i=1}^{N} \langle m, h_i \rangle^2}$.

The following section tailors this framework to the case of time series and presents an example of distance in the discrete case.

2.2.4 Back to time series

Summary. We started by observing that the Z-normalized Euclidean distance is sensitive to the deformations induced by linear trends, making the similarity search less reliable when such deformations are present. However, this distance remains widely used due to its fast computation. To bridge the gap, we have investigated the properties of the Z-normalized Euclidean distance that make up its efficiency, and we have presented a framework to create distances that are invariant to custom groups of deformations while respecting the efficiency properties.

The case of time series. The group action that we considered for the time series is defined as follows:

$$((\lambda, h), f) \in (\mathbb{R}_+^* \ltimes \mathsf{H}) \times \mathsf{L}^2(\mathsf{I}, \mathbb{R}^d, \mu) \mapsto \lambda f + h \in \mathsf{L}^2(\mathsf{I}, \mathbb{R}^d, \mu) \tag{2.13}$$

where H is finite dimensional subspace of $\mathsf{L}^2(\mathsf{I}, \mathbb{R}^d, \mu)$. Essentially, we assume that the signals belong to a Hilbert space in which they can be decomposed on a functional basis where a finite dimensional subspace is, in fact, the action of the deformations. Down the line, the customization of the distance depends on the choice of basis for the deformation subspace.

From continuous to discrete. Thanks to the measure theory, the transition from continuous to discrete time series has been made easy using dirac measures. Indeed, the set $\mathsf{L}^2(\mathsf{I}, \mathbb{R}^d, \mu)$ has been defined with an arbitrary Borel measure μ. Therefore, by taking the discrete measure $\mu = \sum_{i=1}^{l} \delta_{t_i}$, where $(t_i)_{i \in [\![1,l]\!]} \subset \mathsf{I}$ corresponds to the sampling of the interval I, we have the following equalities:

$$\langle f, g \rangle_L = \int_{\mathsf{I}} \langle f(t), g(t) \rangle d\mu(t) = \sum_{i=1}^{l} \langle f(t_i), g(t_i) \rangle \tag{2.14}$$

for any f and g in $\mathsf{L}^2(\mathsf{I}, \mathbb{R}^d, \mu)$. It allows to work with continuous signals while performing computation with their discretized versions.

More importantly, up to some continuity considerations, the converge in law of $\frac{1}{n} \sum_{i=1}^{n} \delta_{\frac{i}{n}}$ toward the Lesbegue measure on $[0, 1]$ ensures the weak convergence of the discretized distance value to its continuous counterpart as the sampling gets refined.

Univariate Z-normalized Euclidean distrance. As an introductory example, we retrieve, with the framework, the Z-normalized Euclidean distance in the univariate case. The set of functions that we consider is $\mathsf{L}^2([0, l], \mathbb{R}, \lambda)$ where λ is the Lebesgue measure on $[0, l]$ and $l \in \mathbb{N}^*$. The distance is invariant to offset shift, which is the subspace of constant functions, and it is generated by the unit norm function $e : \ t \in [0, l] \mapsto 1/\sqrt{l} \in \mathbb{R}$.

According to Proposition 3 the invariant embedding of a non-constant function f is the function: $(f - \langle f, e \rangle_L e)/\|f - \langle f, e \rangle_L e\|_L$ which in the discrete case with the measure $\sum_{i=1}^{l} \delta_i$ leads to $i \in [\![1, l]\!] \mapsto (f(i) - \mu_f)/\sqrt{l}\sigma_f$ where $\mu_f = l^{-1} \sum_{i=1}^{l} f(i)$ and

$\sigma_f^2 = l^{-1}\sum_{i=1}^{l}(f(i)-\mu_f)^2$. Following Equation (2.11), the distance between the non-constant functions f and g is:

$$\tilde{d}(\mathbf{f},\mathbf{g}) = \left\| \frac{\mathbf{f}-\mu_f\mathbf{1}}{\sqrt{l}\sigma_f} - \frac{\mathbf{g}-\mu_g\mathbf{1}}{\sqrt{l}\sigma_g} \right\| = \frac{1}{\sqrt{l}} d_Z(\mathbf{f},\mathbf{g}) \ ,$$

where $\mathbf{f} = (f(1),\dots,f(l)) \in \mathbb{R}^l$ and $\mathbf{1} = (1,\dots,1) \in \mathbb{R}^l$. We have retrieved the Z-normalized Euclidean distance up to a constant factor.

Fast computation. An invariant distance defined as (2.12) inherits from the same fast computation properties as the Z-normalized Euclidean distance, see Equations (2.2) and (2.4). Algorithm 6 describes the computation of the distance profile with an invariant distance. Assuming that the deformation group H is a subspace of dimension K, its computation time is in $\mathcal{O}(Kn\log(n))$. Also, the computation of the K-NN graph with simple modifications of Algorithm 4 to include the coefficient computed with Algorithm 5, and its computation time remains $\mathcal{O}(n^2)$.

Algorithm 5 H-RollingInvariantNorm

Require: T a time series, l the subsequence length, $\mathcal{H}$ an orthonormal basis of H.

$n \leftarrow length(T), K \leftarrow length(\mathcal{H})$
$coefs \leftarrow \{\}$
$norm \leftarrow SelfDotProduct(T,l)$ ▷ *see [ZM24]*
for $k = 1,\cdots,K \in \mathsf{H}$ **do**
  $coefs[k] \leftarrow FFTRollingDotProduct(T, H[k])$ ▷ *see Algorithm 2*
  $norm \leftarrow norm - coefs[k]^2$
$norm \leftarrow \sqrt{norm}$
return $norm, coefs$

Algorithm 6 $(\mathbb{R}_+^* \ltimes \mathsf{H})$-InvariantDistanceProfile

Require: T a time series, Q a query time series, $\mathcal{H}$ an orthonormal basis of H.

$n \leftarrow length(T),\ l \leftarrow length(Q)$
$Q \leftarrow InvariantEmbedding(Q,\mathcal{H})$ ▷ *see* (2.10) *and* (2.9)
$TQ \leftarrow FFTRollingDotProduct(T,Q)$
$norm \leftarrow RollingInvariantNorm(T,l,\mathcal{H})$ ▷ *see Algorithm 5*
$D \leftarrow \sqrt{2(l - TQ/norm)}$
return D

Conclusion. We have presented a general framework to define distances invariant to a custom set of rigid deformations. We have detailed the framework in the case of time series, and we have shown that such distances can be combined with distance or matrix profile algorithms as they conserve the efficiency properties of the Z-normalized Euclidean distance. In the next section, we present and evaluate a distance that overcomes the

sensitivity of the Z-normalized Euclidean distance to the trend by considering the set of affine functions as deformations.

2.3 An illustration: the LT-normalized Euclidean distance

2.3.1 Construction of the distance

The group of deformations. As illustrated in Section 2.2.1, similarity search under the Z-normalized Euclidean distance is less reliable whenever the time series presents a trend. However, when it can be assumed that the trend is smooth and behaves linearly at the scale of subsequences, it would be interesting to have a distance that is invariant to the set of affine functions: $\{t \in [0, l] \mapsto at + b \mid (a, b) \in \mathbb{R}^d \times \mathbb{R}^d\}$. Fortunately, this is a vector space that is generated by the vectors $t \in [0, l] \mapsto e_j \in \mathbb{R}^d$ and $t \in [0, l] \mapsto te_j \in \mathbb{R}^d$ where $(e_j)_{j \in [\![1,d]\!]}$ is the orthonormal basis of $\mathbb{R}^d$.

The distance definition. Following the framework described in Section 2.2.3, the distance can be defined from the orthonormal basis of the deformations, which can be obtained from the generating vectors by the Gram-Schmidt process. In the case of discrete univariate time series, the LT-normalized Euclidean distance has the following definition:

Definition 5 (LT-normalized Euclidean distance)**.** *The LT-normalized Euclidean distance between non-affine sequences* $\mathbf{x} \in \mathbb{R}^l$ *and* $\mathbf{y} \in \mathbb{R}^l$ *is:*

$$d_{LT}(\mathbf{x}, \mathbf{y}) = \left\| \frac{\mathbf{x} - (\alpha_x \mathbf{t} + \beta_x \mathbf{1})}{\|\mathbf{x} - (\alpha_x \mathbf{t} + \beta_x \mathbf{1})\|} - \frac{\mathbf{y} - (\alpha_y \mathbf{t} + \beta_y \mathbf{1})}{\|\mathbf{y} - (\alpha_y \mathbf{t} + \beta_y \mathbf{1})\|} \right\| \tag{2.15}$$

with $\mathbf{t} = (1, \ldots, l)$*,* $\alpha_x = cov(\mathbf{x}, \mathbf{t})/\sigma_t^2$*, and* $\beta_x = \mu_x - \alpha_x \mu_t$*. Additionally,* $cov(\mathbf{x}, \mathbf{t}) = l^{-1}\langle \mathbf{x}, \mathbf{t} \rangle - \mu_x \mu_t$*,* $\mu_t = (l + 1)/2$ *and* $\sigma_t^2 = (l^2 - 1)/12$*.*

Note that the invariant embedding of $\mathbf{x}$ removes the affine sequence that best fit $\mathbf{x}$ for the L^2-norm as the coefficient (α_x, β_x) of the linear regression problem:

$$\text{argmin}_{(a,b) \in \mathbb{R}^2} \|\mathbf{x} - (a\mathbf{t} + b\mathbf{1})\|^2$$

The following equation details the recursive formulation of the LT-normalized Euclidean distance between $\mathbf{x} \in \mathbb{R}^l$ and $\mathbf{y} \in \mathbb{R}^l$:

$$d_{LT}(\mathbf{x}, \mathbf{y}) = \sqrt{2 \left(1 - \frac{\langle \mathbf{x}, \mathbf{y} \rangle - l(\mu_x \mu_y + \alpha_x \alpha_y \sigma_t^2)}{\eta_x \eta_y}\right)} \tag{2.16}$$

where $\eta_x = \|\mathbf{x} - (\alpha_x \mathbf{t} + \beta_x \mathbf{1})\|$.

2.3.2 Experimental settings

We evaluated the performance of the LT-normalized Euclidean distance on three data mining tasks for time series:

- **Motif pair discovery:** Identifying the two most similar non-overlapping subsequences in a time series.
- **Similarity search:** Identifying all non-overlapping subsequences in a time series similar to a query subsequence.
- **Motif set discovery:** Identifying sets of subsequences encompassing every occurrence of distinct repeated patterns in a time series.

For reproducibility, the source code and all datasets are available on github[1]. In what follows, we present the datasets and the metrics to evaluate the tasks.

Datasets

We conducted our experimental evaluation on several labeled datasets constructed from real and synthetic time series. In this section, we succinctly describe the datasets, but a detailed description of all datasets can be found in Appendix A.1.

Real-world data. We have considered the following real-world univariate datasets:

(R-1) **mitdb-1**: ECGs from the The MIT-BIH Arrhythmia Database [Gol+00; MM01]. It contains 100 time series randomly selected from healthy patients such that they only contain normal heartbeats.

(R-2) **mitdb-2**: We randomly selected 100 ECGs from MIT-BIH. The number of repeated patterns varied between 1 and 4.

(R-4) **ptt-ppg**: Photoplethysmogram (PPGs) from the Pule-Transit-Time PPG dataset [Meh+22]. It contains 100 time series of a single pattern randomly selected from running subjects.

(R-6) **arm-coda**: Trajectories from the arm-coda datasets [Com+24]. It contains 64 time series of subjects performing various upper-limb movements.

Synthetic data. We have generated one dataset per data mining task with the following scenarios:

(S-1) **pair**: For pair motif discovery. There is 1 pattern of length 100 that repeats twice. The trend smoothness varies in order to estimate the distance robustness. The dataset contains 200 time series.

(S-2) **single**: For similarity search. There is 1 pattern of length 100 that repeats 50 times. The dataset contains 100 time series.

(S-3) **fixed**: There are 5 patterns of length 100. For each pattern, the number of occurrences is sampled uniformly between 2 and 10. The dataset contains 100 time series.

[1] `https://github.com/thibaut-germain/lt-normalized`

Performance metrics

We evaluate algorithms' performances for all three experiments with the precision, recall, and f1-score metrics for time series [Tat+18].

The computation of these metrics requires the additional step of pairing real and predicted local events. In the case of motif set discovery, it is a two-level assignment problem: predicted motif sets must be assigned to real motif sets, and predicted occurrences must be assigned to real ones between paired motif sets. The complexity of the assignment problem is lower in the case of similarity search and motif pair discovery, as there is only one motif set. Indeed, it becomes a single-level assignment problem: predicted occurrences must be assigned to predicted ones. In both cases, the optimal pairings maximize the total overlapping length between real and predicted events. These assignments can be efficiently computed with the Hungarian matching algorithm [Kuh55; Sar+21].

Additionally, the computation of the metrics relies on a threshold $\tau \in [0, 1]$ that controls the overlapping ratio paired occurrences. For precision (resp. recall), a motif occurrence is counted as a true positive if the ratio between the overlap length and the predicted (resp. real) occurrence length is greater than the threshold τ. Appendix A.2 provides clear definitions of the metrics in the case of motif set discovery.

2.3.3 Experimental results

In what follows, we present the experimental results for each task. In the last section, we evaluate the scalability of STOMP algorithm [Zhu+16] combined with the LT-normalized Euclidean distance when the length of the time series increases.

Motif pair discovery

Presentation of the experiment. In this experiment, we investigated the influence of the trend on the performance of LT-normalized and Z-normalized Euclidean distances for solving the best motif pair problem. This problem [Lin+18] consists of finding the pair of non-overlapping subsequences whose distance is minimal compared to all other non-overlapping subsequence pairs. The matrix profile provides an exact solution to this problem. Following the formalism of Equation (2.3), the solution corresponds to the edge of the nearest neighbor graph with minimal distance. We used this resolution scheme to compare the performance of the distances. Figure 2.4 illustrates the best motif pair problem and its resolution with the matrix profile. The top figure shows a time series of the pair dataset (S-1). The next figures show the LT-normalized and Z-normalized matrix profiles with the predicted best motif pair locations. The true motif pair was recovered with the LT-normalized Euclidean distance, while the Z-normalized Euclidean distance identified a pair of nearly linear subsequences corresponding to the trend.

To evaluate the influence of the trend on the best motif pair prediction, we considered the pair dataset (S-1), where time series have been generated with different values for the random walk variance. This parameter controls the trend's regularity: the regularity decreases as the variance increases. To measure the performance, we evaluate the accuracy score. A best motif pair prediction is counted as a true positive if, for each subsequence,

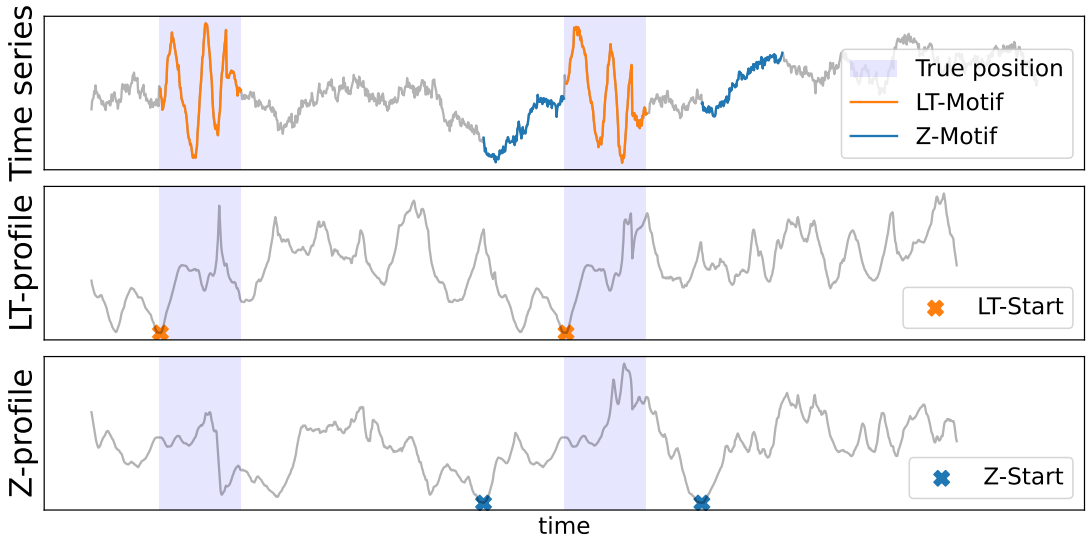


Figure 2.4 Top: Synthetic time series with a trend and one motif that occurs twice. True motif locations are highlighted in light purple. The predicted best motif pair is colored in orange for the LT-normalized Euclidean distance and blue for the Z-normalized Euclidean distance. **Middle:** Matrix profile with the LT-normalized Euclidean distance. The starting location of the predicted best motif pair is in orange. **Bottom:** Matrix profile with the Z-normalized Euclidean distance. The starting location of the predicted best motif pair is in blue.

the predicted location overlaps the real location by at least 50%. Figure 2.5 shows the accuracy scores of both distances as a function of the variance of the random walk.

Results. When the variance of the random walk is null, there is no trend, and the signal-to-noise ratio is equal to 22 dB on average. In this case, both distances are expected to predict the best motif pairs correctly, and indeed, both empirical scores are equal to one. However, the empirical results show that as soon as the variance of the random walk increases, the Z-normalized accuracy score decreases. On the other hand, the LT-normalized accuracy score remains consistently high for low random walk variances (between 0 and 0.2). Then, it decreases as the regularity of the trend decreases. Indeed, as the random walk variance increases, the trend is less likely to be linear at the scale of the motif.

Thanks to its invariance to linear trend, the LT-normalized Euclidean distance is more robust to the deformations induced by the trend for detecting the best motif pairs.

Similarity search

Experiment presentation. In this experiment, we compare the performance of LT-normalized and Z-normalized Euclidean distances on the distance profiling problem (Section 2.1.2).

We performed our experiment on datasets where the time series have one pattern that repeats multiple times: single (S-2), mitdb-1 (R-1), and ptt-ppg (R-4). Figure 2.6

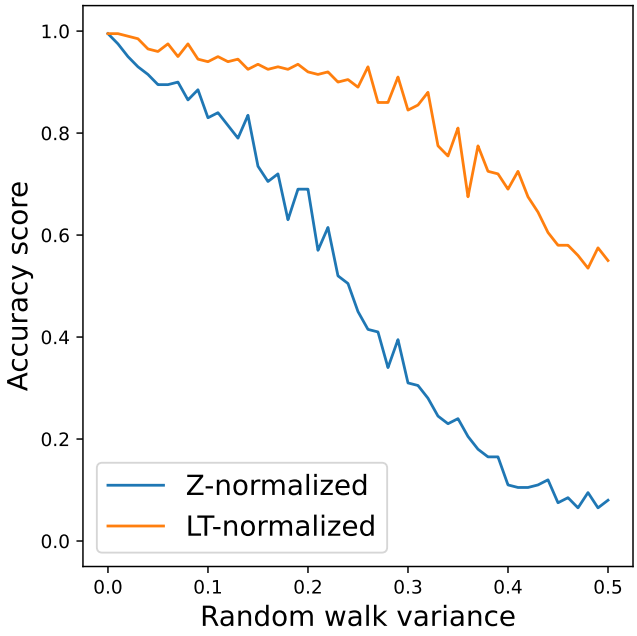


Figure 2.5 Accuracy scores for the LT-normalized and Z-normalized Euclidean distances as a function of the random walk variance.

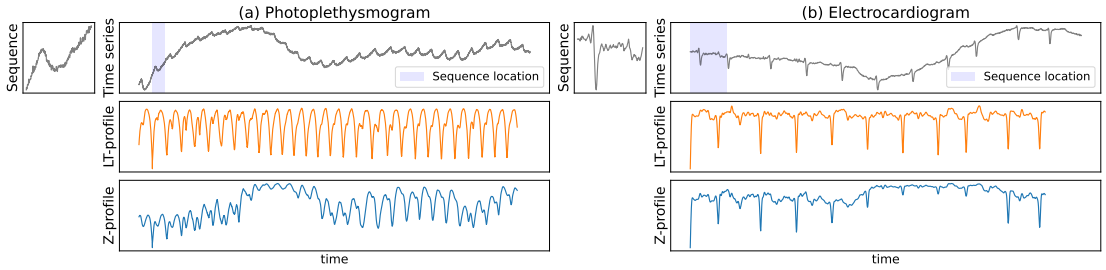


Figure 2.6 Similarity search on: **(a)** photoplethysmogram, and **(b)** electrocardiogram. In both cases, the top left is the query subsequence, the top right is the time series with the query subsequence location in blue, the middle is the LT-normalized Euclidean distance profile, and the bottom is the Z-normalized Euclidean distance profile. Due to the trend, some occurrences of the query subsequences are missed with the Z-normalized Euclidean distance profile. At the same time, they are all identifiable with the LT-normalized Euclidean distance profile.

illustrates the resolution of the similarity search problem on a PPG (a) and an ECG (b). In both cases, the top right plot shows the query sequence corresponding to the first occurrence of the repeated pattern. The top right plot shows the raw signal and the plots below show the LT-normalized and Z-normalized distance profiles. For both time series, the distance profiles are minimal at the starting locations of occurrences of the query sequences. However, the Z-normalized distance profile is sensitive to the trend, and the distance remains high for some occurrences. On the contrary, the trend less affects the LT-normalized distance profile, and the distance remains consistently low at the starting

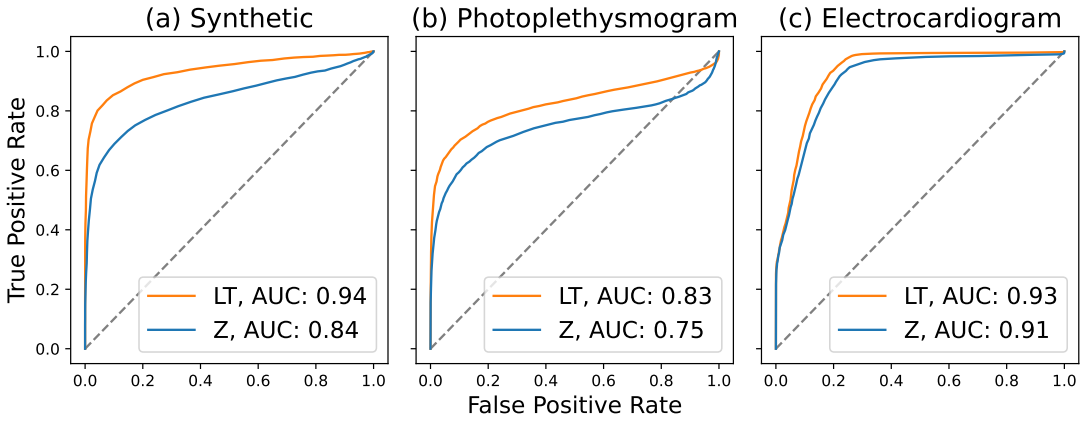


Figure 2.7 ROC curves of the similarity search problem for LT-normalized (orange) and Z-normalized (blue) distances on the datasets: **(a)** s-search, **(b)** ptt-ppg, and **(c)** mitdb-1. The LT-normalized Euclidean distance performs better than the Z-normalized Euclidean distance.

location of occurrences. The LT-normalized Euclidean distance is better suited for the similarity search on these two time series.

Results. We computed ROC curves for each distance and dataset according to the procedure described in [Pap+22b]. We counted a predicted occurrence as valid if it overlapped with a real occurrence by at least 75%. The results are shown in Figure 2.7. On average, the LT-normalized Euclidean distance outperformed the Z-normalized Euclidean distance as its AUC score is higher across all datasets. It is also worth noticing that the ROC curves of the LT-normalized Euclidean distance are consistently above those of the Z-normalized Euclidean distance. Indeed, the LT-normalized Euclidean distance is a generalization of the Z-normalized Euclidean distance to a broader class of deformations. As a result, the LT-normalized Euclidean distance profiles are more robust to the trend-induced deformations. Therefore, the number of true occurrences detected with the LT-normalized Euclidean distance is at least as good as that of the Z-normalized Euclidean distance in many cases.

Motif set discovery

Experiment presentation. In this experiment, we evaluated the performance of LT-normalized and Z-normalized Euclidean distances in solving the motif set discovery problem.

The motif set discovery problem [Lin+02] consists of identifying and clustering all occurrences of repeated patterns present in a time series. A heuristic based on the matrix profile exists to solve this problem [Zhu+16; Ben+20]. This heuristic can be extended to the LT-normalized Euclidean distance, and we used it to evaluate the performance of both distances. We also added two baselines, a matrix profile with the Euclidean distance and a second matrix profile with the Z-normalized Euclidean distance where time-series are preprocessed using a trend removal algorithm: A Seasonal-Trend Decomposition

Table 2.1 Motif Set Discovery. Euclidean (Euc), Z-normalized (Z), LT-normalized (LT), Trend removal & Z-normalized (STL+Z).

dataset	algorithm metric	Euc	LT	STL+Z	Z
single (S-2)	fscore	0.22	**0.83**	0.81	0.69
	precision	0.84	**1.00**	0.97	0.98
	recall	0.13	**0.72**	0.70	0.54
fixed (S-3)	fscore	0.34	**0.59**	0.57	0.56
	precision	0.33	**0.54**	0.52	0.51
	recall	0.41	**0.69**	0.66	0.66
mitdb1 (R-1)	fscore	0.40	**0.57**	0.55	0.53
	precision	**0.99**	0.98	0.98	0.97
	recall	0.26	**0.42**	0.40	0.39
mitdb2 (R-2)	fscore	0.14	**0.43**	0.43	0.39
	precision	0.65	**0.73**	0.72	0.72
	recall	0.10	**0.36**	0.33	0.32
ptt-ppg (R-4)	fscore	0.23	0.54	**0.62**	0.47
	precision	0.95	0.97	0.97	**0.98**
	recall	0.13	0.39	**0.47**	0.33
arm-coda (R-6)	fscore	0.14	**0.22**	0.19	0.19
	precision	0.15	**0.19**	0.18	0.18
	recall	0.22	**0.35**	0.31	0.33

Procedure Based on LOESS (STL) [Cle+90]. In terms of settings, the algorithm requires the number of sets to discover, which we assumed to be known, a subsequence similarity ratio, which we set to 3, and a subsequence length, which we set to be the average motif length for each dataset. The STL algorithm period is also set to the average motif length.

We ran the experiment on all datasets except the pair dataset. We evaluated the performance using the precision, recall, and f1-score metrics. We counted a pair of predicted/real occurrences as valid for the precision (resp. recall) metric if the length of their intersection is greater than 50% of the length of the predicted (resp. real) occurrence.

Results. Experimental results are shown in Table 2.1. On all datasets except ptt-ppg (R-4), the ranking based on the f1-score remains identical: (1) LT, (2) STL+Z, (3) Z, and (4) Euc. On ptt-ppg, STL+Z is first, and LT is second. STL+Z algorithm first removes the trend with the STL algorithm before applying the motif discovery algorithm with the Z-normalized. LT and STL+Z are the best performers, meaning that removing deformations induced by the trend is helpful for motif discovery on these datasets.

The main difference between the LT and the STL+Z algorithms is the management of the trend. The STL+Z removes the trend by leveraging a decomposition of the time series in season, trend, and remainder components. On the other hand, the LT algorithm only assumes that the trend is approximately linear at the scale of the motifs. LT performs better than STL+Z on most datasets. It validates the assumption that the trend is locally

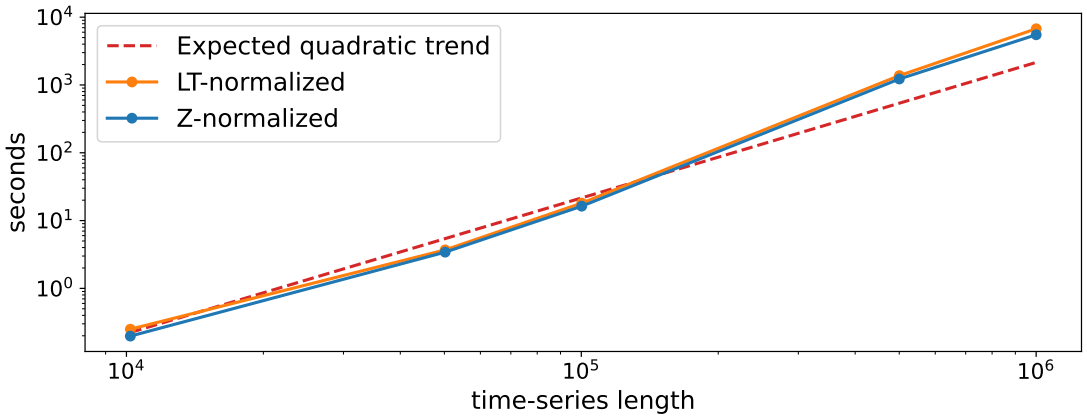


Figure 2.8 Scalability of the matrix profile with the time series length for LT-normalized (blue) and Z-normalized (orange) Euclidean distances.

linear on these datasets.

Additionally, the LT algorithm is almost parameter-free, while the STL+Z algorithm requires setting several parameters. Also, STL+Z is more time-consuming: it takes around 1 minute to process 100K samples with STL+Z, compared to a dozen seconds for LT. In the presence of a trend, the LT-normalized Euclidean distance is an efficient and robust first approach for motif discovery.

Scalability

Experiment presentation. In this experiment, we evaluated the scalability of the matrix profile with respect to the time series length for the LT-normalized and Z-normalized Euclidean distances. We considered the STOMP algorithm [Zhu+16] to compute the matrix profile with both distances. We generated 50 time series based on the m-set scenario (S-2) with lengths of 10K, 50K, 100K, 500K, and 1M. We measured the computation time for a subsequence length of 100.

Results. The average computation time is shown in Figure 2.8. Even though the LT-normalized Euclidean distance generalizes the Z-normalized Euclidean distance and performs better on several tasks, the matrix profile's computation time is equivalent for both distances and evolves according to its quadratic complexity.

Conclusion

In the present chapter, we have outlined the general principles of similarity search and reviewed in detail the algorithms that solve the distance profiling problem or compute the matrix profile data structure. In both cases, these algorithms are restricted to the Z-normalized Euclidean distance as they rely on its recursive formulation to perform fast computations. While widely used, the Z-normalized Euclidean distance is sensitive to some deformations, like the presence of a trend. To overcome this issue, we have

proposed a general framework to create distances that preserve the recursion formula while being invariant to custom sets of deformations. In particular, we have illustrated the framework by introducing the LT-normalized Euclidean distance. This distance is invariant to amplitude scaling, offset shift and linear trend. We have shown that this distance is more robust to the deformation induced by a trend than the Z-normalized Euclidean distance, making it more reliable for similarity search or motif discovery.

In fact, similarity search is a critical subtask for motif discovery, and in the next chapter, we take advantage of the matrix profile and our custom distances to propose a novel algorithm for motif discovery based on persistent homology, a tool from topological data analysis.

Chapter 3

Motif discovery

Key points:

1. Motif discovery consists of the unsupervised detection and localization of local patterns that repeat themselves in a time series.

2. The mathematical definition of a motif is not unique, leading to the development of algorithms based on varying criteria. Some algorithms prioritize motif frequency, while others focus on the similarity between motif occurrences. Most methods depend on core hyperparameters, such as the number of motifs, motif length, or a similarity threshold, which are often difficult to define and typically determined through trial and error.

Contributions:

1. This chapter introduces an algorithm called PersistentPattern (PEPA) for discovering variable-length motifs without requiring prior knowledge of the similarity between motif occurrences. PEPA works by embedding a time series into a graph and summarizing it through persistent homology, a tool from topological data analysis, which then allows the identification of relevant motifs from the graph's summary.

2. An adaptive version of the algorithm that infers the number of motifs to discover from the graph summary is also presented.

3. A benchmark of 9 labeled datasets, including 6 real-world datasets, is introduced for motif discovery. Empirical evaluations show that PEPA significantly outperforms state-of-the-art algorithms.

Associated papers:

- Thibaut Germain, Charles Truong, and Laurent Oudre. "Persistence-based motif discovery in time series". In: *IEEE Transactions on Knowledge and Data Engineering* (2024)

- Thibaut Germain, Charles Truong, and Laurent Oudre. "Interactive motif discovery in time series with persistent homology". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2024, pp. 383–387

Contents

3.1 Introduction

The task of motif discovery. This chapter addresses the task of motif discovery, which corresponds to the process of identifying recurrent local patterns and locating their occurrences within a single time series without any prior assumptions regarding their shape or location. Motifs can reveal valuable insights into the underlying dynamics of the time series. For example, in electrocardiograms (ECGs), the heartbeat is the reference motif. However, ECGs of patients experiencing premature ventricular contractions present a second motif specific to the malfunction [MM01] (see Figure 3.1b). Additionally, motifs can serve as concise representations of long time series, facilitating downstream tasks such as classification and anomaly detection. These summarized representations can reduce computational complexity, enhance performance, and aid in interpreting results [TL17]. Motif discovery algorithms have been applied in various domains such as industry [ZGC20; SMR13], medicine [Liu+15; SR+15], and biology [FN19; Lee+18].

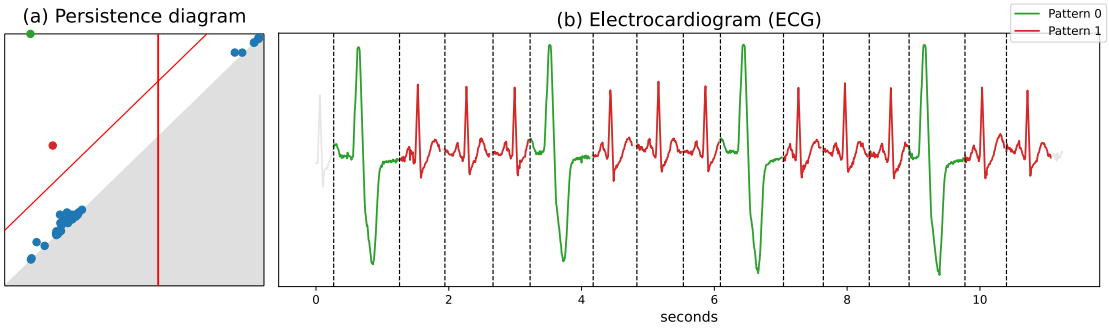


Figure 3.1 On the left, the representation of the time series by a persistence diagram, and on the right, the motif sets discovered by our algorithm on an electrocardiogram (ECG) of a patient suffering from premature ventricular contraction (PVC). **(a) Persistence diagram:** it is a simplified representation of the ECG that shows the existence of two significant motif sets (in green and red). **(b) Electrocardiogram:** pattern 0 (green) represents heartbeats with PVC, and pattern 1 (red) represents normal heartbeats. Vertical dashed lines on the ECG indicate the start location of patterns' occurrences.

Limitations of motif discovery algorithms. The mathematical definition of motif is not unique, leading to the development of algorithms based on different criteria. Some algorithms prioritize the motif frequency [SL22; GSS16; Lin+02], focusing on identifying the most frequently recurring patterns, while others emphasize the similarity between motif occurrences [Zhu+16; CCN10; ZMK19; Lin+18], aiming to detect patterns with highly consistent occurrences. Most of these algorithms rely on three core parameters: the number of motifs to discover, the motif length, and a similarity threshold between motif occurrences. These parameters highlight the current limitations of state-of-the-art algorithms. Indeed, the number of motifs to discover is imposed, and few guarantees exist whether the number of motifs is overestimated or underestimated. As well, the first algorithms supposed that all occurrences of all motifs have the same length [Lin+02; BHL14; GSS16], but more recent algorithms leverage this issue by searching for motifs within a length range [Sen+14; Lin+18]. Finally, the similarity threshold is hard to determine as it depends on the variances between occurrences of each repeated pattern. In practice, this parameter is set by trial-and-error [SL22].

The proposed algorithm. With regard to the current limitations, we propose a scalable algorithm that finds motifs of variable length without requiring a similarity threshold. The algorithm is based on a novel criterion: the persistence of motifs. Persistent homology is a central tool in topological data analysis [BCY18] that efficiently tracks topological features at different spatial resolutions. In our context, the algorithm tracks motifs for all similarity thresholds and returns motifs that persist across the largest ranges of scales. Intuitively, the persistence of a motif simultaneously measures the similarity between its occurrences and their dissimilarity with the rest of the time series. The algorithm discovers motifs by mapping a time series onto a graph and creating a summary of the graph from which the most persistent motifs are identified. The algorithm also provides an intuitive visual representation of the graph summary, called persistence diagram, that informs

about the number of relevant repeated patterns in a time series (see Figure 3.1a). Taking advantage of this representation, we also present an adaptive version of the algorithm that infers the number of motifs to discover from the persistence diagram. Finally, in our experimental evaluation, we show that:

- Both algorithms significantly outperform 6 state-of-the-art algorithms on 9 labeled datasets, including 6 real-world datasets.
- Hyperparameters have limited influence on the algorithms' performances.
- Like state-of-the-art algorithms, the theoretical and empirical time complexity is quadratic in the time series' length.

Chapter outlines. The chapter is organized as follows: In Section 3.2, we review the related work and detail our contributions. In Section 3.3, we present both algorithms. In Section 3.4, we describe the experimental settings. In Section 3.5, we review the experimental results and in Section 3.6, we present an interactive application for motif discovery.

3.2 Background

In this section, we first recall some definitions related to time series and motif discovery, then we discuss related work and finally, we describe the scientific positioning of our approach.

3.2.1 Definitions

We denote by S^l the set of all subsequences of length l of a time series $\mathbf{s} \in \mathbb{R}^n$, and we assume a distance function $d : \mathbb{R}^l \times \mathbb{R}^l \mapsto \mathbb{R}_+$ for the following definitions.

Definition 6. *(r-match) Given a threshold $r > 0$, the subsequences $\mathbf{s}_i^l$ and $\mathbf{s}_j^l$ of a time series $\mathbf{s} \in \mathbb{R}^n$ are r-matching iff $d(\mathbf{s}_i^l, \mathbf{s}_j^l) < r$.*

Several motif discovery algorithms consider the following definitions of motif set:

Definition 7. *(Spherical motif set, [Lin+02]) Given a threshold $r > 0$, the spherical motif set associated with a sequence $\mathbf{c}$ of length l is the largest set of non-overlapping subsequences of length l of $\mathbf{s}$ such that all subsequences are r-matching with $\mathbf{c}$. The sequence $\mathbf{c}$ is called the core element of the spherical motif set.*

Definition 8. *(Bi-spherical motif set, [Lin+18]) Given a threshold $r > 0$, the bi-spherical motif set associated with a pair of sequences $(\mathbf{c}_1, \mathbf{c}_2)$ of length l is the largest set of non-overlapping subsequences of length l of S such that all subsequences are r-matching with either $\mathbf{c}_1$ or $\mathbf{c}_2$. The sequences $\mathbf{c}_1$ and $\mathbf{c}_2$ are called the core elements of the bi-spherical motif set.*

3.2.2 Related work

One name several meanings. In the literature, motif discovery in time series refers to a number of distinct problems that belong to two primary categories:

- **Motif Pair Discovery**: Identifying the two most similar non-overlapping subsequences in a time series.
- **Motif Set Discovery**: Identifying sets of subsequences that encompass every occurrence of distinct repeated patterns in a time series.

In this chapter, we focus on the motif set discovery task. For more information on motif pair discovery, we refer the interested reader to the reviews [TL17; SL22].

Concerning motif set discovery, current algorithms address the problem either with a frequency or similarity approach. Both approaches are further described in the following paragraphs. As we will see, these approaches are disjoint and result in different definitions of motif sets, which are illustrated in Figure 3.2. As well, the main algorithms for motif set discovery are summarized in Table 3.1.

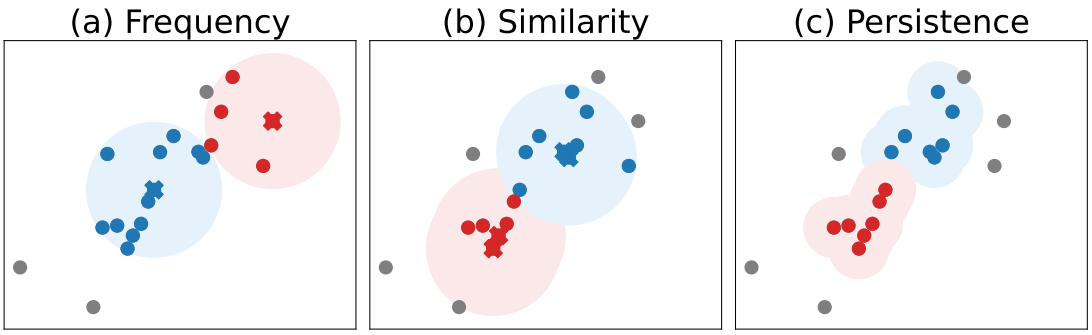


Figure 3.2 Three approaches for motif set discovery. The points represent the subsequences and the algorithms search for two motif sets. The predicted sets are in red and blue, and the points in gray are the outliers. The cross represents the barycenter of motif sets and they are as well subsequences. **(a) Frequency based:** Clusters are obtained with SetFinder [BHL14]. The clusters are the largest sets contained in non-overlapping balls of radius r and centered on some subsequences. **(b) Similarity based:** Cluster are obtained with VALMOD [Lin+18]. The clusters are sets included in balls of radius r centered on the two most similar pairs of subsequences. **(c) Persistence based:** Clusters obtained with PEPA. The clusters are formed incrementally by similarity.

Frequency-based algorithms. The frequency-based algorithms aim to identify the sets of subsequences representing the most frequently repeated patterns. The first motif set discovery algorithm, named EMMA [Lin+02], follows the frequency-based approach. Given the number of motifs to discover, a fixed subsequence length, and a similarity threshold $r > 0$, the algorithm iteratively finds the largest spherical motif set with radius $r > 0$ and centered on a subsequence of length l of the time series. Additionally, the motif sets are chosen so that the spheres do not overlap, meaning that the core elements are at

least a distance of $2r$ from each other. For computational efficiency, subsequences are discretized using Symbolic Aggregate approXimation (SAX) [Lin+07], and subsequences with similar symbols are grouped in sets. The sets are refined in a post-processing step to obtain the final spherical motif sets according to the Euclidean distance. Similarly to EMMA algorithm a more recent algorithm SetFinder [BHL14], returns exact solutions when working with the Euclidean distance or the Z-normalized Euclidean distance [DAV19]. This algorithm computes the spherical motif set of all subsequences and selects the largest sets while preserving the non-overlapping constraint between motif sets. For EMMA and SetFinder, motif sets' core elements are an approximation of barycenters by real subsequences of the time series, which, unfortunately, makes the discovery of motif sets sensitive to noise. The LatentMotif [GSS16] algorithm addresses this issue by considering an optimization problem that maximizes the total motif frequencies in order to learn the core elements of the spherical motif sets. The non-overlapping constraint between spheres is relaxed and encoded via a penalty function incorporated in the optimization criteria. Nevertheless, an optimal solution is not guaranteed as the optimization problem is non-convex.

All previous algorithms consider the subsequence length fixed and equal across all motif sets. However, a time series may contain repeated patterns of variable length. Early on, the Grammarviz algorithm [Sen+14] has been proposed to address this issue. The algorithm initially discretizes the time series as a long sequence of symbols with the SAX representation. Then, the longest and most repeated patterns are identified with the grammar induction algorithm Sequitur [NW97] that creates a hierarchical structure between the repeated patterns. While being time-efficient, the algorithm is sensitive to the discretization step, as the identification of repeated patterns relies on an exact match between symbolic representations of subsequences. Several variants of Grammarviz have been proposed to address this problem [Sen+18; GL17; GL19].

A recent algorithm, k-motiflets [SL22], focuses on finding the best motif set that contains k non-overlapping subsequences of a fixed length with minimal pairwise distances. Compared to previous algorithms, the similarity threshold is more intuitive as the set is based on the number of occurrences. In addition, a motif set is not centered on a subsequence but needs to contain the k subsequences included in a sphere of minimal radius. The authors also suggest heuristics to determine appropriate numbers of occurrences and subsequence lengths. However, this algorithm focuses on one of the motif sets for a time series with multiple repeated patterns.

The frequency-based algorithms rely on a similarity threshold to determine the radius of spheres enclosing the motif sets. This threshold can be difficult to set and is assumed to be the same for all motif sets. When this assumption does not hold, the motif sets may contain false occurrences or miss true occurrences.

Similarity-based algorithms. Similarity-based algorithms aim to identify sets of subsequences that represent repeated patterns with minimal variability between occurrences, regardless of frequency. Unlike frequency-based methods, which may overlook patterns that appear only once or twice, similarity-based approaches are capable of detecting such occurrences.

Table 3.1 VL: variable length, **NP:** number of parameters, **Complexity:** worst case time complexity. n: time series length, l: subsequence length, k: number of motif' occurrences, K: number of subsequence' neighbors.

Approach	Algorithm	VL	NP	Complexity
Frequency	EMMA		5	$\mathcal{O}(ln^2)$
	SetFinder		3	$\mathcal{O}(n^3)$
	LearnMotifs		4	$\mathcal{O}(ln)$
	k-Motiflets		4	$\mathcal{O}(kn^2 + nk^2)$
	GrammarViz	✓	5	$\mathcal{O}(ln)$
Similarity	STOMP		3	$\mathcal{O}(n^2)$
	VALMOD	✓	5	$\mathcal{O}((l_{max} - l_{min})n^2)$
	MDLC	✓	3	$\mathcal{O}(n^3/l_{min} + (l_{max} - l_{min})n^2)$
Persistence	PEPA	✓	3	$\mathcal{O}(Kn^2)$
	A-PEPA	✓	2	$\mathcal{O}(Kn^2)$

One of the earliest similarity-based algorithms, MDLC [Rak+12a], clusters subsequences of variable length such that the description length of the time series with the clusters is minimal. It is a greedy bottom-up algorithm that, at each iteration, either creates a cluster, adds a subsequence to a cluster, or merges two clusters. The choice of action is based on the number of bits saved. Unlike the more recent algorithms, STOMP [Zhu+16; Ben+20] and VALMOD [Lin+18], MDLC does not require the number of clusters as input, and it stops clustering when the time series description length cannot be improved.

The declinations of STOMP and VALMOD into motif discovery algorithms leverage the 1-NN matrix profile, a data structure representing the nearest non-overlapping subsequence graph [Yeh+16]. Given the number of motifs, a fixed subsequence length, and a similarity ratio $\lambda > 0$, STOMP algorithm iteratively finds spherical motif sets whose core element corresponds to the left member of the most similar non-overlapping subsequence pair. STOMP also maintains the non-overlapping constraint between all subsequences of all motif sets. To that end, a mask containing all subsequences that overlap with previously selected motif sets is maintained across iterations. The most similar pairs are found with the matrix profile, and the associated spherical motif sets are retrieved by distance profiling with the MASS algorithm [ZM24].

The second algorithm, VALMOD, differs from STOMP in two ways: it searches for motif sets with subsequences of variable length within a range $[l_{min}, \ldots, l_{max}]$ and it considers bi-spherical motif sets whose core elements are the most similar subsequence pairs. More precisely, the subsequences have the same length in each motif set, but the length can differ from one set to another. The z-normalized Euclidean distance is divided by the subsequence length to compare subsequences of different lengths. The matrix profile also stores the length of the nearest non-overlapping subsequence. Finally, motif sets are found using the same resolution scheme as STOMP algorithm.

In contrast to frequency-based algorithms, similarity-based algorithms assume that the radius of the balls in which motif sets are contained differs for each set. The radius is proportional to the z-normalized Euclidean distance between the subsequences of the most

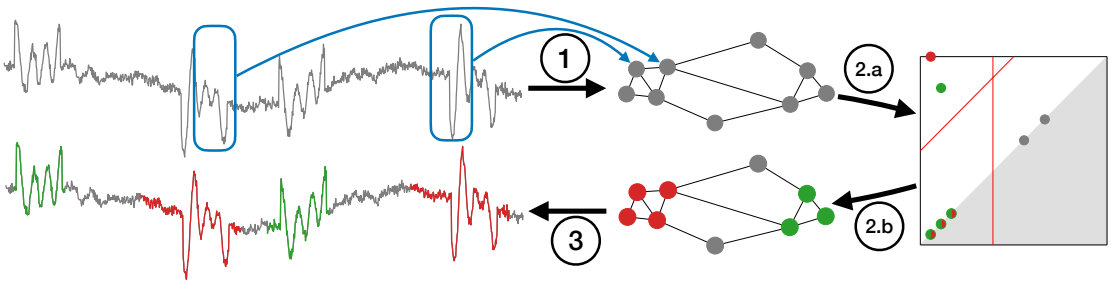


Figure 3.3 (1) From time series to graph: Nodes are subsequences, and edges depend on the distance between them. **(2) Graph clustering from persistent homology. (a) From graph to persistence diagram:** The graph is summarized by a diagram where each point is a connected subgraph, and its location depends on the weight of some edges. **(b) From persistence diagram to clusters:** Subgraphs associated with motif sets are in the upper left corner. They can be isolated with two thresholds: the red lines. There are two clusters, red/green; the bi-colored points are subgraphs included in the subgraphs of cluster, but their membership cannot be determined from the persistence diagram. The gray points are subgraphs associated with irrelevant parts of the time series. **(3) From clusters to motif sets:** time-adjacent subsequences are merged in each cluster.

similar pair associated with the motif set. The proportionality is defined by a similarity ratio $\lambda > 0$. With such a definition, motif sets are sensitive to the nearest neighbor pairs, and small perturbations can lead to different sets.

3.2.3 Contributions and scientific positioning

Sensitivity to the similarity threshold. All presented algorithms consider motif sets as collections of subsequences contained in balls whose radius is determined by a similarity threshold. However, setting this parameter is not straightforward as it requires prior knowledge about the similarity between subsequences of repeated patterns. In practice, the threshold is often set by trial and error [SL22], but this strategy is not tractable for large time series.

A novel motif set definition robust to similarity thresholds. Our algorithm, called PersistentPattern (PEPA), takes a different approach. It creates motif sets without any prior knowledge about the similarity between subsequences. Indeed, the algorithm generates motif sets for all possible similarity threshold values, ranging from 0 to $+\infty$. Some motif sets persist across multiple threshold values, and the algorithm selects motif sets with the largest persistence range. In addition, the algorithm does not impose a spherical constraint on the motif sets. Instead, the subsequences are incrementally grouped to form the motif sets based on their similarity. It allows the motif sets to adapt to the local shape of the neighborhood of the subsequences. Ultimately, the algorithm searches for repeated patterns of variable length that are significantly different from each other regardless of their frequency. This approach is illustrated in Figure 3.2.

A heuristic to infer the number of motifs. All presented algorithms also require the number of motif sets as a parameter. Although difficult to define without prior knowledge, our persistence-based approach provides a simplified representation of a time series from which it is possible to infer the number of motif sets. Thus, we will also present an adaptive version of the main algorithm (A-PEPA) that automatically infers the number of motif sets to be discovered.

3.3 Method

An overview of the method. Our approach is based on two main ingredients: a graph that encodes the structural relationships between all subsequences in the time series, and the use of persistent homology (a tool derived from topological data analysis) to isolate and identify the motif sets. The algorithm PEPA can be broken down into three steps illustrated in Figure 3.3:

1. **From time series to graph:** Transforming a time series into a graph where nodes represent subsequences and edges are weighted with the distance between subsequences. The graph is an adaptation of the k-nearest neighbor graph, which incorporates similarity and time dependence of subsequences.

2. **Graph clustering through persistent homology:** Identifying clusters representing motif sets and separating them from the isolated nodes of the graph representing irrelevant parts of the time series.

3. **From clusters to motif sets:** Merging temporally adjacent subsequences in each cluster to form the variable length motif sets.

3.3.1 From time series to graph

This section describes the transformation of a time series $\mathbf{s} \in \mathbb{R}^n$ into an undirected weighted graph $\mathcal{G}_\mathbf{s}$. For the moment, let d be a distance function between subsequences. We will further detail the distance later in this section.

Definition 9 (Undirected weighted graph)**.** *An undirected weighted graph* $\mathcal{G} = (\mathsf{V}, \mathsf{E})$ *consists of a set of vertices (also called nodes)* V *and a set of weighted edges* $\mathsf{E} \subset \{(x, y, w_{xy}) \mid (x, y) \in \mathsf{V} \times \mathsf{V},\ w_{xy} \in \mathbb{R}_+\}$*, where* w_{xy} *is the weight of the edge between nodes* x *and* y*.*

In $\mathcal{G}_\mathbf{s}$, the set of nodes is composed of all subsequences of length l of $\mathbf{s}$ (denoted as S^l) and the edges between subsequences are defined according to two criteria:

- **Similarity:** each subsequence $\mathbf{s}_i^l$ is connected to its K most similar non-overlapping subsequences.

- **Time:** it connects with its time adjacent subsequences $(\mathbf{s}_{i-1}^l, \mathbf{s}_{i+1}^l)$.

More formally, let $\mathbf{n}_i^k$ denote the k-th nearest non-overlapping neighbor of the subsequence $\mathbf{s}_i^l$ and d_i^k the distance between subsequences $\mathbf{s}_i^l$ and $\mathbf{n}_i^k$. The **similarity** edges are defined by the set:

$$\mathsf{E}_1 = \bigcup_{i=1}^{n-l+1} \left\{ \left(\mathbf{s}_i^l, \mathbf{n}_i^k, d_i^k \right) \mid k = 1, \ldots, K \right\}, \tag{3.1}$$

and the **time** edges, by the set:

$$\mathsf{E}_2 = \bigcup_{i=1}^{n-l} \{ (\mathbf{s}_i^l, \mathbf{s}_{i+1}^l, \max\left(d_i^1, d_{i+1}^1\right) \}. \tag{3.2}$$

Intuition on the graph. The final graph $\mathcal{G}_\mathbf{s}$ is defined as $(\mathsf{S}^l, \mathsf{E}_1 \cup \mathsf{E}_2)$. Intuitively, low-weight edges connect similar subsequences, while high-weight edges connect less similar ones. The graph $(\mathsf{S}^l, \mathsf{E}_1)$ is a variant of the well-known k-nearest neighbor graph, as it connects each subsequence to its k-nearest non-overlapping neighbors. In practice, this graph can split a single motif set into several clusters as subsequences are considered independent and they are not compared with time-adjacent subsequences. In such a situation, the number of discovered motif sets is over-estimated, and a non-trivial post-processing is needed to merge similar clusters. We introduce the time edges set E_2 to prevent this phenomenon: in our graph, $\mathcal{G}_\mathbf{s}$, a subsequence is always connected to the subsequences just before $\mathbf{s}_{i-1}^l$ and after $\mathbf{s}_{i+1}^l$. As a result, our method assigns overlapping subsequences that represent the same repeated pattern in a single cluster, which limits the over-clustering effect.

Proposition 4. *The graph $\mathcal{G}_\mathbf{s}$ associated with the time series $\mathbf{s} \in \mathbb{R}^n$ is connected.*

Proof. The graph $(\mathsf{S}^l, \mathsf{E}_2)$ is a path graph; thus, it is connected, and by the union of two graphs, $\mathcal{G}_\mathbf{s}$ is connected. □

Choice of distance. Intuitively, the construction of the graph $\mathcal{G}_\mathbf{s}$ can be described in two steps: first, computing the similarity graph, which corresponds to the K-NN matrix profile, and then, inferring the time edges from the similarity graph. As we have seen in Section 2.1.3, the computation of the K-NN matrix profile can be done in $\mathsf{O}(Kn^2)$ with distances invariant to some rigid deformations which also verify a recursive formulation (see Section 2.2.4). In what follows, we assume to work with such distances. However, if efficiency is not a concern, one can use more complex distances like the DTW or other elastic distances. For the experiments described in subsequent sections, the LT-normalized Euclidean distance serves as a baseline (see Section 2.3).

Improving the distance. Any distance invariant to rigid deformation defined through the framework described in Section 2.2.4 take value in $[0, 2]$. Given the expected properties of the graph $\mathcal{G}_\mathbf{s}$, the distance should be able to distinguish between subsequences corresponding to the same repeated pattern and those that do not. To enhance this

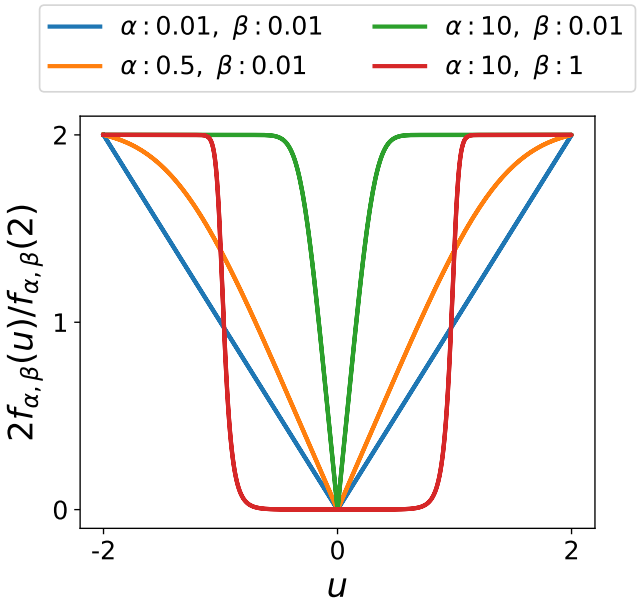


Figure 3.4 The (α, β)-rectification: The blue line represents distance behavior with the absolute value on the range of $[-2, 2]$. It is obtained for small value of α and β. As α increases, the distance becomes more restrictive as its value remains low on smaller neighborhoods centered on 0 and tends towards 2 outside the neighborhood. For high α, the distance becomes more permissive when β increases as the distance value remains low in larger neighborhoods centered on 0.

behavior, we introduce the (α, β)-rectification, which applies a soft polarization to the distance values near the boundaries of the interval $[0, 2]$. The rectified distance can be understood as a parametric, kernelized version of the original distance, with parameters controlling the variability tolerance. Formally, the (α, β)-rectification is defined as follows:

Definition 10. *Let $\alpha \in \mathbb{R}_+^*$, $\beta \in [0, 2[$, and $d : \mathbb{R}^l \times \mathbb{R}^l \mapsto [0, 2]$ be a distance function, the (α, β)-rectified distance between $\mathbf{x} \in \mathbb{R}^l$ and $\mathbf{y} \in \mathbb{R}^l$ is given by:*

$$d_{\alpha,\beta}(\mathbf{x}, \mathbf{y}) = 2 f_{\alpha,\beta}(d(\mathbf{x}, \mathbf{y}))/f_{\alpha,\beta}(2) \tag{3.3}$$

with $f_{\alpha,\beta}(x) = \sqrt{\tanh(\alpha\beta^2) + \tanh(\alpha(x^2 - \beta^2))}$

Figure 3.4 shows the influence of the parameters α and β on the function $f_{\alpha,\beta}$. The parameter α controls the polarization of the distance on the limits of the interval $[0, 2]$ while β acts as a threshold on the raw distance value. In fact, let β be fixed then $\lim_{\alpha\to\infty} f_{\alpha,\beta}/f_{\alpha,\beta}(2) = \mathbb{1}_{]\beta,2]}$.

3.3.2 Graph clustering through persistent homology

Persistent homology & graph clustering. The graph $\mathcal{G}_\mathbf{s}$ is such that overlapping subsequences from the same motif set are connected with edges of low weight. Therefore,

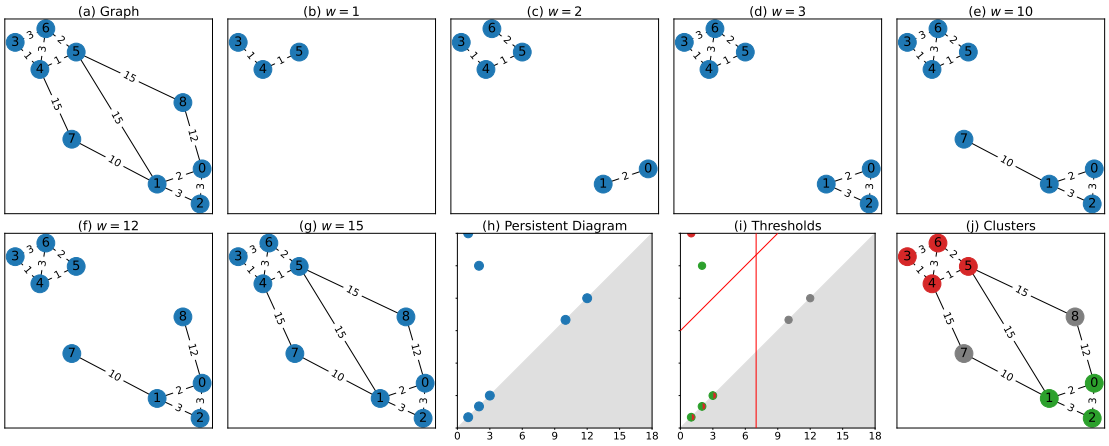


Figure 3.5 Illustration of the graph clustering algorithm through persistent homology. **(a) The graph to cluster:** The number on the nodes is their id, and the weights are the distance between nodes. **(b)-(g) The NNVR filtration milestones of the graph:** The edges are added in order of increasing weight. **(h) The corresponding persistence diagram:** The births and deaths of connected subgraphs traced along the filtration are summarized with a 2D scatterplot where births are on the x-axis and deaths are on the y-axis. **(i) The birth and persistence thresholds:** The red and green points in the upper-left corner indicate two clusters. The birth threshold (vertical red line) and the persistence threshold (off-diagonal red line) are set to isolate this region. The bi-colored points are subparts of the cluster, but their membership cannot be determined from the persistence diagram. The gray points are associated with irrelevant parts of the time series. **(j) Clustering result:** Clusters are in red and green; irrelevant nodes are in gray.

motif sets can be retrieved by searching for connected subgraphs of $\mathcal{G}_\mathbf{s}$ with edges of low weight. Persistent homology is well-suited for identifying and isolating such subgraphs. Persistent homology is a central tool in the field of topological data analysis [BCY18], used to track the persistence of topological features of data at multiple scales with respect to a scaling parameter. To summarize the persistence of these features, a 2D scatter plot known as a Persistence Diagram is created, allowing for the identification of noteworthy topological features. For a thorough description of persistent homology and its application in various fields, readers can refer to [EH+08; PLX22].

In our context, the topological features are the connected subgraphs of $\mathcal{G}_\mathbf{s}$, and the scaling parameter is the edge weight. The graph clustering algorithm consists of three steps:

1. Computing the persistence of connected subgraphs
2. Identifying connected subgraphs related to motif sets from the persistence diagram
3. Forming clusters from the chosen connected subgraphs

Computing the persistence of connected subgraphs

A graph filtration. The persistence of connected subgraphs of $\mathcal{G}_\mathbf{s}$ is computed through a sequence of nested graphs. The sequence starts with the empty graph and adds edges

one by one, in order of increasing weight until it reaches the final graph. In persistent homology, such sequence is called a filtration. There are several types of filtration, and we have implemented the Nearest Neighbor Vietoris-Rips Filtration (NNVR) [BTO24].

Definition 11 (Nearest Neighbor Vietoris-Rips Filtration)**.** *Let $\mathcal{G} = (\mathsf{V}, \mathsf{E})$ be a weighted graph with $(w_i)_{i=1,\dots,m}$ being the edge weights in ascending order. The nearest neighbor Vietoris-Rips filtration is the sequence of nested graphs:*

$$\emptyset = \mathcal{G}_{w_1} \subsetneq \mathcal{G}_{w_2} \subsetneq \dots \subsetneq \mathcal{G}_{w_{m-1}} \subsetneq \mathcal{G}_{w_m} = \mathcal{G}$$

where $\emptyset$ is the empty graph and $\mathcal{G}_{w_i} = (V_{w_i}, E_{w_i})$ such that:

$$V_{w_i} = \left\{ v_x \in \mathsf{V} \mid \min_{(v_x, v_y, w_{xy}) \in E} w_{xy} \leq w_i \right\}$$

and

$$\mathsf{E}_{w_i} = \{(v_x, v_y, w_{x,y}) \in E \mid w_{xy} \leq w_i\}$$

Tracking the persistence of connected subgraphs. By convention, when adding an edge in the filtration, the nodes it connects are added first if they are not already in the filtration, then the edge itself is added. If both nodes need to be added, one is arbitrarily added before the other. Alongside the filtration, we keep track of the birth and death dates of connected subgraphs:

- **Birth:** The birth of a connected subgraph occurs when a node is added to the graph; its birth date is equal to the weight of the associated edge.
- **Death:** A connected subgraph dies when an edge connects it to an older connected subgraph. Its death date is equal to the weight of the connecting edge.

By definition, each subsequence is the seed node of a connected subgraph, so they all have a birth date equal to the distance to their nearest non-overlapping neighbors. Since the graph of a time series $\mathcal{G}_{\mathbf{s}}$ is connected, one connected subgraph never dies; its death date is set to $+\infty$. We denote $(b_i, d_i)_{i \in I}$ as the set of birth and death dates of all connected subgraphs traced by the filtration. The persistence of the i^{st} connected subgraph corresponds to its lifetime: $d_i - b_i$. The connected subgraphs are summarized with a 2D-scatterplot called Persistence Diagram, where births are on the x-axis, and deaths are on the y-axis. Each point is counted with multiplicity since several connected subgraphs can have the same birth and death dates.

An example. Figure 3.5 shows milestones of an NNVR filtration on a graph (Figure 3.5a to Figure 3.5g) and the corresponding persistence diagram (Figure 3.5h). When weight $w = 1$, node 3 has killed nodes 4 and 5, so their persistence is zero. With weight $w = 2$, node 0 kills node 1 to form a second independent connected subgraph. Then, nodes 6, 2, 7, and 8 are added and immediately killed for weights $w = 2, 3, 10$, and 12. At weight $w = 15$, the subgraph associated with node 0 is killed by that of node 3 and the graph is complete. The highest point (in red) on the persistence diagram corresponds to the subgraph that never dies.

Algorithm 7 ComputePersistence

Require: $\mathcal{G} = (i, j, w_{ij})_{(i,j)\in I}$ a graph, I is sorted by weight.
$\mathcal{P} \leftarrow \{\}$ Parent dictionary, $\mathcal{B} \leftarrow \{\}$ Birth dictionary, $\mathcal{D} \leftarrow \{\}$ Death dictionary, $MST \leftarrow ()$ Minimum spanning tree
for $(i, j) \in I$ **do**
  $P_1 \leftarrow FindParent(i)$
  if P_1 is empty **then**
    $\mathcal{P}[i] \leftarrow i,\ \mathcal{B}[i] \leftarrow w_{ij}$
  $P_2 \leftarrow FindParent(j)$
  if P_2 is empty **then**
    $\mathcal{P}[j] \leftarrow j,\ \mathcal{B}[j] \leftarrow w_{ij}$
  if $P_1 \neq P_2$ **then**
    if $\mathcal{B}[P_1] < \mathcal{B}[P_2]$ **then**
      $\mathcal{P}[P_2] \leftarrow P_1,\ \mathcal{D}[P_2] \leftarrow w_{ij}$
    else
      $\mathcal{P}[P_1] \leftarrow P_2,\ \mathcal{D}[P_1] \leftarrow w_{ij}$
    $MST.append((i, j, w_{ij}))$
return $\mathcal{B}, \mathcal{D}, MST$

Computing the births and deaths. The birth and death of connected subgraphs are tracked by maintaining a union-find data structure throughout the filtration. The algorithm is equivalent to the Kruskall's algorithm for computing the minimum spanning tree (MST). Algorithm 7 describes the procedure for computing the connected subgraphs persistence from a graph whose edges are ordered by increasing weight. The FindParent command follows the chain of parent pointers from a query node until a root node. This root node represents the connected subgraph to which the query node belongs. The algorithm also stores the MST to efficiently retrieve the clusters later on.

Identifying connected subgraphs related to motif sets from the persistence diagram

An interpretable persistent diagram. As shown in Figure 3.6a, the persistence diagram can be divided into three interpretable regions:

1. **Top-left corner:** Points represent connected subgraphs associated with motif sets.
2. **Lower-left corner:** Points represent connected subgraphs associated with minor variations of the motif sets.
3. **Right side:** Points represent connected subgraphs associated with irrelevant parts of the time series.

On one hand, the graph associated with a time series is constructed so that non-repeating or noisy linear subsequences are far from any other subsequence. Thus, connected subgraphs composed of these subsequences have a late birth and are located in the right part of the persistence diagram. On the other hand, subsequences that overlap

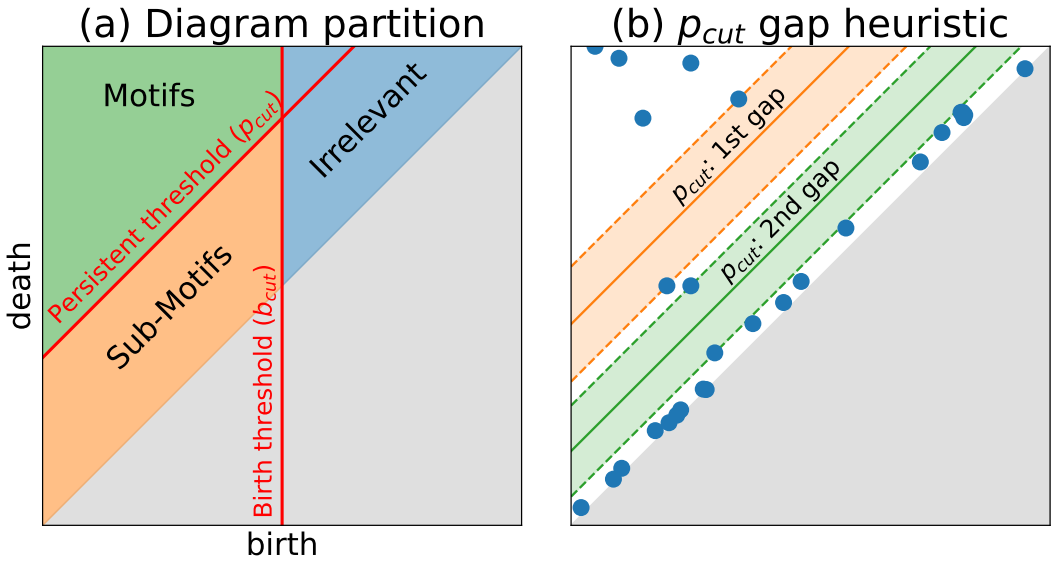


Figure 3.6 (a) Diagram partition: Top-left corner, points associated with motif sets. Lower-left corner, points associated with subparts of motifs sets. Right side, irrelevant parts of the time series. **(b) Persistence threshold heuristic:** The orange line is the persistence threshold of the largest gap, and the green line is the threshold of the second largest gap.

any occurrence of a repeated pattern are close to each other and far from all other subsequences. The connected subgraphs associated with repeated patterns have early births and late deaths. They are located in the top-left corner of the persistence diagram. Finally, the lower-right corner corresponds to connected subgraphs associated with minor variations of repeated patterns.

Inferring the cluster thresholds. The region associated with motif sets can be isolated with a vertical line and an off-diagonal line as presented in Figure 3.6-A. The vertical line corresponds to a threshold on birth dates (b_{cut}). It defines the difference between irrelevant subsequences and subsequences belonging to motif sets. This threshold is computed with Otsu's method [Ots79], an algorithm introduced in image processing to transform a grayscale image into a black-and-white image. The off-diagonal line corresponds to a threshold on the persistence (p_{cut}). Connected subgraphs associated with motif sets have a persistence greater than this threshold. To discover N motif sets, the persistence threshold is set to the average between the N-th and $(N+1)$-th most persistent connected subgraphs whose birth dates are less than the birth threshold.

Create clusters from the selected connected subgraphs

The clusters are computed by maintaining a union-find data structure throughout the filtration of the minimum spanning tree (MST) of $\mathcal{G}_{\mathbf{s}}$. The MST is computed in the first clustering step (Section 3.3.2), and it is the smallest graph that contains all information about the birth and the death of all the connected subgraphs traced by the filtration of $\mathcal{G}_{\mathbf{s}}$. The algorithm that computes the clusters is similar to the ComputePersistence algorithm

Algorithm 8 ComputeMotifSets

Require: S time series, I indexes of subsequences in clusters, $\mathcal{C}$ cluster dictionary, $\mathcal{B}$ birth dictionary, l subsequence length.

1: $\mathcal{O} \leftarrow \{\}$ Occurrence dictionary, $IDX \leftarrow ()$ Index to keep, $M \leftarrow ()$ Motif sets, $p\_idx \leftarrow I[0] - 1$ Previous index, $o\_id \leftarrow 1$ Occurrence ID
2: **for** $i \in sort(I)$ **do**
3: **if** $(i - 1 \neq p\_idx)$ or $(\mathcal{C}[i] \neq \mathcal{C}[p\_idx])$ **then**
4: $o\_id \leftarrow o\_id + 1$
5: $\mathcal{O}[i] \leftarrow o\_id$
6: $I' \leftarrow BirthOrderedIndex(I, \mathcal{B})$ ▷ *order the index list I by increasing order of birth date.*
7: **for** $i \in I'$ **do**
8: $IndexToKeep = True$
9: $J \leftarrow OverlappingIndex(i, IDX, l)$ ▷ *select index in IDX overlapping with i.*
10: **if** J *is not empty* **then**
11: **for** $j \in J$ **do**
12: **if** $O[i] \neq O[j]$ **then**
13: $IndexToKeep = False$
14: **break**
15: **if** $IndexToKeep$ *is* $True$ **then**
16: $IDX.append(i)$
17: $M \leftarrow MotifSetsFromSubsequences(S, IDX, \mathcal{C}, l)$
18: **return** M

(Algorithm 7) with two modifications. Specifically, the edges connecting two connected subgraphs with persistence higher than the persistence threshold are not considered when going through the filtration. This is done by changing the condition of the if-loop Line 9 to $(P_1 \neq P_2) + ((w_{ij} - \mathcal{B}[P_1] \leq p_{cut}) * (w_{ij} - \mathcal{B}[P_2] \leq p_{cut}))$. Second, after the main for-loop, the parent dictionary is updated, and the nodes whose birth date exceeds the birth threshold are removed. Ultimately, the algorithm returns N clusters from the parent dictionary, each composed of subsequences of length l of $\mathbf{s}$.

3.3.3 From clusters to motif sets

Merging time adjacent subsequences. Recall that a motif set is a set of non-overlapping subsequences of possibly varying length where elements of each motif should represent occurrences of the same repeated pattern. Our clustering approach produces clusters of overlapping subsequences, which must be refined to create a motif set. In particular, we merge specific overlapping subsequences to form a single one representing an occurrence. To that end, subsequences with the latest birth date are removed until the non-overlapping constraint is satisfied when merging subsequences into a single one per occurrence. The refined clustering is then a motif set.

Algorithm 9 AdaptivePersistenceThreshold

Require: $\mathcal{B}$ birth dictionary, $\mathcal{D}$ death dictionary, b_{cut} birth threshold, M number of gap

$P \leftarrow ()$ persistence list, $p_{cut} \leftarrow 0$ persistence threshold
for $i = 1, \ldots, n - l + 1$ **do**
    if $\mathcal{B}[i] \leq b_{cut}$ **then**
        $P.append(\mathcal{D}[i] - \mathcal{B}[i])$
$P \leftarrow Sort(P)$
for $i = 1, \ldots, M$ **do**
    $j \leftarrow \text{argmax}(P),\ p_{cut} \leftarrow (P[j+1] - P[j])/2$
    $P \leftarrow P[0 : j+1]$
return p_{cut}

Computing motif set. Algorithm 8 shows the procedure for computing the motif sets from clusters. The first for-loop computes the occurrence membership of all subsequences. Two subsequences belong to the occurrence if they are in the same cluster like all temporally consecutive subsequences between them. The second for-loop refines the clusters to enforce the non-overlapping constraint. If a subsequence overlaps with at least one subsequence of another occurrence with an earlier birth date, it is removed from its cluster. Finally, the motif sets are formed by merging the temporally consecutive subsequences in each cluster.

3.3.4 Adaptive algorithm: A-PEPA

In this section, we present an adaptive version of the PEPA algorithm called A-PEPA, which infers the number of motif sets from the persistence diagram. The only difference between PEPA and A-PEPA is the computation of the persistence threshold.

The PEPA algorithm isolates motif sets with a birth threshold and a persistence threshold based on the number of motif sets to discover. The adaptive version of the algorithm, A-PEPA, infers the persistence threshold by looking at successive gaps in persistence as shown in Figure 3.6b. A large gap indicates that repeated patterns (points above the gap) significantly differ from all other patterns in the time series (points below the gap). Depending on the application, the second or higher-order gap may be more interesting than the largest; some variations of more persistent repeated patterns should be considered as different motif sets. Algorithm 9 shows the procedure for computing the adaptive persistence threshold. In practice, we set the adaptive persistence threshold to the second-largest persistence gap.

3.3.5 Time complexity and parameter tuning

Time complexity. The time complexity of PEPA and A-PEPA is in $\mathcal{O}(Kn^2)$, where n is the length of the time series, and K is the number of nearest neighbors.

Indeed, the graph $\mathcal{G}_{\mathbf{s}}$ is computed in $\mathcal{O}(Kn^2)$ by following the procedure of the STOMP algorithm [Zhu+16]. The graph clustering algorithm is in $\mathcal{O}(Kn \log(Kn))$ in the worst case because the Algorithm 7 requires maintaining a union-find data structure over $\mathcal{G}_{\mathbf{s}}$

which has Kn edges. The computation of both thresholds is in $\mathcal{O}(n)$, and the algorithm that computes the clusters from the selected connected subgraphs is in $\mathcal{O}(n\log(n))$ since it requires maintaining a union-find data structure of the MST of $\mathcal{G}_{\mathbf{s}}$ which has n edges. The Algorithm 8 is in $\mathcal{O}(n\log(n))$ in the worst case because it requires sorting the subsequences by increasing order of birth dates. The bottleneck of PEPA and A-PEPA is the computation of the graph in $\mathcal{O}(Kn^2)$.

Parameter tuning. The PEPA algorithm has three parameters:

- The number of motif sets to discover: $N \in \mathbb{N}^*$. Note that this number is empirically estimated when using A-PEPA.
- Two parameters linked to the graph construction: the length of subsequences $l \in \mathbb{N}^*$ and the number of nearest neighbors $K \in \mathbb{N}^*$.

Like other motif discovery algorithms, setting the number of motif N depends on expert knowledge. However, with PEPA, this number can be deducted through the persistence diagram, Figure 3.6, and the motifs sets can be updated in $\mathcal{O}(n\log(n))$.

Empirical results (Section 3.5.3) shows that PEPA and A-PEPA are not sensitive to the number of neighbors K when it exceeds 5 (the relative error to the optimal is less than 1%). As this parameter influences the algorithms' computational time, we advise setting it to 5.

Experiment on the influence of the window length l (Section 3.5.3) shows that PEPA and A-PEPA retrieve motifs whose length are at most twice the window length. In practice, we recommend setting it to the length of the smallest motif.

3.4 Experimental settings

This section describes the datasets, performances metrics and the algorithms implementation details for our experimental evaluation. For reproducibility, the source code and all datasets are available on a Github repository[1]

3.4.1 Datasets

We conducted the experiments on 9 labeled datasets constructed from real and synthetic time series. Table 3.2 presents the main characteristics of the datasets related to motif set discovery. While the following paragraphs succinctly describe the datasets, a detailed presentation of each dataset can be found in Appendix A.1.

Real-world data. We have considered the following real-world univariate datasets:

(R-1) **mitdb-1**: ECGs from the The MIT-BIH Arrhythmia Database [Gol+00; MM01]. It contains 100 time series randomly selected from healthy patients such that they only contain normal heartbeats.

[1] `https://github.com/thibaut-germain/Persistent-Pattern-Discovery`

(R-2) **mitdb-2**: We randomly selected 100 ECGs from MIT-BIH. The number of repeated patterns varied between 1 and 4.

(R-3) **mitdb800**: ECGs sampled at a lower frequency than MIT-BIH. It results in a dataset containing a 100 long time series with a number of repeated patterns that vary between 1 and 4.

(R-4) **ptt-ppg**: Photoplethysmogram (PPGs) from the Pule-Transit-Time PPG dataset [Meh+22]. It contains 100 time series of a single pattern randomly selected from running subjects.

(R-5) **refit**: Aggregated time series of electrical consumptions of dishwasher, food mixer, washing machine, and tumble dryer for 10 houses. We kept 10 time series for each house in which the appliances were not used simultaneously. It resulted in a dataset of 100 time series with a maximum of 3 motif sets.

(R-6) **arm-coda**: Trajectories from the arm-coda datasets [Com+24]. It contains 64 time series of subjects performing various upper-limb movements.

Synthetic data. We have generated three datasets following three scenarios of increasing order of complexity for motif discovery:

(S-2) **single**: For similarity search. There is 1 pattern of length 100 that repeats 50 times. The dataset contains 100 time series.

(S-3) **fixed**: There are 5 patterns of length 100. For each pattern, the number of occurrences is sampled uniformly between 2 and 10. The dataset contains 100 time series.

(S-4) **variable**: There are 5 patterns with length uniformly sampled between 100 and 200. For each pattern, the number of occurrences is sampled uniformly between 2 and 10. The dataset contains 100 time series.

3.4.2 Performance metrics

We evaluate the performance with precision, recall, and f1-score metrics [Tat+18]. However, motif discovery in time series is an unsupervised task, and compared to supervised tasks, the computation of these metrics requires the additional step of pairing real and predicted motif sets. This step is a two-level assignment problem: predicted motif sets must be assigned to real motif sets, and predicted occurrences must be assigned to real ones between paired motif sets. The optimal pairings maximize the total overlapping length between real and predicted motif sets, and they can be efficiently computed with the Hungarian matching algorithm [Kuh55; Sar+21]. The precision, recall, and f1-score computation rely on the optimal pairings and a threshold $\tau \in [0, 1]$ that controls the overlapping ratio. Any metric's score is the average of the individual metric score between paired motif sets; the averaging can be macro or weighted. For precision (resp. recall), a motif occurrence is counted as a true positive if the ratio between the overlap length

Table 3.2 N number of repeated patterns, if $< k$, there are at most k patterns. μ_l average pattern length, σ_l standard deviation of pattern length, min/max minimum/maximum pattern length, n time series length, # number of time series.

Type	Name	N	μ_l	σ_l	min/max	n	#
real	(R-1) mitdb-1	1	320	60	215/461	20k	100
	(R-2) mitdb-2	< 4	280	70	69/496	20k	100
	(R-3) mitdb800	< 4	95	25	24/165	20k	100
	(R-4) ptt-ppg	1	325	45	201/461	20k	100
	(R-5) refit	< 3	100	20	47/143	20k	100
	(R-6) arm-coda	5	525	105	272/886	8k	64
synthetic	(S-2) single	1	100	0	100/100	8k	100
	(S-3) fixed	5	100	0	100/100	3k	100
	(S-4) variable	5	150	30	100/200	4k	100

and the predicted (resp. real) occurrence length is greater than the threshold τ. This threshold is set to 50% for all experiments. The resolution of the motif sets assignment problem and the metrics' computation are detailed in Appendix A.2.

We also rank methods according to the f1-score and compute critical difference diagrams [Dem06]. The associated test significance level is set to 0.05. We use Friedman's test to reject the null hypothesis, and we compute the critical differences using Nemenyi post-hoc test.

3.4.3 State-of-the-art methods and implementation details

The evaluation was performed on a server with Intel(R) Xeon(R) Gold 5220R CPU @ 2.20GHz, and 250 GB of RAM. We compared PEPA and A-PEPA with SetFinder (SF) [BHL14], LatentMotif (LM) [GSS16], Grammarviz (GM) [Sen+18], MDLC (MC) [Rak+12a], STOMP (SM) [Zhu+16], and VALMOD (VM) [Lin+18]. For fairness, we implemented all of the algorithms in Python except Grammarviz. Indeed, they all rely on a fast computation of the distance profiles, and we implemented a common structure based on the algorithms [Zhu+16].

VALMOD algorithm efficiently computes all matrix profiles within a subsequence length range thanks to STOMP algorithm and a pruning strategy. Therefore, STOMP algorithm provides a lower bound of VALMOD scalability performances. For simplicity, we implemented a greedy version of VALMOD algorithm that does not consider the pruning strategy. Predicted motif sets remain identical, and we use STOMP algorithm as a lower bound for VALMOD scalability performance.

For Grammarviz, we used a JAVA implementation provided by the authors [Sen+18].

3.5 Experimental Evaluation

Our experimental evaluation has four components:

- A qualitative evaluation on physiological signals,Section 3.5.1.

- A quantitative comparison of PEPA and A-PEPA with 6 state-of-the-art algorithms on several labeled real and synthetic datasets, Section 3.5.2.
- Several experiments to show the influence of the main parameters of PEPA and A-PEPA: the subsequence length, the number of nearest neighbors, and the persistence threshold heuristic for A-PEPA, Section 3.5.3.
- A scalability experiment, Section 3.5.4.

3.5.1 Qualitative evaluation

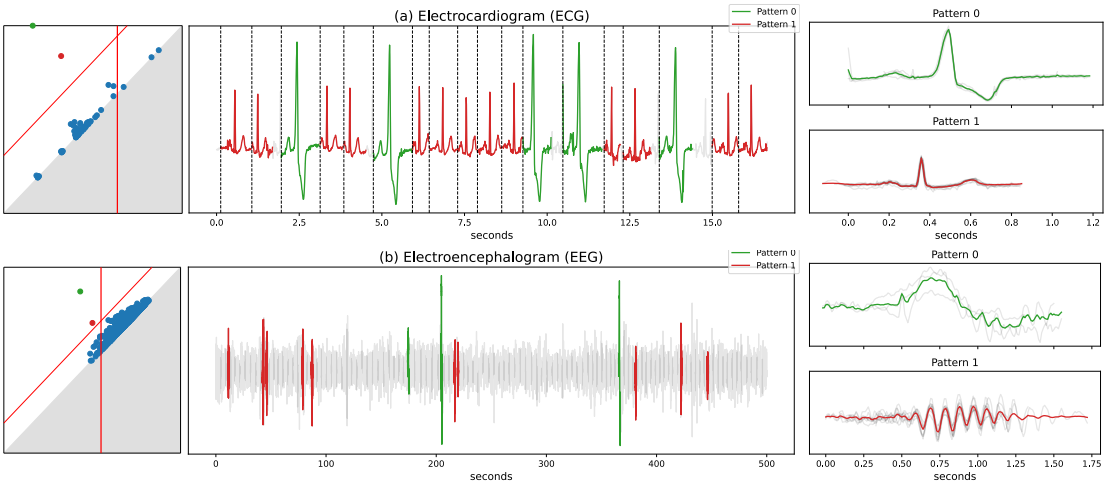


Figure 3.7 (left): persistence diagrams, (middle): time series with colored motif sets, (right): motif sets with barycenters. **(a) Electrocardiogram:** ECG of a patient with premature ventricular contraction (PVC). The persistence diagram shows two significant motif sets; pattern 0 represents heartbeats with PVC, and pattern 1 represents normal heartbeats. Vertical dashed lines on the time series plot indicate the start location of the pattern occurrences. **(b) Electroencephalogram:** single-channel EEG of a patient in a second stage of sleep. The persistence diagram indicates two significant motif sets; pattern 0 represents K-complexes, and pattern 1 represents sleep spindles. Both patterns represent short bursts of brain activity that help resist awakening by external stimuli.

In this section, we illustrate the visual interpretability of our algorithm and its ability to detect meaningful patterns in two types of physiological data.

ECG data. Electrocardiograms of patients suffering from premature ventricular contractions (PVCs) contain a typical pattern for normal heartbeats and another typical pattern for heartbeats with PVCs. Several ECGs in the mitdb800 database correspond to patients suffering from PVCs, and we ran the adaptive algorithm on a 16-second portion of one of them. We set the window length to 500ms and the number of neighbors to 5. The persistence diagram, Figure 3.7a (left), suggests that two motif sets have been properly isolated with the birth and persistence thresholds. Figure 3.7a (right) shows the motif sets; the first set corresponds to heartbeats with PVC, and the second corresponds to normal heartbeats. Figure 3.7a (middle) shows that all heartbeats are detected and

well classified except one normal heartbeat. Illustrations of motifs sets discovered with other motif discovery algorithms can be found in supplementary material.

EEG data. During the second stage of sleep, the brain activity slows down, except for short bursts of activity that help resist awakening by external stimuli. On an electroencephalogram (EEG), these short bursts of activity fall into two categories: the K-complexes and the sleep spindles [Mue+09]. A K-complex is the succession of high-voltage positive and negative peaks that last for about 600ms and occur every 1 or 2 minutes. Sleep spindles correspond to 11 to 16 Hz voltage oscillations and last for about 0.5 to 1.5 seconds. We ran the adaptive algorithm on the EEG of a patient in the second stage of sleep [Mue+09]. It is a single-channel EEG sampled at 100hz, and we selected a 500-second window. We set the window length to 1 second and the number of neighbors to 5. The persistence diagram, Figure 3.7b (left), shows that the algorithm has detected two motif sets. The first motif set gathers K-complexes, and the second set corresponds to sleep spindles, Figure 3.7b (right).

In both cases, the algorithm has detected patterns that account for the patients' physiological state. The persistence diagrams ensure the relevance of these patterns because they significantly detach from the rest of the time series.

3.5.2 Comparison with state-of-the-art algorithms

Experiment presentation. In this experiment, we evaluate the performance of PEPA and A-PEPA with 6 state-of-the-art algorithms on two tasks of increasing complexity:

- **occurrence detection:** Ability to localize pattern occurrences regardless of their motif set membership.
- **motif set discovery:** Ability to localize pattern occurrences and classify them according to their motif set membership.

Performances were evaluated in terms of precision, recall, and f1-score (Section 3.4.2) on all datasets presented in Section 3.4.1.

For PEPA and A-PEPA, the window length is set to the average pattern length minus its standard deviation, and the number of neighbors is set to 5 for each dataset. For SetFinder, LatentMotif, STOMP, and VALMOD, the window length is set to the average pattern length, and the radius is set with a gridsearch on each dataset. For Grammarviz, the window length is set to the average pattern length; the radius, the alphabet, and the word size are set with a gridsearch on each dataset. Parameters settings can be found in supplementary materials.

Results. For the occurrence detection task, results are shown in Table 3.3, and the critical difference diagram in Figure 3.8. We make several comments:

- PEPA and A-PEPA are the best-performing methods, with a mean rank significantly higher than other methods.
- Our approach has a relatively low f1-score on the refit data (0.31 only). However, it is still better than other methods by a margin. Motifs in refit are similar to square

waves, and normalized Euclidean distances have difficulties fully recovering such patterns.

- On ptt-pgg, methods based on the Z-normalized distance have low recalls (0.43 at best), contrary to PEPA and A-PEPA, which have markedly higher recall (0.62 and 0.66) thanks to the LT-normalized distance. Indeed, motifs in PPG signals are significantly affected by the trend induced by subjects' motions.

For the motif set discovery task, results are shown in Table 3.4 and Figure 3.9:

- Again, the mean rank of PEPA and A-PEPA are significantly better than other methods. As the number of motifs is known, PEPA performs better than A-PEPA. Therefore, if a good calibration of PEPA is possible, it should be preferred over the adaptive scheme.
- Overall, f1-scores are lower on the motif discovery task because pattern occurrences must be classified, not just localized.
- Unlike A-PEPA, MDLC performances drop significantly from occurrence detection to motif set discovery. MDLC groups detected occurrences in too many sub-clusters, whereas A-PEPA better estimates the number of motifs as depicted in Section 3.5.3.

3.5.3 Influence of the parameters

In this section, we evaluate the influence of three parameters: the window length, the number of neighbors, and the persistence threshold heuristic in A-PEPA.

Influence of the window length. Our approach is tested on dataset (S-3), where all patterns have the same length. We run each algorithm with the window length parameter ranging from 50% to 150% of the pattern length. All other parameters are identical to those defined previously. For VALMOD and MDLC, the minimum/maximum window lengths are 50%/150%.

The results are shown in Figure 3.10. PEPA performs better than other methods for all metrics and most window lengths. Its f1-score is stable and close to the maximum for window lengths between 80% and 100%, proving its robustness to the window length parameter. A-PEPA has the best precision, but its recall drops as the number of motifs tends to get overestimated. However, its f1-score remains high and similar to PEPA, showing its robustness to the window length parameter. In light of this result, we recommend underestimating the length of the true patterns when using PEPA and A-PEPA.

Influence of the number of neighbors. Setting the number of neighbors is mandatory to compute the K-nearest neighbor graph (Section 3.3.1). Theoretically, a larger number of neighbors leads to better performance but higher computation time and storage. In this experiment, we measured the relative errors between f1-scores obtained for different numbers of neighbors and the f1-scores obtained with the whole graph on all datasets. The

Table 3.3 Occurrence detection. SF: SetFinder, GM: Grammarviz, LM: LatentMotif, SM: STOMP, VM: VALMOD, MC: MDLC

dataset	algorithm / metric	SF	GM	LM	SM	VM	MC	PEPA	A-PEPA
(S-2) single	f1-score	**0.98**	0.07	0.27	0.71	0.86	0.85	0.96	0.97
	precision	**0.99**	0.69	0.59	0.98	0.97	0.81	0.96	0.98
	recall	**0.97**	0.04	0.19	0.58	0.78	0.89	0.95	0.96
(S-3) fixed	f1-score	0.49	0.20	0.50	0.82	0.82	0.66	**0.90**	0.89
	precision	0.59	0.57	0.68	0.82	0.78	0.60	0.94	**0.94**
	recall	0.47	0.13	0.41	0.84	0.85	0.74	**0.87**	0.85
(S-4) variable	f1-score	0.49	0.02	0.48	0.86	0.76	0.76	**0.95**	0.95
	precision	0.81	0.12	0.74	0.87	0.72	0.75	0.97	**0.97**
	recall	0.37	0.01	0.37	0.86	0.83	0.78	**0.94**	0.93
(R-1) mitdb-1	f1-score	0.34	0.01	0.11	0.58	0.67	**0.77**	0.71	0.75
	precision	0.78	0.20	0.96	**0.97**	0.95	0.91	0.91	0.92
	recall	0.28	0.01	0.06	0.46	0.64	**0.68**	0.60	0.67
(R-2) mitdb-2	f1-score	0.64	0.03	0.30	0.58	0.68	0.67	0.86	**0.87**
	precision	0.93	0.45	**0.97**	0.97	0.97	0.86	0.94	0.95
	recall	0.53	0.02	0.19	0.44	0.59	0.55	0.80	**0.80**
(R-3) mitdb800	f1-score	0.75	0.13	0.40	0.56	0.70	0.49	**0.89**	0.89
	precision	0.90	0.96	0.87	0.97	**0.98**	0.92	0.96	0.97
	recall	0.67	0.07	0.28	0.43	0.59	0.34	**0.84**	0.84
(R-4) ptt-ppg	f1-score	0.49	0.01	0.12	0.49	0.52	0.71	0.73	**0.75**
	precision	0.91	0.11	0.92	**0.97**	0.87	0.85	0.96	0.97
	recall	0.38	0.00	0.07	0.36	0.43	0.61	0.62	**0.66**
(R-5) refit	f1-score	0.07	0.14	0.12	0.00	0.01	0.07	**0.31**	0.29
	precision	0.06	0.21	0.17	0.00	0.01	0.05	**0.23**	0.22
	recall	0.17	0.19	0.11	0.02	0.04	0.14	**0.56**	0.51
(R-6) arm-coda	f1-score	0.25	0.00	0.44	0.51	0.28	0.54	**0.62**	0.59
	precision	0.24	0.02	0.54	0.47	0.45	0.50	**0.61**	0.59
	recall	0.38	0.01	0.41	0.58	0.28	0.63	**0.66**	0.62

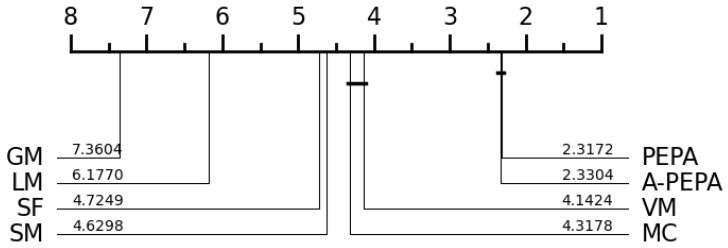


Figure 3.8 Occurrence detection critical difference diagram. The rank is based on the f1-score. PEPA and A-PEPA perform significantly better than any other algorithm. Performances between PEPA and A-PEPA are not significantly different.

number of neighbors ranges from 1 to 15 for PEPA and A-PEPA, and other parameters remain identical.

Results shown in Figure 3.11 prove that the number of neighbors has little influence on the performance of PEPA and A-PEPA. Regardless of the algorithm and for more than 5 neighbors, the average relative error does not exceed 1%, the average error is less than 0.05%, and the standard deviation is less than 0.2%. In practice, we recommend setting the number of neighbors to 5; it leads to good performance while maintaining low

Table 3.4 Motif set Discovery. SF: SetFinder, GM: Grammarviz, LM: LatentMotif, SM: STOMP, VM: VALMOD, MC: MDLC

dataset	algorithm / metric	SF	GM	LM	SM	VM	MC	PEPA	A-PEPA
(S-2) single	f1-score	**0.98**	0.07	0.27	0.71	0.86	0.34	0.96	0.84
	precision	**0.99**	0.69	0.59	0.98	0.97	0.99	0.96	0.98
	recall	**0.97**	0.04	0.19	0.58	0.78	0.21	0.95	0.78
(S-3) fixed	f1-score	0.34	0.14	0.41	0.66	0.55	0.61	**0.84**	0.81
	precision	0.28	0.20	0.45	0.71	0.54	0.70	**0.86**	0.85
	recall	0.45	0.11	0.41	0.67	0.63	0.57	**0.85**	0.81
(S-4) variable	f1-score	0.32	0.02	0.39	0.72	0.43	0.75	**0.80**	0.80
	precision	0.31	0.02	0.49	0.74	0.42	**0.86**	0.81	0.82
	recall	0.34	0.01	0.37	0.75	0.54	0.71	**0.82**	0.81
(R-1) mitdb-1	f1-score	0.34	0.01	0.11	0.58	0.67	0.20	**0.71**	0.45
	precision	0.78	0.20	0.96	0.97	0.95	**0.98**	0.91	0.96
	recall	0.28	0.01	0.06	0.46	**0.64**	0.12	0.60	0.31
(R-2) mitdb-2	f1-score	0.49	0.02	0.24	0.41	0.51	0.31	**0.68**	0.59
	precision	0.66	0.31	0.73	0.73	0.77	**0.82**	0.75	0.78
	recall	0.44	0.01	0.17	0.32	0.47	0.24	**0.65**	0.53
(R-3) mitdb800	f1-score	0.35	0.06	0.23	0.25	0.33	0.08	**0.46**	0.41
	precision	0.42	0.53	0.49	0.51	0.52	**0.71**	0.50	0.53
	recall	0.34	0.04	0.19	0.22	0.31	0.05	**0.45**	0.38
(R-4) ptt-ppg	f1-score	0.49	0.01	0.12	0.49	0.52	0.19	**0.73**	0.50
	precision	0.91	0.11	0.92	0.97	0.87	**0.97**	0.96	0.96
	recall	0.38	0.00	0.07	0.36	0.43	0.11	**0.62**	0.37
(R-5) refit	f1-score	0.08	0.10	0.10	0.00	0.01	0.07	0.17	**0.20**
	precision	0.07	**0.20**	0.13	0.00	0.01	0.14	0.14	0.18
	recall	0.16	0.16	0.09	0.02	0.04	0.06	**0.35**	0.33
(R-6) arm-coda	f1-score	0.28	0.00	0.39	0.30	0.14	**0.53**	0.32	0.32
	precision	0.21	0.00	0.40	0.31	0.18	**0.55**	0.30	0.31
	recall	0.54	0.00	0.44	0.41	0.19	**0.63**	0.45	0.45

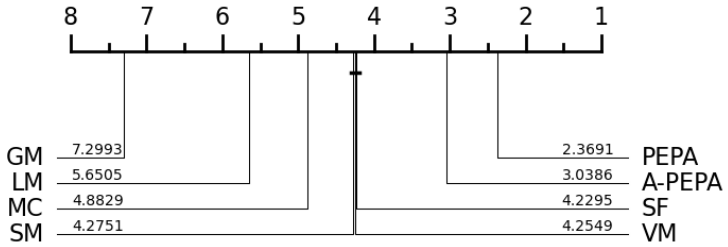


Figure 3.9 Motif set discovery critical difference diagram. The rank is based on the f1-score. PEPA performs significantly better than any other algorithm. The second best performer is A-PEPA. Other algorithms perform significantly worse.

computational time and storage.

Influence of the persistence threshold heuristic in A-PEPA. In this experiment, we evaluate the ability of A-PEPA to detect the exact number of motif sets, and we compare its performances with the other adaptive method, MDLC, by measuring the error between the real and predicted number of motif sets on all datasets presented in

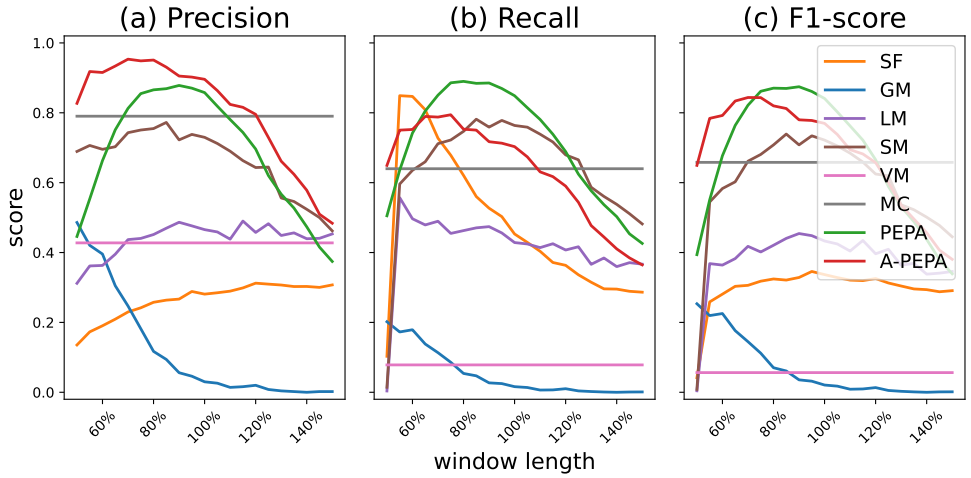


Figure 3.10 Precision, recall, and f1-score of all algorithms as a function of window length. The experiment is run on the fixed dataset, and the window length is expressed as a percentage of the pattern length. The performance in f1-score of PEPA and A-PEPA is similar. They outperform all other algorithms in their best configuration over a wide range of window lengths.

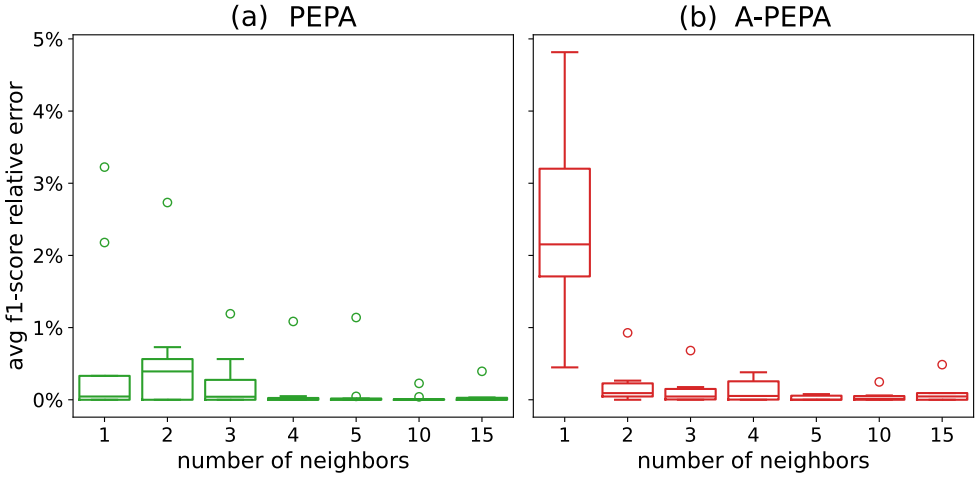


Figure 3.11 Average f1-score relative error per dataset as a function of the number of neighbors for PEPA and A-PEPA and on all datasets. The baselines correspond to the scores obtained on the whole graph. For both algorithms, the relative error is less than 1% for more than 5 neighbors and never exceeds 9%.

Section 3.4.1. We set the persistence threshold heuristic of A-PEPA on the second-largest gap. It enforces A-PEPA to consider variations of the most persistent subgraphs as potential motif sets. The results are shown in Figure 3.12.

On all datasets except arm-coda (R-6), A-PEPA better estimates the number of motifs compared to MDLC. The average absolute mean error is 2.1 for A-PEPA and 12.0 for MDLC, with a standard deviation of 2.4 vs 10.8. It reflects the performance drop

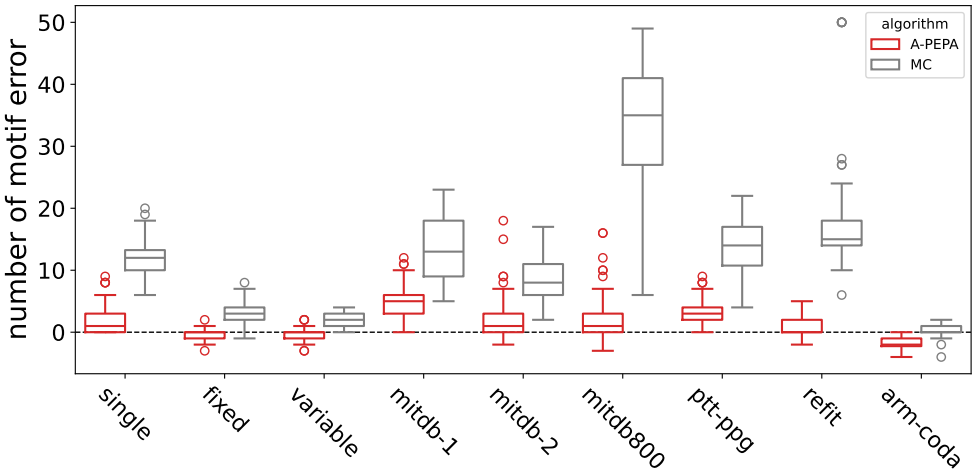


Figure 3.12 Boxplots of the error in estimating the number of motif sets with A-PEPA for all datasets.

observed in Section 3.5.2 when MDLC clusters detected occurrences, whereas A-PEPA better retrieves the number of motif sets thanks to the persistent diagram.

Congruently with its heuristic settings (second most persistent gap), A-PEPA overestimates the number of motifs on single motif datasets: single (S-2), mitdb-1 (R-1), ptt-ppg (R-4). The number of motif sets is also greatly overestimated in rare cases for mitdb-2 (R-2), mitdb800 (R-3); the second gap leads to the inclusion of many sub-motif sets. However, the estimation is more accurate for datasets with a larger number of motif sets and shows less variability.

In practice, we recommend setting the persistence threshold heuristic to the largest or second largest gap. Alternatively, the relevance of the threshold can be verified with the persistence diagram, and recomputing of motif sets is done in $\mathcal{O}(n \log(n))$ in the worst case according to Section 3.3.5.

3.5.4 Scalability

In this experiment, we evaluated the scalability with the time series length of PEPA and state-of-the-art algorithms.

Experiment presentation. We evaluated the algorithm runtime on a dataset consisting of synthetic time series of the following lengths: 10K, 50K, 100K, 500K, and 1M. We generated 10 time series for each length following the fixed scenario (S-3) and only modified the space between successive occurrences. We did not consider Grammarviz, since it is implemented in Java, and VALMOD, since we implemented a greedy version that does not consider its pruning strategy. Nevertheless, the performance of STOMP is a lower bound of the performance of VALMOD since it runs STOMP as an initialization. We considered the algorithms SetFinder, LatentMotif, STOMP, MDLC, PEPA, and A-PEPA.

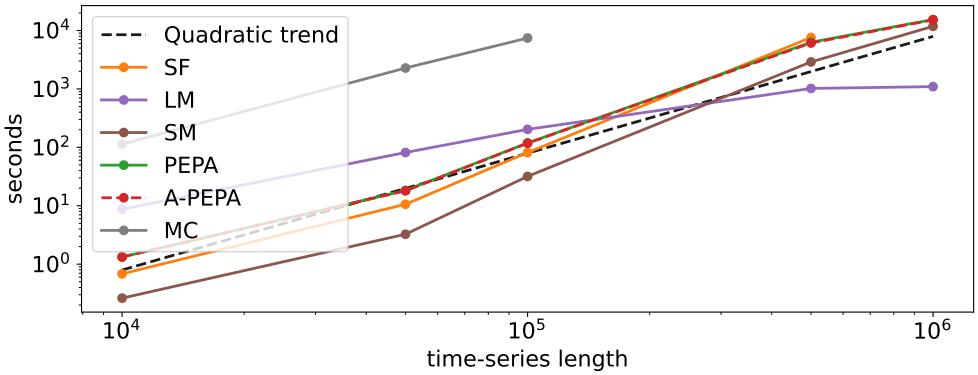


Figure 3.13 Algorithms scalability with the length of the time series. SF: SetFinder, LM: LatentMotif, SM: STOMP.

The parameters of each algorithm were identical to those defined in the benchmark for the fixed dataset. The timeout was set to 24 hours.

Results. The average runtimes per length are shown in Figure 3.13. MDLC is the worst time-performing algorithm and times out after 100K. SetFinder does not scale with the length of the time series and times out after 500K. On the other hand, LatentMotif is the fastest for large lengths (>300K), but it is slow for small lengths (<50K). This trend is due to the optimization scheme that limits the number of distance profiles computed. PEPA A-PEPA and STOMP scale according to their quadratic time. PEPA and A-PEPA performances are identical. STOMP is slightly faster than PEPA, but its advantage decreases as the length of the time series increases. Indeed, STOMP has to recompute some distance profiles to create the motif sets, while PEPA creates motif sets from the precomputed graph.

3.6 An application for interactive motif discovery

The PEPA algorithm can be naturally declined in an interactive and user-friendly application thanks to three characteristics:

- **A visual interpretation of the time series:** Once the graph of a time series is built, it is easy to set the parameters that define the motif sets, namely the birth and persistence threshold, thanks to the visual interpretation of the persistence diagram (see Section 3.3.2).
- **A fast computation of the motif sets:** Once the thresholds are defined, the time to compute the motif sets is linear in the length of the time series (see Section 3.3.5). As a result, adjusting the thresholds results in near-instantaneous updates of the motif sets, enabling real-time interactivity between the user and the algorithm.
- **Heuristics to adjust the parameters:** Once the graph is built, 4 parameters can be modified, and we have proposed heuristics to set them. On the persistence dia-

gram, the birth threshold can be set with the Otsu heuristics (see Section 3.3.2), and the persistence threshold can be set from the jumps in persistence (see Section 3.3.4). However, motifs can be sensitive to these thresholds, but their adjustment can be facilitated by modifying the (α, β)-rectification of the distance. A heuristic for setting the rectification and its implication in the PEPA algorithm is discussed in the following paragraph.

Setting the (α, β)-rectification. Sometimes, points on the persistence diagram cluster around the bottom-left corner, indicating that the distance is overly permissive, treating all subsequences as highly similar. Conversely, when points concentrate in the top-right corner, the distance is too restrictive, classifying subsequences as highly dissimilar. In these extreme cases, the motif sets become sensitive to the birth and persistence thresholds, as the points are confined to a limited region of the persistent diagram. There is a need to expand the decision area by spreading the points on the persistence diagram.

The (α, β)-rectification provides a solution to that issue (see Section 3.3.1). Precisely, the rectification adjusts the distance permissiveness and restrictiveness. With a judicious choice of the (α, β) parameters, the rectification counteracts the concentration effect by spreading the points on the persistence diagram. It is important to note that the rectification function is strictly increasing for any choice of (α, β) parameters. As a result, changes in these parameters do not alter the structure of the connected subgraphs tracked during filtration; only their birth and death dates are affected. Modifying the rectification implies only changing the weights of the minimum spanning tree extracted during the filtration. Similar to the birth and persistence thresholds, the time to compute the modifications induced by the change of rectification is linear in the length of the time series.

To facilitate the distribution of points on the persistence diagram, we propose a heuristic for selecting the parameters α and β. It involves solving the following optimization problem:

$$(\alpha^*, \beta^*) = \underset{(\alpha,\beta)\in\mathbb{R}_+^*\times[0,2]}{\operatorname{argmin}} D_{C.S}(B_{\alpha,\beta}, U_{[0,2]}) \tag{3.4}$$

where $D_{C.S}$ is the Cauchy Schwartz divergence [KHP11], $B_{\alpha,\beta}$ is the Gaussian kernel densities associated to the (α, β)-rectified birth dates, and $U_{[0,2]}$ is the densities of the uniform distribution on $[0, 2]$. The goal is to find the parameters that uniformly spread the birth of the connected subgraphs along the interval $[0, 2]$. The convexity of the problem is not guaranteed, and we perform a non-exhaustive grid search to find suitable parameters. This heuristic is computationally efficient for short time series, but the search process can become time-consuming for longer ones. Enhancements to the optimization procedure could further improve its performance in such cases.

The operating system. The system is implemented in Python with the Dash library. It is accessible from a webpage[2] or can be run locally[3]. We also provide a demonstration

[2]Webpage: `https://persistent-pattern-discovery.onrender.com`
[3]Github: `https://github.com/thibaut-germain/Persistent_Pattern_Discovery_App`

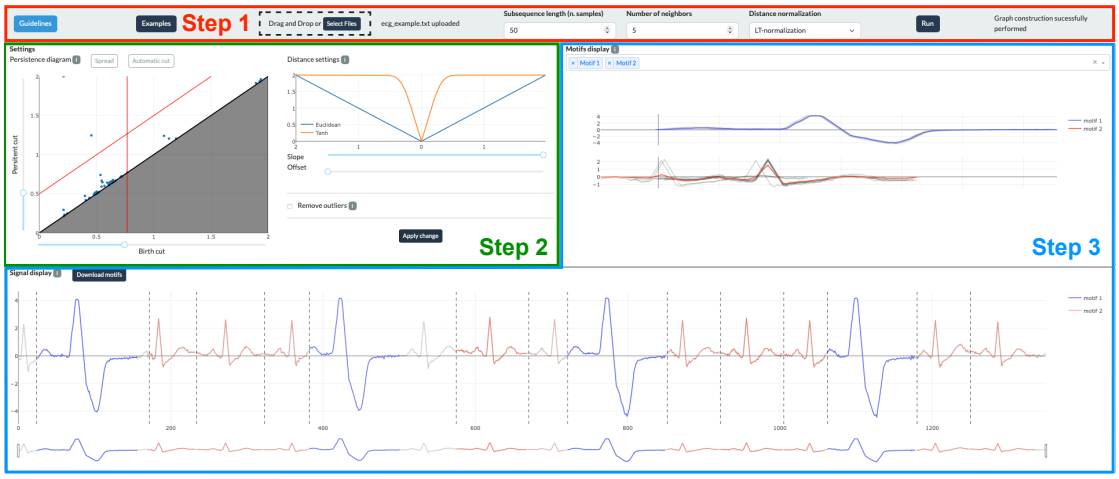


Figure 3.14 The application interface: the user has discovered normal and abnormal heartbeats in an ECG.

video[4].

The user interface follows the workflow of the PEPA algorithm (see Figure 3.3), and it is divided into three blocks as depicted in Figure 3.14:

- **Upper block (red):** Associated with the "From time series to graph" step, the user uploads a time series, sets parameters related to the graph construction, and runs it.
- **Middle left block (green):** Associated with the "Graph clustering with persistent homology" step, it is the core interactive component of the system. The user can modify the distance function and set the thresholds from the resulting persistence diagram to his wish.
- **Middle right & lower blocks (blue):** Associated with the "From clusters to motif sets" step, the lower block displays the time series and highlights the discovered motifs. The middle-right block displays motifs individually.

After step 1, the system stores the time series, the graph, and the persistence diagram. These elements allow smooth back and forth between steps 2 and 3, providing a playground for deepening the user's knowledge about the time series. We also provide detailed guidelines and information in the system itself.

An use case: the detection of abnormal heartbeats in an ECG. The MIT-BIH dataset compiles ECGs from patients experiencing premature ventricular contractions (PVCs). In Figure 3.14, we explore a 16-second recording segment. The signal is displayed as soon as it is uploaded. Then, after computing the graph, the user optimizes the visualization of the persistence diagram by automatically setting the (α, β)-rectification with the `Spread` button, which runs the heuristic defined in Equation (3.4). The persistence

[4]Demonstration video: `https://youtu.be/F2bwCKiR-i8`

diagram suggests two motifs and a clear distinction between the motifs and the irrelevant part of the signals. By clicking on the `Automatic cut` button, the user runs the heuristics that set the persistence and the birth thresholds. After applying the changes, the motifs are displayed; the red motif corresponds to the normal heartbeats, while the blue motif corresponds to the abnormal ones, which account for the PVCs. Thanks to the system's responsiveness, the user can adjust the rectifications and the thresholds to define the motif sets accurately.

Conclusion. In this chapter, we have presented a novel motif discovery algorithm, PEPA, that overcomes the limitations of prior approaches by leveraging persistent homology. Unlike traditional methods, we abandoned the restriction of motif sets to be included within balls whose radius is based on a predefined similarity threshold. Instead, we leverage persistent homology to track the sets of subsequences that remain consistent across a wide range of similarity thresholds. These stable, corresponding to motif sets, are easily identifiable from the persistent diagram, a visual representation of the time series. We also proposed an A-PEPA, an adaptive version of the algorithm that infers the number of motif sets to discover from the persistent diagram. The experiments demonstrate that both algorithms significantly outperform state-of-the-art algorithms while maintaining comparable computational complexity. We also leveraged the algorithm's interpretability and efficiency to create an interactive and user-friendly application dedicated to motif discovery.

Part II

Global scale tasks
&
Elastic deformations

Chapter 4

Mice ventilation analysis and its application to the study of the cholinergic system

Key points:

1. This chapter provides the biological context that motivates the development of the methods discussed in the subsequent chapters.

2. Mouse respiration can be monitored using plethysmography, with the resulting signals reflecting the mouse's physiological state. Analyzing such signals is valuable in various settings, such as studying the effects of drugs on the respiratory system, monitoring mouse models of human diseases, or assessing airway irritants. This thesis analyzes the impact of inhibiting an enzyme involved in the respiratory control through drug exposure across different mouse genotypes.

3. However, current approaches primarily focus on critical descriptors such as duration or inhaled volume, neglecting the full dynamics captured by the shape of the respiratory cycle, highlighting the need for shape-based analysis methods.

Contributions:

1. This chapter introduces a new algorithm for segmenting mice respiratory cycles (inspiration and expiration) from plethysmography signals. By incorporating physiological constraints, the method accurately detects the start of inspiration and expiration, offering greater robustness to respiratory variations compared to previous approaches.

Contents

This chapter introduces the applicative context that inspired the content of the following two chapters. Rather than serving as a comprehensive state-of-the-art review, it aims to provide an introduction to the biological context, offering a clearer understanding of the proposed methodological tools. The primary objective of the application is to study mice ventilation and its changes when exposed to toxic molecules. For clarity, the physiological mechanisms involved in breathing are simplified, focusing only on the functions relevant to the study of mice respiration.

4.1 Analyzing mice ventilation from plethysmography signals

Respiration is a fundamental physiological function that ensures the vital supply of O2 during inspiration and the elimination of CO2 during expiration. However, breathing can vary depending on an organism's physiological and environmental context. For instance, the breathing rate increases during prolonged physical effort to meet the body's O2 demands. The body also protects the lungs against potentially toxic invaders through various protective mechanisms. For example, the inhalation of chili peppers triggers reflexes, or more appropriately, different reflexes are triggered by different stimuli: some cause coughing, others sneezing, bronchoconstriction, and so on. Assessing ventilation and its alterations in response to physiological or environmental changes in conscious, spontaneously breathing animals is crucial in various applications, such as studying the effects of drugs on the respiratory system [Mur02], monitoring mouse models of human diseases [Wil+17], or evaluating airway irritants [Vij+93]. Ventilation is typically accessed from the respiratory airflow signal recorded with plethysmography techniques.

4.1.1 Plethysmography

For mice, several plethysmographs exist: whole-body (WBP, [BT70]), dual-chamber (DCP, [Hoy12]), head out-of-body (HOP, [Vij+93]). All plethysmographs have essentially the same functioning; a mouse is placed within an airtight box, and its breathing induces air volume changes that are recorded using pneumotachographs or pressure transducers, which convert them into airflow or a pressure signal. The choice between them is based on a trade-off between invasiveness and accuracy of measurement [BI03].

For WBP, the mouse is placed in near-natural conditions (the mouse is neither anesthetized nor constrained in its movements): it is a small box where the mouse

is not restrained and can move freely. The pressure difference between the box and the atmosphere is measured over time, reflecting changes in volume, humidity, and temperature of the air entering and leaving the mouse's lung. However, the ventilation function is poorly measured due to the artifact induced by the mouse movements, and the experiment reproducibility depends on many environmental parameters [BI03; Bru+22].

The DCP consists of two sealed compartments, with the animal's head in one compartment and its body in the other. The mouse is not anesthetized but it is constrained in a tube with the nose pointing into the nasal compartment (respiration is primarily nasal in mice) [Mai+18]. This device allows for independent monitoring of the nasal airflow and the airflow caused by the thoracic movements of an animal. DCP is a relevant approach for assessing the ventilatory mechanics of the respiratory system, and it provides information on ventilatory and lung function [Hoy12]. As the mouse is constrained, the duration of respiration recording is limited to less than one hour. As an alternative, the HOP uses only the thoracic compartment, imposing less constraint on the mouse [Vij+93]. Recordings from DCP and HOP directly reflect the air inhaled and exhaled during respiration. These methods have been used for several decades to monitor changes in mouse respiration caused by airborne chemicals on the airways [Vij+93] and have been improved to limit air leakage from collar [Bru+22]. Figure 4.1 describes the DCP functioning and the mouse respiratory system while Figure 4.2 presents a real example of nasal airflow signal. Thoracic airflow looks similar to nasal airflow, and by convention, a positive nasal airflow indicates an inspiration, and a negative nasal airflow indicates an expiration.

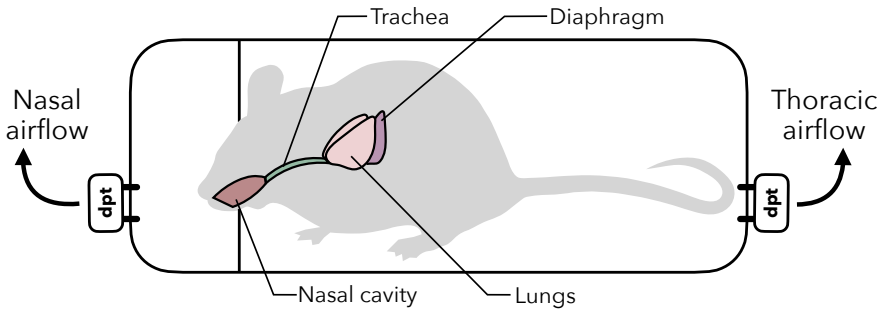


Figure 4.1 Double chamber plethysmograph (DCP) and mouse respiratory system. *dpt* stands for differential pressure transducer which measures the pressure then converted in airflow. DCP measures the nasal airflow (airflow coming in and out the nasal cavity), and the thoracic airflow (changes in the volume occupied by the chest cage).

4.1.2 Inferring ventilation modalities from airflows.

Respiratory cycle, the ventilation atom. Breathing consists of a succession of respiratory cycles, each composed of an inspiration followed by an expiration. Inspiration is an active phase in which the diaphragm muscle contracts, expanding the chest cavity and creating negative pressure causing the lungs to fill with air. Expiration is a passive phase in which the diaphragm and chest muscles relax, reducing lung space, which expels

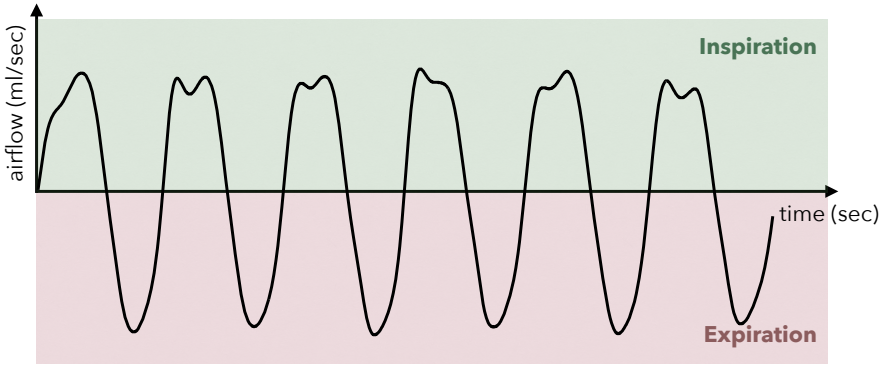


Figure 4.2 Nasal airflow. During inspiration, the airflow is positive (green) and during expiration, the airflow is negative (red).

CO2-rich air by pressure difference. Inspiration and expiration are the fundamental mechanisms that cannot cease to ensure the survival of an organism through the exchange of O2 and CO2. All other ventilation modalities, such as vocalization, coughing or sneezing, are performed within the framework of a respiratory cycle. Regardless of the purpose for mobilizing the respiratory system, the process is coordinated by the nervous system, which synchronizes the contraction and relaxation of various muscles. Plethysmography signals capture the airflow that results from coordinated movement produced by different muscles, and thus analyzing respiratory cycles and their evolution from these signals can provide insights into both the underlying cause for activating the respiratory system and how the muscles are mobilized and controlled by the nervous system.

A global and local scale problem. Assuming an algorithm to segment a plethysmography signal respiratory cycles, the analysis of ventilation can be broken down into two tasks: embedding respiratory cycles and studying the dynamic of the embedded sequences. From a time series perspective, the embedding process must simplify the complexity of the time series while preserving its physiological relevance, facilitating the evaluation and interpretation of ventilation dynamics. While dynamic analysis focuses on local events, embedding can be considered a global scale task that can be dealt with statistical algorithms. Figure 4.3 illustrates the divisions of the ventilation analysis in two tasks. The following two chapters primarily focus on embedding respiratory cycles, and we also present basic tools for analyzing the dynamics of the embedded sequences. The development of more statistically robust methods for dynamic analysis is left for future work.

Limitations of current ventilation quantifiers. Current methods for describing ventilation typically rely on aggregating descriptors of respiratory cycles over time. Indeed, from airflow signals, common descriptors of the respiratory cycle include the inspiration/expiration duration or the air volume inhaled/exhaled. These descriptors are derived from detecting remarkable points like inspiration starts or ends. However, current detection algorithms are sensitive to noise and breathing alterations, making point detection unreliable in extreme cases. More importantly, they only reveal part of the

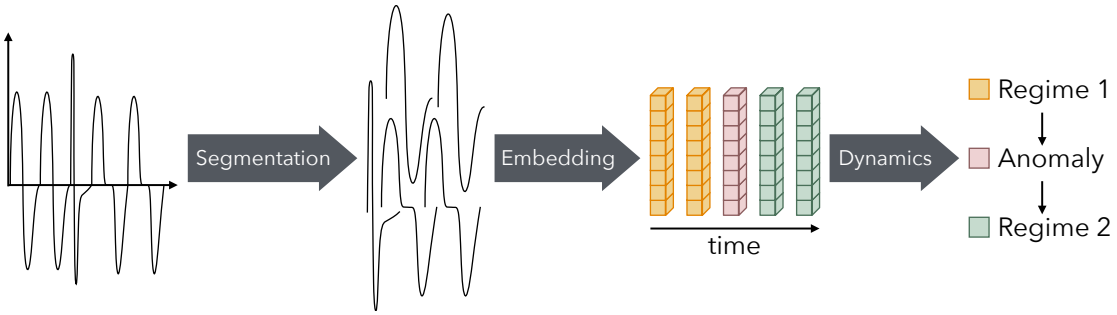


Figure 4.3 General workflow for mice ventilation analysis

information contained in respiratory airflows, leading to a poor description of respiratory cycles. Finally, these descriptors are usually average through time, leading to features like respiratory frequency of average volume inhaled/exhaled to describe ventilation. While essential to quantify respiratory exchange, these features result in smooth representations of airflow signals that cannot represent the heterogeneous and fast-changing ventilation dynamic.

A sensitive respiratory cycle segmentation algorithm. EMKA offers high performance plethysmography systems. The biologists using this equipment analyze the recorded data with the IOX2 software [Mai+18]. It benefits from a user-friendly interface and offers numerous tools to interact and explore plethysmography signals. Notably, the software enables the computation of various respiratory cycle descriptors, including those previously mentioned. These descriptors' computation relies on an algorithm that segments the respiratory cycles from an airflow signal. While effective under many conditions, this algorithm shows limitations when mice present severe respiratory alterations. Indeed, the inspiration and expiration starts are inferred using tangential information and a notable value in the airflow: the inspiratory flow peak. However, the position of the peak may vary depending on the presence of pauses in the cycle, or even disappear in case of bronchoconstriction while the cycle limits may remain unchanged. As detailed in the next section, in the experimental application, mice are exposed to toxic molecules and exhibit significant respiratory cycle disruptions, highlighting the need for a more robust segmentation algorithm.

Improving ventilation description. The repertoire of ventilation modalities is varied, and it notably includes respiratory reflexes whose role is to protect the respiratory system, i.e., the lungs and the airways, from potentially harmful substances. Reflexes are selected to either expel irritant molecules from the respiratory system or block their entry. These reflexes are generally automatic and beyond conscious control. For instance, swallowing is a reflex that closes the upper airways to prevent food from entering the airways. Exposure to irritant molecules can also trigger reflexes such as sneezing and coughing, which help clear irritants, or bronchoconstriction and apnea, which prevent further inhalation of harmful substances.

Respiratory reflexes have been well studied; some are even visible in plethysmography signals. Proposing embedding of respiratory cycles that reflect characteristic events like reflexes would be meaningful in describing ventilation. Unfortunately, common respiratory cycle descriptors fail to account for some reflexes. For instance, some provoke pauses during a respiratory cycle, and their physiological meaning differs depending on the pause location. Indeed, a pause before inspiration suggests that a mouse has difficulties activating ventilatory muscles, while a pause after inspiration suggests difficulties relaxing the ventilatory muscles. As depicted in Figure 4.4, both reflexes cannot be differentiated solely by duration and volume. The differentiation is made possible by including additional descriptors that somehow encode the shape difference between reflexes.

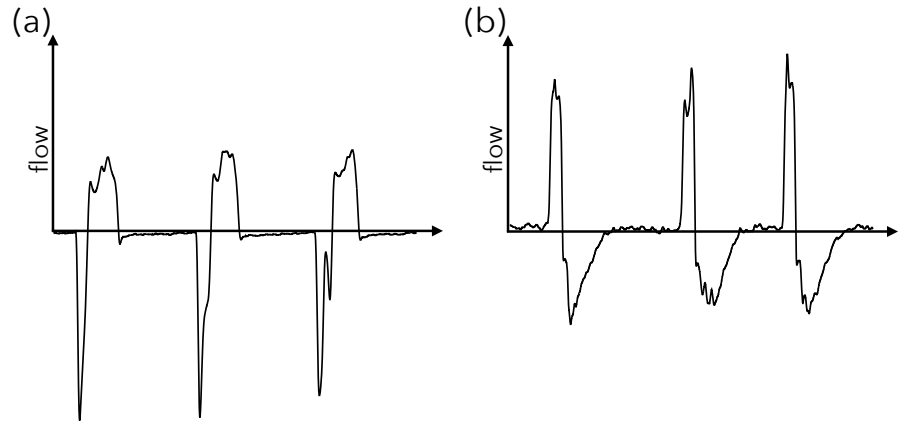


Figure 4.4 Illustrations of two mice ventilation modalities carrying different physiological meanings. **(a)** A pause appears in the respiration after inspiration, meaning that the mouse has difficulties relaxing ventilatory muscles. **(b)** A pause appears in the respiration after expiration, meaning that the mouse has difficulties activating ventilatory muscles. However, simple descriptors like cycle duration or volume inhaled/exhaled cannot differentiate the two modalities.

Toward shape-based unsupervised methods. In a simple case, constructing an embedding that pairs respiratory cycles to characteristic ventilation modalities like reflexes could be framed as a classification problem. However, there are more suitable approaches, as classification requires the creation of a labeled dataset, which would be both time-consuming (approximately 12,000 respiratory cycles to annotate for an hour of recording) and require a good understanding of the data. However, as illustrated earlier, the shape of respiratory cycles offers valuable insights into the underlying ventilation modality. Shape-based unsupervised methods are a more appealing alternative approach for creating embedding directly from a respiratory cycle dataset, provided the notion of shape is encoded correctly. However, unsupervised algorithms are governed by geometrical or statistical principles and are often disconnected from the applicative context. Ensuring an ongoing dialogue between the data and the biological context is crucial for establishing meaningful correspondences.

For example, a recent clustering method attempted to classify respiratory cycle patterns from airflows recorded in a WBP [SF21]. They used principal component

analysis and a hierarchical clustering algorithm to identify groups of common respiratory cycle patterns. The groups reveal shape variations that are not distinguishable from standard features like respiratory frequency or inhaled/exhaled volume. While these groups held physiological significance, enabling the tracking of critical changes over time, they could not be directly linked to specific respiratory activities because WBP signals combine multiple parameters (volume, temperature, and humidity).

Scope of the next chapters. The next chapter introduces an embedding technique based on a time series clustering algorithm. This fast method comes with several visualizations that provide insights about common ventilation modalities and a general understanding of the experiment. The subsequent chapter presents an embedding technique that leverages shape analysis algorithms. This method maps respiratory cycles to vectors that encode shape information, enabling the application of statistical methods to study ventilation dynamics. However, both embedding techniques depend on a reliable segmentation of respiratory cycles, which is discussed in the following section.

4.1.3 Segmenting respiratory cycles

A plethysmograph records the breathing airflows from air volume changes within an airtight chamber. We take the convention that the breathing airflows are positive during inspiration and negative during expiration.

Current approaches. Intuitively, inspiration and expiration starts correspond to zeros of a breathing airflow signal. Current segmentation algorithms take this convention. However, the signal may suffer from noise or respiratory alterations that blur inspiration and expiration starts. To overcome this issue, two heuristics are commonly used to estimate starts. The first one defines an inspiration (resp. expiration) start as the last (resp. first) zeros before (resp. after) the airflow reaches a user-defined threshold [SF21]. The second heuristic, used by emka [Mai+18], defines inspiration and expiration starts as the zeros-crossing values of tangents computed between some critical points (essentially percentages of the maximum flow during inspiration).

Both heuristics are sensitive to outliers, often present when severe alterations occur, and may fail in some situations, like when the inspirations or expirations happen in several steps. Figure 4.6(a,b) illustrate heuristics sensitivity on a real plethysmography signal with severe breathing alterations. Both heuristics fail to correctly estimate several inspiration starts because of noise and apnea during inspirations.

An approach based on physiological constraints. Previous heuristics show that setting inspiration and expiration starts from a breathing airflow is difficult and approximate. However, a natural and robust definition of inspiration and expiration starts is possible by considering the lungs' inflating state obtained by robust integration of the breathing airflow $S \in \mathbb{R}^n$:

$$v_t = \sum_{i=1}^{t} s_i - \hat{a}t + \hat{b}, \quad \forall t \in [\![1, n]\!] \tag{4.1}$$

where $t \in [0, n] \mapsto \hat{a}t + \hat{b} \in \mathbb{R}$, is the linear approximation of S. Indeed, a linear trend often appears during the integration process, and it is sensibly due to volume, temperature, or humidity changes inside the chamber caused by breathing. During inspiration, the lungs inflate and reach a maximum relative volume at the end of the inspiration by physical constraints. Conversely, the lungs have a minimal relative volume at the end of an expiration. Therefore, inspiration starts can be identified as the local minima of the lungs' volume signal, and expiration starts at its local maxima. From an algorithmic perspective, to ensure an alternation between inspiration and expiration, the segmentation algorithm first searches for all local minima with a peak-searching procedure based on prominence and then searches for the maximum between two consecutive local minima.

Specifically, the peak detection algorithm consists in finding all local maxima and removing those whose prominence is below a prominence threshold. Considering a time series $\mathbf{x} \in \mathbb{R}^n$, a data point x_t is considered as a local maximum if $x_t > x_{t-1}$ and $x_t > x_{t+1}$. Given a user-defined window length w_{len}, the prominence of a local maximum x_u is computed as follows:

1. **Finding left prominence:** Denote s_u the last time point such as the sequence $(x_{l_u}, ..., x_{u-1})$, where $l_u = \max(0, u - \lfloor w_{len}/2 \rfloor)$, intersects the horizontal line $y = x_u$. If there is no intersection, $s_u = l_u$. Left prominence is defined as: $pl_u = x_u - \min(x_{s_u}, ..., x_{u-1})$.

2. **Finding right prominence:** Denote e_u the first time point such as the sequence $(x_{u+1}, ..., x_{r_u})$, where $r_u = \min(T, u + \lfloor w_{len}/2 \rfloor)$, intersects the horizontal line $y = x_u$. If there is no intersection, $e_u = r_u$. Right prominence is defined as: $pr_u = x_u - \min(x_{u+1}, ..., x_{e_u})$.

3. **Set prominence:** Prominence of the local maximum x_u is defined as $p_u = \max(pl_u, pr_u)$

Figure 4.5 illustrates the prominence computation. Considering a minimum prominence p_{min}, only local maxima with a prominence greater than p_{min} are considered as peaks. This method is implemented in Python Scipy package[1].

Algorithm 10 Computing inspiration & expiration starts

Require: S a time series, w_{len} peak search window length , p_{min} minimum prominence
1: $V \leftarrow ComputeVolume(S)$ ▷ *See Equation* (4.1)
2: $insp\_start \leftarrow SeachPeak(-V, w_{len}, p_{min})$
3: $K \leftarrow length(insp\_starts)$
4: $exp\_start \leftarrow ()$
5: **for** k=1,...,K-1 **do**
6: $\quad s \leftarrow insp\_start[k], e \leftarrow insp\_start[k+1]$
7: $\quad$ exp $\_start.append(argmax(S[s:e]) + s)$
8: **return** $insp\_start, exp\_start$

[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html

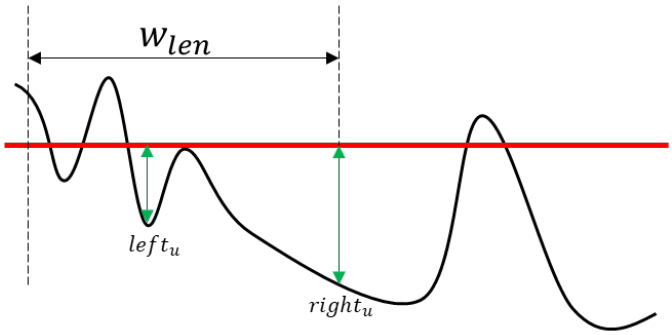


Figure 4.5 The prominence is $left_u$. There is no intersection between the horizontal line and the curve on the right side, the right search space is bounded by the user defined window size.

Algorithm 10 describes the procedure for detecting inspiration and expiration starts. Once detected, the plethysmography signal can be segmented into a dataset of respiratory cycles or a pair of datasets, one for inspiration and one for expiration. Figure 4.6c illustrates the segmentation on a real plethysmography signal with severe breathing alterations. Compared to other heuristics, all inspiration and expiration starts are well estimated. Note that the procedure has been primarily developed to segment signals recorded with DCP, specifically for mice's nasal airflow as their breathing is mainly nasal. However, the algorithm field of application can be extended to other plethysmographs.

4.2 The experimental application

4.2.1 The biological context

Acetylcholine, a neurotransmitter controlling ventilation and much more. Acetylcholine (ACh) neurotransmitter is a key mediator for the conscious or unconscious regulation of breathing. This chemical compound allows signal transmission across cholinergic neurons or other cell types. It is present in several organs and tissues, and it mediates the regulation of numerous physiological mechanisms, including ventilation. For instance, it is present in the central nervous system (CNS), specifically in the brain, where it plays an important role in memorizing and learning [DB07]. It is also present in the peripheral nervous system (PNS), notably in neuromuscular junctions (NMJ), where it permits the transmission of signals from a neuron to a muscle fiber, causing muscle contraction [Sla15]. More specifically, in the somatic nervous systems ACh is involved in conscious contractions of skeletal muscles and in the autonomic nervous system, it mediates several vegetative functions like digestion, pupil dilatation, heart rate regulation, or breathing [MLA23]. Figure 4.7 describes the nervous system taxonomy.

A fast and powerful transmission process. The signal transmission between two neurons or a neuron and another cell type occurs at a synapse, a functional contact zone between two cells. To pass a signal between cholinergic cells, the transmitter neuron releases ACh in the synaptic cleft, which rapidly binds and unbinds with receptors located at the surface (membrane) of the target cell (neuron or muscular fiber). To terminate the

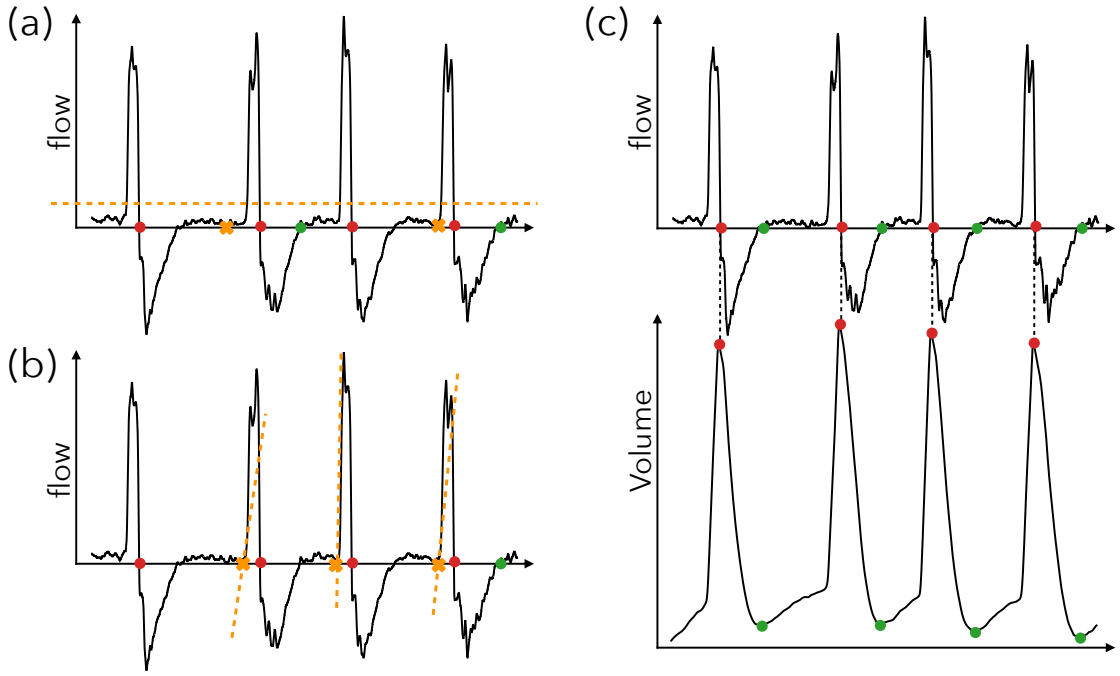


Figure 4.6 Heuristics for segmenting respiratory cycles tested on a real nasal airflow signal with severe breathing alterations. Correctly estimated inspirations and expirations starts are represented by dots in green and red, respectively. Poorly estimated starts are represented by orange crosses. **(a)** The threshold-based heuristic misses two inspiration starts, as the signal crosses the zero line during inspiration. **(b)** The tangent-based heuristic misses all inspiration starts as the lines passing by points at 5% and 10% of the maximum inspiration flow disregard the apnea occurring during inspiration. **(c)** The volume-based heuristic correctly estimates all starts thanks to the robustness to noise of the volume signal and the peak search algorithm.

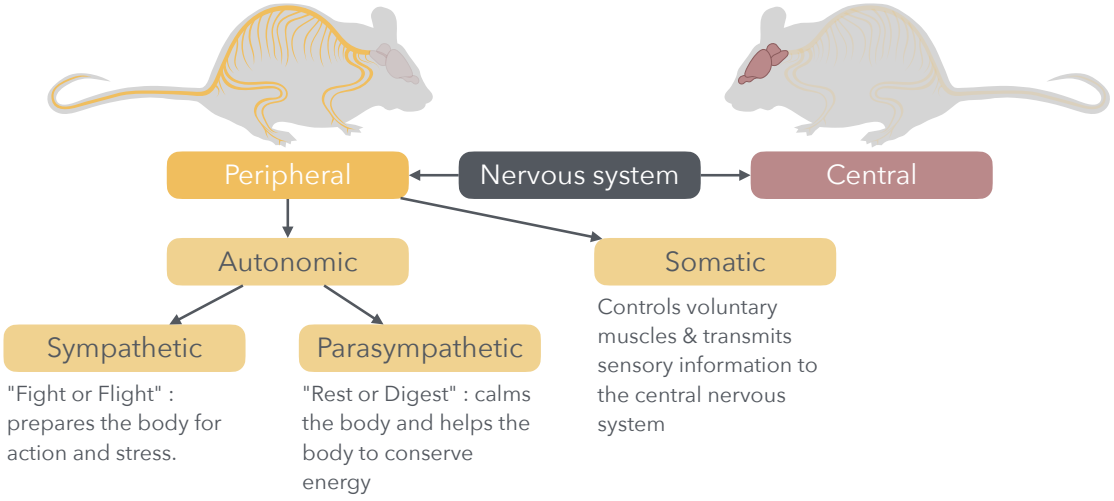


Figure 4.7 Mouse nervous system taxonomy

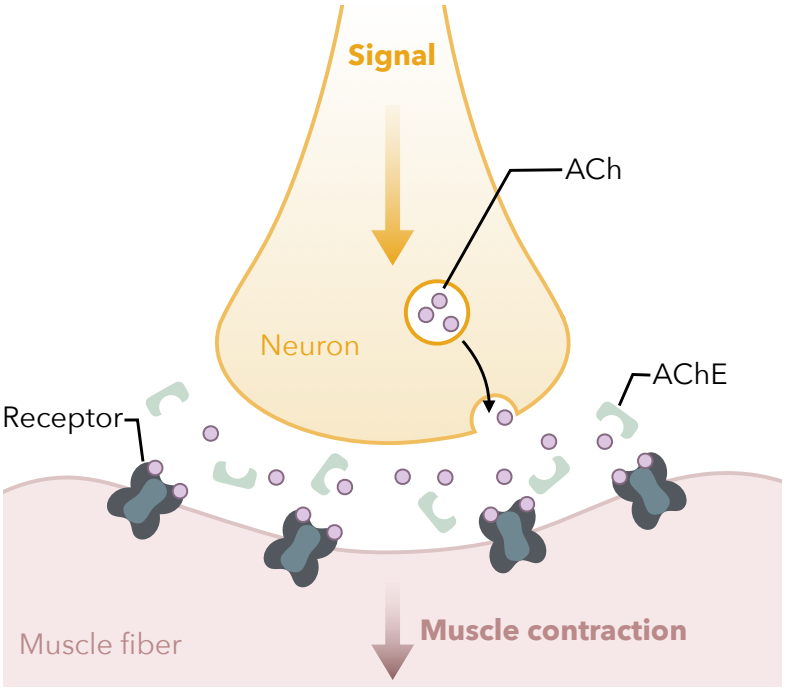


Figure 4.8 Cholinergic signal transmission at a neuromuscular junction. To transform a cholinergic signal in a muscle contraction, the transmitter neuron releases ACh in the synaptic cleft, which rapidly binds and unbinds with receptors located at the surface of the muscle fiber. To terminate the signal transmission, AChE destroys ACh by hydrolysis.

signal transmission, the Acetylcholinesterase (AChE) enzyme destroys ACh by hydrolysis. AChE is also located in the synaptic cleft, and its time to hydrolyze ACh is extremely short, no more than a few milliseconds. In particular, AChE concentration is high in NMJs [Ber+11], since ACh is abundantly released in NMJs when passing a signal to ensure rapid muscle responses. Figure 4.8 illustrates the cholinergic transmission process at a neuromuscular junction.

Receptors are complex chemical structures. Membraneous receptors receive and transduce a signal carried by chemical messengers into the target cell. While receptors are dedicated to chemical messengers, it is difficult for a cell to implant receptors at specific locations on its membrane due to the structural complexity of receptors. Consequently, cells are extremely sensitive to the relative position between their receptors and chemical messenger sources. One strategy organisms employ to specialize receptors and avoid unwanted activations consists of combining receptors with enzymes able to destroy chemical messengers before they reach the receptors.

AChE, an enzyme with multiple roles. The role of AChE is not limited to ending a signal transmission, but it also intervenes in the receptor's specialization in different ways. Two roles are commonly recognized for the AChE [MLA23]:

- **Ending signal transmissions:** At a synapse level, AChE ends the signal transmission by hydrolyzing ACh after the receptor activation.

- **Avoiding ACh spillover:** At a NMJ and in synapses of the CNS, AChE prevents ACh from spreading out of the synaptic cleft to block the activation of receptors located on the muscle fiber or other membrane neurons and in the neighborhood of the synapse. These receptors can trigger unwanted mechanisms, and AChE prevents this by hydrolyzing all ACh before it can leave the synapse.

Closely related to AChE, another enzyme capable of hydrolyzing ACh exists, the Butyrylcholinesterase (BChE) [Loc15]. This enzyme is present in the liver and the plasma (blood) and is often described as a backup for AChE. Recently, it has been suggested that AChE and BChE play preventive roles [PPK21]:

- **A local shield against ACh activation:** In some localized areas, cells should not be activated by ACh. However, they present ACh receptors on their membrane, and in order to their activation, these receptors can be abundantly surrounded by AChE or BChE enzymes.

- **Preventing ACh from spreading throughout the body:** In the body, many cells produce ACh, while many others are sensitive to ACh. To prevent certain mechanisms from being triggered involuntarily by the diffusion of ACh in the plasma, AChE or BChE enzymes are also diffusely present in the plasma to drain it of ACh.

The centrality pitfall. Hypothetically, if AChE and BChE are inhibited, ACh is no longer hydrolyzed, and it remains present in large quantities around the emitter cells but also diffuses into the body. Consequently, signal transmissions cannot be terminated, and many cells find themselves involuntarily activated, erratically triggering unwanted physiological mechanisms. Given the wide range of physiological processes involving ACh, the consequences of AChE and BChE inhibition are numerous and of varying severity.

Organophosphorus compounds: lethal ChE inhibitors. Examples of ChE (AChE & BChE) inhibitors are the organophosphorus compounds (OPCs). They are irreversible inhibitors, meaning they permanently prevent ChE from hydrolyzing ACh. Because of their notable toxicity, these inhibitors are widely used as pesticides, Parathion/Malathion (banned worldwide but still used), or as weapons of mass destruction, nerve agents (Novichok, Sarin, and so on). An exposure, even to low doses over a short period, may trigger characteristic reactions like: miosis, dim and blurred vision, headaches, bronchoconstriction, hypersecretion in airways, nausea with vomiting and diarrhea, muscle contractions leading to paralysis, deterioration of mental state, loss of consciousness, convulsions, and apneas, which can lead to death [Ner18]. A 2020 study states that approximately 740,000 unintentional expositions to OPC pesticides, including 7,446 deaths, were reported from 141 countries in the course of one year. However, due to inadequate reporting, they have estimated that the number of expositions worldwide should be around 385 millions, including 11,000 deaths [Boe+20].

Most deaths caused by OPCs poisoning are due to respiratory failure (respiratory muscle paralysis, bronchospasm, bronchorrhea, central apnea) and pulmonary dysfunction [Can06; TR87]. Understanding the influence of OPCs on breathing becomes essential to protect vital functions more adequately after exposure to OPCs. However, such analysis must be carried out with a methodology capable of dissecting the complex role played by AChE/BChE and the consequences of their inhibition.

Inferring OPCs influences on mice with genetics. From a treatment perspective, the research currently focuses on improving the accessibility of a medicine, the oxime (Organophosphate-inhibited AChE reactivator), to the CNS for reactivating AChE enzymes that have been inhibited after an exposition to OPCs. Meanwhile, it has been shown that mice without AChE survive. This is because BChE in the brain and peripheral tissues hydrolyzes ACh. However, BChE inhibitors that do not penetrate the brain have a respiratory arrest within minutes [Cha+03], stressing the need to understand better the consequences of OPC exposition for designing suitable treatments.

To establish this argument, researchers have genetically selected mice without AChE enzymes in the whole body. These mice present severe phenotypic alterations but survive thanks to ACh hydrolysis by BChE, playing a backup role for AChE. Researchers have shown that the BChE does not play a role in the regulation of breathing in the CNS. Hypothetically, an OPC exposition should have a minor effect on these mice due to the AChE scarcity. However, the mice still suffer from severe breathing alterations when exposed, indicating that protecting CNS functions from OPC exposition is insufficient for survival [Cha+03; Duy+01].

Plethysmography and genetic. Leveraging genetics to reduce problems' complexity is a well-established approach in biology, made possible by significant advances in genomics. In the specific context of studying the effects of exposures to OPCs, the success of this approach also depends on the ability to accurately quantify respiratory changes and assess the distress state of mice from plethysmography signals. This need forms the basis for the work presented in the following two chapters.

Experiment positioning. Recent studies have demonstrated that the inhibition of AChE/BChE in the CNS or PNS alone does not fully explain the breathing alterations observed after exposures to OPCs [Cha+03; Min+02]. The fact that ACh is also used by cells to communicate outside the nervous system may explain this observation. In line with this, it has recently been shown that small amounts of ACh are synthesized by non-neuronal cells, such as solitary chemosensory cells (SCC) in the respiratory tract. Specifically, when SCC detects a specific irritant molecule, the cell releases ACh. On the one hand, ACh activates neighboring cells, potentially triggering mucociliary clearance, a protective reflex that traps irritants in mucus, which is then evacuated by micro-vibrations and reflexes when the mucus is too abundant. On the other hand, ACh may directly activate a sensory neuron that triggers a respiratory reflex [Kra+11].

These observations laid the foundation for the experiment conducted by Aurélie Nervo [Ner+19] and discussed in the next section. The hypothesis to be tested in the experiment was formulated as follows:

"Severe alterations in breathing after ChE inhibition are due to an excess of ACh which, in addition to modifying cholinergic synaptic transmission in the CNS and PNS, escapes from cholinergic synapses, mainly those at NMJs, to reach non-synaptic cholinergic receptors in sensory nerve afferent pathways, possibly regulated in normal circumstances by non-neuronal ACh.", from [Ner18]

In essence, the experiment aimed to determine whether the diffusion of ACh throughout the body, following ChE inhibition, leads to abnormal activation of cells typically stimulated by non-neuronal ACh, potentially explaining the observed breathing alterations.

4.2.2 The experiment

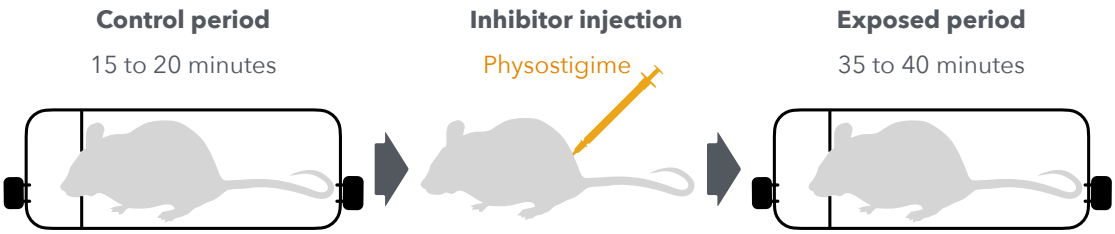


Figure 4.9 Experimental protocol: a mouse of a given genotype is placed in DCP for 15 to 20 minutes before injecting physostigmine, a ChE inhibitor. Afterward, the mouse's breathing is recorded for 35 to 40 minutes.

In the following chapters, we applied our methodologies to a subset of the data from the experiment [Ner+19], and we solely discussed experimental information to this subset. Interested readers can refer to [Ner+19] for a full description and a discussion of the experiment. Note that all experiments were carried out in compliance with the European Committees Council Directive (86/609/EEC) and were approved by Paris Descartes University ethics committee for animal experimentation (CEEA34.EK/AGC/LB.111.12).

The experimental protocol. According to the genetic approach, mice of different genotypes were exposed to a ChE inhibitor, and their breathing was monitored with a double chamber plethysmograph (DCP). Each genotype differently expresses the presence or absence of AChE/BChE at different sites, and the recording procedure was as follows:

1. Phase 1: The mouse is placed in a DCP for 15 or 20 minutes to serve as an internal control.

2. Phase 2: The mouse is removed from the DCP and injected with a ChE inhibitor.

3. Phase 3: The mouse is placed back into the DCP, and its breathing is recorded for 35 or 40 minutes.

For all mice, the nasal and thoracic airflows were recorded at 2,000Hz. By default, the double chamber plethysmograph includes a bandpass filter, whose band limits are 0.250Hz and 35000Hz, which has not been modified. Figure 4.9 illustrates the experimental protocol.

The ChE inhibitor. Mice were injected with physostigmine, a carbamate compound that is readily distributed throughout the body, including the CNS. Physostigmine is a reversible inhibitor with a high affinity for AChE/BChE, that ChE enzymes are inhibited throughout the monitoring process.

Mice. We considered 4 different genotypes and kept signals of 8 mice per genotype. The different genotypes were:

- WT (Wild Type): Mice have all forms of AChE and BChE. As well, their cholinergic system is normal and functional. It is the control group.
- PRiMA KO: Mice have an AChE deficiency in cholinergic neurons of the brain and peripheral nervous systems (autonomic and enteric) [Dob+09].
- ColQ KO: Mice have AChE deficiency in neuromuscular junctions (NMJs) [Fen+99].
- AChE1iRR: Mice have no AChE in skeletal muscle [Cam+08].

Method-related preprocessing. In the next chapters, signals may be preprocessed differently depending on the proposed embedding method. Details of the preprocessing steps will be provided in the respective chapters.

Chapter 5

Symbolic embedding

Key points:

1. This chapter presents a foundational approach for shape-based analysis of plethysmography signals from mice exposed to a drug affecting respiration. The goal is to create a symbolic embedding of respiratory cycles, where each symbol captures physiological information not easily discernible through traditional ventilation descriptors.

Contributions:

1. This chapter introduces a baseline method that compares respiratory cycles using a DTW-based clustering algorithm, resulting in a shape-based symbolic representation where each symbol represents a cluster. Tracking these symbols over time results in a symbolic representation of plethysmography signals.

2. This approach facilitates the discovery of various ventilation modalities that are not captured by conventional descriptors. Notably, the symbolic representation helps identify genotype-specific adaptations to enzyme deficiency and reveals diverse responses to drug exposure.

Associated papers:

- Thibaut Germain et al. "Unsupervised classification of plethysmography signals with advanced visual representations". In: *Frontiers in Physiology* 14 (2023), p. 781

- Thibaut Germain et al. "Unsupervised study of plethysmography signals through DTW clustering". In: *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE. 2022, pp. 3396–3400

Contents

5.1 A symbolic framework for mice ventilation analysis.

As discussed in the previous chapter, mouse ventilation modalities can be inferred through visual interpretation of the shape of respiratory cycles. By applying Algorithm 10 to segment plethysmography signals into a dataset of respiratory cycles, a shape-based clustering approach emerges as a natural solution for grouping these cycles according to ventilation modalities, leading to a symbolic representation with physiological significance. The embedding method we propose combines K-means clustering with Dynamic Time Warping (DTW) distance to preserve visual interpretability. This method serves as a baseline for the analysis of mice ventilation.

Shape modeling. The proposed approach applies a Kmeans clustering on a dataset of respiratory cycles extracted from mice cohorts. To prevent the algorithm from learning meaningless decision boundaries between clusters, the distance functions should be invariant to irrelevant sources of inter-individual variabilities while preserving the shape characteristics relevant to mice ventilation analysis. Essentially, the chosen distance is invariant to amplitude scaling and offset shifts, thus mitigating deformations induced by mouse size and recording distortions. Also, deformations caused by time warping are mitigated in a limited manner to minimize breathing frequency variabilities while remaining sensitive to apneas, an important modality when studying ventilation. Although a formal mathematical definition of an invariant distance function based on group action is not possible due to the lack of a group structure for the deformations set, the following section details the implementation of the distance function and outlines the complete methodology.

5.2 Method

5.2.1 Overview of the method

The method is composed of three main steps:

(a) Extracting inspiration and expiration sequences.

(b) Computing reference sequences with a DTW-based clustering algorithm.

(c) Performing the symbolic embedding of recordings with the extracted reference sequences.

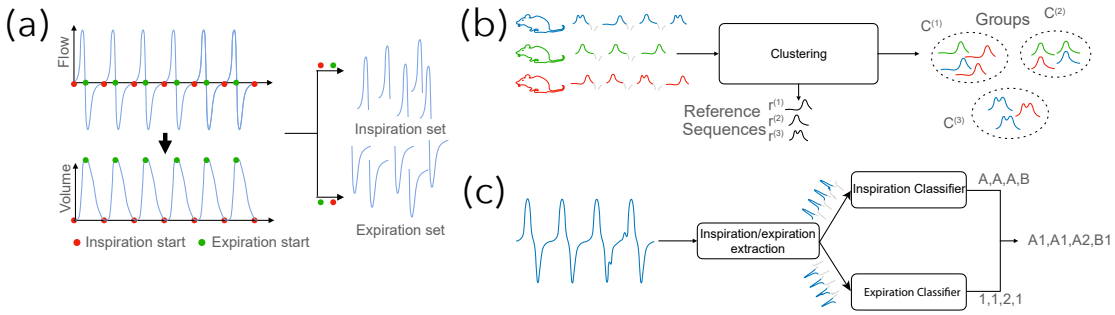


Figure 5.1 (a): Step 1, Extracting inspirations and expirations. **(b)**: Step 2, Computation of the reference sequences, $\mathsf{C}^{(i)}$ denotes the clusters associated to the reference sequence $\mathbf{r}^{(i)}$. **(c)**: Step 3, Symbolic embedding of a recordings.

Step 1: Detection of the respiratory cycles and extraction of the inspiration/expiration sequences. Following the procedure detailed in Section 4.1.3, respiratory cycles are extracted from an airflow signal and decomposed to form a set of inspirations $\{\mathbf{s}_{in}^{(1)}, \ldots, \mathbf{s}_{in}^{(N_s)}\}$ and a set of expiration sequences $\{\mathbf{s}_{out}^{(1)}, \ldots, \mathbf{s}_{out}^{(N_s)}\}$, where N_s is the total number of segmented cycles. Figure 5.1a illustrates the process for creating the inspiration and expiration sets.

Step 2: Computation of the reference sequences. In the second step, a small number of reference sequences from the inspiration and expiration sets are computed. The reference sequences represent groups of inspiration/expiration sequences with similar shapes and are potentially linked to one or several inspiration/expiration modalities. To that aim, a clustering algorithm Kmeans is combined with the Dynamic Time Warping (DTW) distance measure, which computes the similarities between sequences of potentially different lengths. The output of this step is a set of inspiration reference sequences $\{\mathbf{r}_{in}^{(1)}, \mathbf{r}_{in}^{(2)}, \ldots\}$ and a set of expiration reference sequences $\{\mathbf{r}_{out}^{(1)}, \mathbf{r}_{out}^{(2)}, \ldots\}$. Figure 5.1b illustrates the computation process of reference sequences in the case of inspiration.

Step 3: Symbolic embedding of recordings. The objective is to automatically characterize a recording $\mathbf{s}'$ with the reference sequences learned during Step 2. To that end, the signal is first segmented through the procedure described in Step 1. Then, each of the N' inspiration/expiration sequences present in $\mathbf{s}'$ is assigned a symbol representing the closest reference sequence considering the DTW. This procedure results in a symbolic

representation of $\mathbf{s}'$, where each respiratory cycle is replaced by a symbol composed of a letter (which specifies the type of inspiration) and a number (which specifies the type of expiration). Figure 5.1c illustrates the symbolic embedding of a recording.

5.2.2 Computation of the reference sequences

Provided a set of inspiration/expiration sequences, we now aim to compute K reference sequences representing typical inspiration or expiration patterns potentially linked to ventilation modalities. In the following sections, $\mathsf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ represents a set of sequences (either inspiration or expiration) of potentially different lengths.

The clustering algorithm. The K reference sequences are learned from the set X with the Kmeans algorithm [KR09]. This algorithm partitions X in K groups (or clusters) $\{\mathsf{C}^{(1)}, \ldots, \mathsf{C}^{(K)}\}$ of sequences with similar patterns. Roughly, Kmeans is a two-step iterative refinement technique that assigns each sequence to the closest current centroid and then updates each centroid with regard to the new assignments. A centroid is a reference sequence $\mathbf{r}^{(i)}$ which corresponds to the average sequence of the cluster $\mathsf{C}^{(i)}$. In our case, the Kmeans algorithm relies on a DTW distance to assign sequences to clusters, the DTW, and a DTW-based averaging algorithm to compute the reference sequences. Algorithm 11 describes the clustering, and its key components are described in the next paragraphs. Note that during the preprocessing step, all sequences are first centered to zero mean and scaled to unit variance using the z-normalization. It enforces invariance to amplitude scaling and offset shift, which has been experimentally shown to be necessary for learning relevant clusters. Also, the clustering algorithm stops after 10 iterations for all experiments.

Algorithm 11 DTW-Kmeans

Require: X a set of sequences, K the number clusters, *maxiter* maximum iterations

$\mathsf{X} \leftarrow$ z-normalization(X) ▷ *removing amplitue scaling and offset shift*
$\mathcal{C} \leftarrow$ Kmeans++($\mathcal{X}, K$) ▷ *clusters' centroid intilization [AV07]*
$niter \leftarrow 0$
while $niter < maxiter$ **do**
  $A \leftarrow$ DTW-assignement($\mathcal{C}, \mathsf{X}$)
  $\mathsf{C} \leftarrow$ update-centroid($\mathcal{C}, A, \mathsf{X}$) ▷ *BS-DBA procedure, see Algorithm 12*
  $iter = iter + 1$
$A \leftarrow$ DTW-assignement($\mathcal{C}, \mathsf{X}$)
return $\mathcal{C}, A$

Dynamic Time Warping distance. At each iteration, the Kmeans algorithm assigns each sequence to the nearest centroid according to the DTW distance [BC94]. In its original form, the DTW has been created to compare discrete time series of potentially different lengths independently to their time parametrization. Formally, the DTW distance

between $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^n$ is defined by:

$$DTW(\mathbf{x}, \mathbf{y}) = \min_{A \in \mathsf{A}_{m,n}} \langle A, \Delta \rangle_F, \quad \text{where:} \quad \Delta_{ij} = \|x_i - y_j\|^2 \tag{5.1}$$

where $\mathsf{A}_{M,N} \subset \{0,1\}^{M \times N}$ is the set of path matrices that connect the top-left corner $(1,1)$ to the bottom-right corner (m,n) solely with moves: $\rightarrow, \searrow, \downarrow$ [CB17]. The distance can be computed with dynamic programming with a time and space complexity of $\mathcal{O}(mn)$.

DTW is commonly used in times-series data-mining [EA12a; Fu11] where it has notably shown great success in classifying and clustering short time series [Wan+13].

In its original form, the DTW measure is sensitive to noise and outliers, potentially leading to pathological and unrealistic time parametrization. In addition, some ventilation modalities, like those including apneas, depend on the time warping, stressing the need to control the elasticity of the DTW. To overcome these issues, we constrain the DTW with the Sakoe-Chiba constraint, which imposes that the dilations are smaller than a given duration [SC78]. Formally, given a threshold $\alpha > 0$, this constraints the set of warping matrices to the set: $\mathsf{A}^{\alpha}_{m,n} = \{A \mid A \in \mathsf{A}_{m,n}, \text{ with } A_{ij} = 0 \text{ when } |i-j| > \alpha\}$.

Time-series averaging. Updating clusters' centroid is an important subroutine of Kmeans algorithm. Indeed, the quality of each cluster is highly dependent on the quality of its centroid [ASW15]. At each iteration, all sequences in the data set X are assigned to their closest centroids $\{\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(K)}\}$. Then, each centroid is updated by computing the average sequence based on the new assignment.

For any set of sequences $\mathsf{X}' = (\mathbf{x}'^{(1)}, \ldots, \mathbf{x}'^{(M)}) \subset \mathsf{X}$, the average sequence, with respect to the constrained DTW_α, is a minimizer of the cost function:

$$F_{\mathsf{X}'} : \mathbf{y} \in \mathbb{R}^l \mapsto \sum_{\mathbf{x}' \in \mathsf{X}'} \mathrm{DTW}^2_\alpha(\mathbf{y}, \mathbf{x}') \in \mathbb{R} \tag{5.2}$$

where $L > 0$ is the average duration of the sequences in X' .

Accurately and efficiently solving Equation (5.2) is not trivial [NR07; Jai19]. Traditional averaging methods cannot deal with the non-linear mapping between sequences of potentially different lengths, and several algorithms have been proposed to solve this problem [PKG11; Mor+18]. A recent work [SJ18] uses the subdifferentiability property of the optimization function to develop a stochastic subgradient descent algorithm (S-DBA). Specifically, the subgradient of $F_{\mathsf{X}'}$ at a point $\mathbf{y} \in \mathbb{R}^l$ is:

$$\nabla F_{\mathsf{X}'}(\mathbf{y}) = \frac{2}{M} \sum_{u=1}^{M} \left(V^{(u)} \mathbf{y} - W^{(u)} \mathbf{x}'^{(u)} \right)$$

where $W^{(u)} \in \mathsf{A}^{\alpha}_{ln}$ is the optimal warping matrix between $\mathbf{y} \in \mathbb{R}^l$ and $\mathbf{x}'^{(u)} \in \mathbb{R}^n$ and $V^{(u)}$ is a diagonal matrix in $\mathbb{N}^{l \times l}$ such that:

$$V^{(u)}_{i,i} = \sum_{j=1}^{n} W^{(u)}_{ij}$$

We implemented a batch version of S-DBA called BS-DBA for a trade-off between accuracy and speed. BS-DBA is presented in Algorithm 12. If the initialization sequence $\mathbf{y}_{ini}$ is not given, it is set to a vector of size l sampled from an uniform distribution on $[0, 1]$. The learning rate scheduler η is taken from [SJ18]:

$$\eta^{(t)} = \begin{cases} \eta^{(t-1)} - (\eta_0 - \eta_1)/\beta & \text{if } 1 \leq t \leq \beta \\ \eta_1 & \text{otherwise} \end{cases}$$

where $\eta_0 = \eta^{(0)} = 0.05$, $\eta_1 = 0.01$, n_b is the batch size and $\beta = \lfloor N/n_b \rfloor + 1$ is the number of iteration for one epoch. The learning rate only decreases during the first epoch then it remains fix to η_1. The algorithm stopping criteria is the total number of iterations.

Algorithm 12 BS-DBA

Require: $\mathsf{X} = (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)})$ a set of time-series, $\mathbf{y}_{ini}$ (optional) the starting sequence, n_{epochs} the number of epochs, L the length of the averaging time-series, n_b the size of a batch, n_{it} the number of iterations, η the learning rate scheduler.

if $\mathbf{y}_{ini}$ is given **then**
  $\mathbf{y}^{(0)} \leftarrow \mathbf{y}_{ini}$
else
  Initialize $\mathbf{y}^{(0)} \in \mathbb{R}^L$
Initialize best solution $y^* \leftarrow y^{(0)}$
for *epoch* $= 1, \ldots, n_{epochs}$ **do**
  Batches $\leftarrow$ *randomly partition* X *in batches of size* n_b
  for *batch* $\in$ *batches* **do**
    for $\mathbf{x}^{(k)} \in$ *batch* **do**
      $P^{(k)} \leftarrow$ *Optimal warping path between* $\mathbf{y}^{(t-1)}$ *and* $\mathbf{x}^{(k)}$
      $W^{(k)} \leftarrow$ *Warping Matrix of* $P^{(k)}$
      $V^{(k)} \leftarrow$ *Valence Matrix of* $P^{(k)}$
    Update temporal solution:

$$\mathbf{y}^{(t)} \leftarrow \mathbf{y}^{(t-1)} - \eta^{(t)} \frac{2}{n_b} \sum_{k=1}^{n_b} \left(V^{(k)}\mathbf{y} - W^{(k)}\mathbf{x}^{(k)} \right)$$

    Update best solution such that: $\mathbf{y}^* = \operatorname{argmin}\left(F_{\mathsf{X}}(\mathbf{y}^*), F_{\mathsf{X}}(\mathbf{y}^{(t)})\right)$
    if $t \geq n_{it}$ **then**
      break
return $\mathbf{y}^*$

5.2.3 Characterization and symbolization of recordings

From a recording $\mathbf{s}'$, we first perform the segmentation process described in Section 4.1.3 to extract the inspiration/expiration sequences. Then, we use a 1-NN (nearest neighbor) algorithm to assign each sequence to the reference sequence, which is the closest to it, in the sense of the DTW_α measure.

To avoid incoherent symbols, some inspiration/expiration sequences are treated as outliers if their distance to their reference sequence is higher than a threshold. The threshold is different for each reference sequence. It corresponds to the η-quantile of the distance distribution observed within the reference sequence cluster during the learning step. By default, we choose the threshold value $\eta = 0.95$.

This procedure yields a symbolic representation of $\mathbf{s}'$, where each respiratory cycle is replaced by a symbol composed of a letter (which specifies the type of inspiration) and a number (which specifies the type of expiration).

Connection with ventilation pattern descriptors. In the present work and for the purpose of validation, we have used four descriptors:

- **Inspiratory/Expiratory Time (Ti/Te, s):** Duration of inspiration/expiration.
- **Nasal Inspiratory/Expiratory Volume (NIV/NEV, ml):** Volume of air in/out during inspiration/expiration.

5.3 Experimental settings

A python implementation of the method is available on Github [1].

Dataset preprocessing. Our data set includes all 32 nasal airflows from the dataset present in Section 4.2.2. All signals have been down-sampled to 250Hz. It includes 8 recordings for each genotype: WT, PRiMA, AChE1iRR, ColQ. All mice were exposed to the same inhibitor: physostigmine.

On average, a mouse's respiratory cycle lasts about 0.3 seconds. The original data set contains approximately 350,000 cycles; therefore, updating reference sequences from the entire data set would have been too time-consuming. Thus, we extracted 1800 cycles for each recording that were evenly selected in time. This subsampling corresponded to approximately 36 cycles per minute, resulting in a set of 57,600 cycles that were divided into an inspiration training data set and an expiration training data set.

Experimental protocol. In order to test our approach, we have run and evaluated the results of the following experiment:

1. Extraction of training data set for inspiration/expiration.
2. Computation of inspiration/expiration referent sequences.
3. Symbolization of all signals in the data set.

[1] `https://github.com/thibaut-germain/DCP_Clustering`

Hyperparameters. The main parameters are presented below. Parameters for respiratory cycle detection have been set based on physician knowledge of the typical respiratory cycles. For the clustering algorithm, the number of clusters has been set arbitrarily and the Sakoe Chiba radius authorizes small dilatation.

- **Respiratory cycle detection (Step 1)**:
  - Prominence : 0.03 ml
  - Window length : 2 s
  - Minimum inspiration/expiration duration : 0.05 s
  - Maximum inspiration/expiration duration : 2 s
- **Clustering algorithm (Step 2, identical settings for inspiration and expiration):**
  - Number of clusters: 5
  - Number of iterations for Kmeans: 10
  - Sakoe Chiba radius: 0.01 s
  - Reference sequence length: 0.2 s
- **Symbolization (step 3):**
  - Quantile threshold: 0.95

5.4 Results

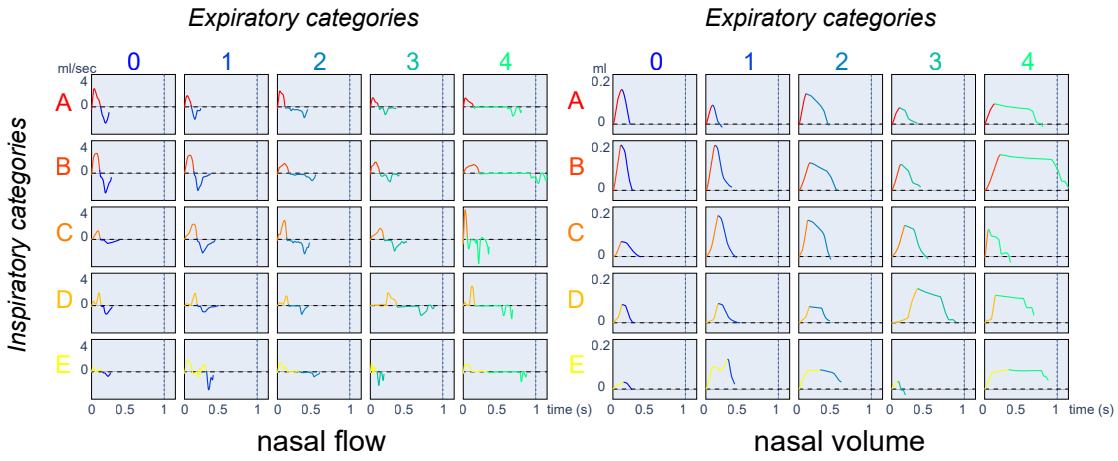


Figure 5.2 Respiratory cycle map displays with nasal airflow ($ml.s^{-1}$) on the left and nasal volume (ml) on the right. Positive flow corresponds to inspiration and negative flow corresponds to expiration.

Categorization of the respiratory cycles. We first aim to categorize breathing cycles, inspirations, and expirations. The limits of inspiration and expiration are unambiguously defined from the volume obtained by integrating the flow [Vij+93]. We define a referent cycle as the association between a referent inspiration and a referent expiration. Considering K_1 referent inspirations and K_2 referent expirations, there exist K_1K_2 referent cycles. In order to compare them, we develop a map where each row corresponds to a referent inspiration and each column to a referent expiration. Each referent cycle is represented by an actual cycle selected as follows:

- Among the identically labeled cycles in the training database, we select the cycle whose cumulative DTW distance (DTW distance to the referent inspiration + DTW distance to the referent expiration) is the smallest.

- The respiratory cycle map can be displayed using either the nasal airflow or the nasal volume. In any case, the inspiration/expiration phases are matched, accordingly, to their attributed colors. For inspiration, the color scale goes from red to yellow; for expiration, it goes from blue to green.

- Inspiration/expiration referent sequences are ordered in increasing order according to the average duration observed in each group. Therefore, as the number/letter increases, the average inspiration/expiration duration is longer. Visually, lighter colors (yellow/green) correspond to longer duration.

In our experiment, we set the number of inspiration and expiration referent sequences to 5, as presented in Figure 5.2. Short duration cycles (`A0`, `A1`, `B0`, `B1`) are characterized by a nasal airflow of sinusoidal shape. All 25 of the resulting classes are used in the following sections to visualize and compare the respiratory cycles of mice of different strains before and after physostigmine injection.

Distribution of respiratory cycle categories. In order to study the importance of each reference cycle for a given symbolic representation, we introduce a new visualization of the histogram that takes the form of a heat map. The respiratory cycle map (RC map) corresponds to a heat map where rows are inspiration symbols and columns are expiration symbols as presented in Figure 5.3a. Thus, each cell corresponds to a referent cycle, and its value is set to the percentage of time assigned to that specific referent cycle. To ease the study of less frequent referent cycles, we use a thresholded version of the respiratory RC map where all reference sequences that represent more than 20 % of the total duration are assigned to the threshold value of 20 %. A RC map provides a quick understanding of the dominant ventilation modality of a mouse. In addition, RC maps can be aggregated over a population, allowing comparisons of a mouse's ventilation modality to the average one.

In Figure 5.3b, RC maps are grouped by genotype: WT, PRIMA, AChE1iRR, ColQ. For each genotype, the two left columns gathered RC maps before injection, and the two left columns gathered RC maps after injection. The bottom line corresponds to the average RC maps observed per genotype before and after drug injection.

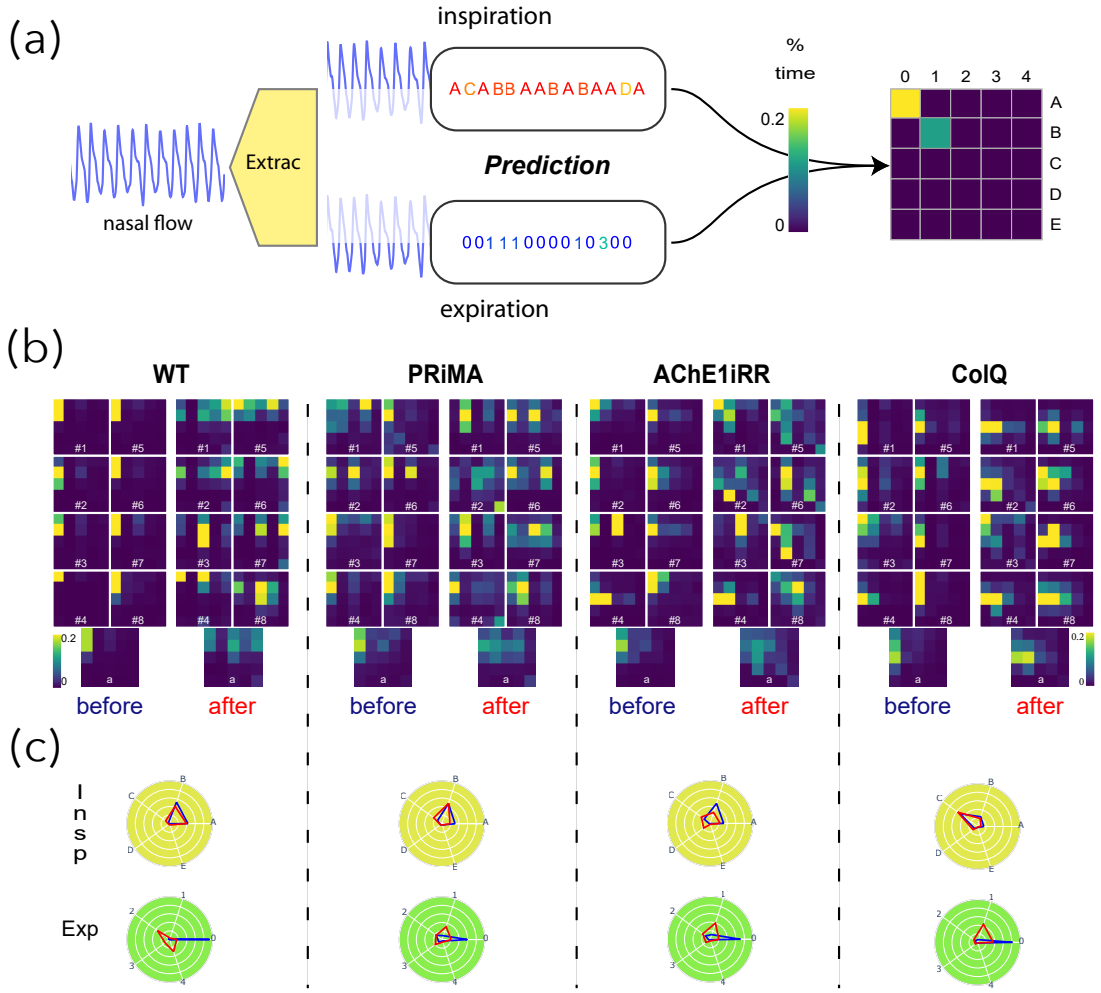


Figure 5.3 **(a)** Respiratory Cycle map (RC map) built-up process. **(b)** Respiratory RC maps: All RC maps are truncated at the threshold value of 20%. RC maps are grouped by genotype: WT, PRIMA, AChE1iRR, ColQ. For each genotype, the two left columns and the two right collumns gathered RC maps respectively before and after physostigmine injection. Numbers on RC maps correspond to the mouse id. The bottom line corresponds to the average RC maps observed per genotype before and after drug injection. **(c)** Average reference sequence polar plots: Polar plots are grouped by genotype. Inspirations are on the top, and expirations are on the bottom. The values on each angular axis correspond to the average percentage of time assigned to the associated reference sequence. The blue polygon corresponds to the values observed before injection, and the red polygon corresponds to the values observed after the injection.

In addition, we have created two conjoint polar plots, one for inspiration and one for expiration. Each angular axis corresponds to a referent sequence, and the value on each axis is equal to the percentage of time assigned to that specific referent sequence. These values are linked together to form a polygon. As for RC maps, the visualization can be done at the individual level or aggregated over a group of mice. This representation complements RC maps as it decorrelates inspiration from expiration, easing the study of both mechanisms independently as presented in Figure 5.3c.

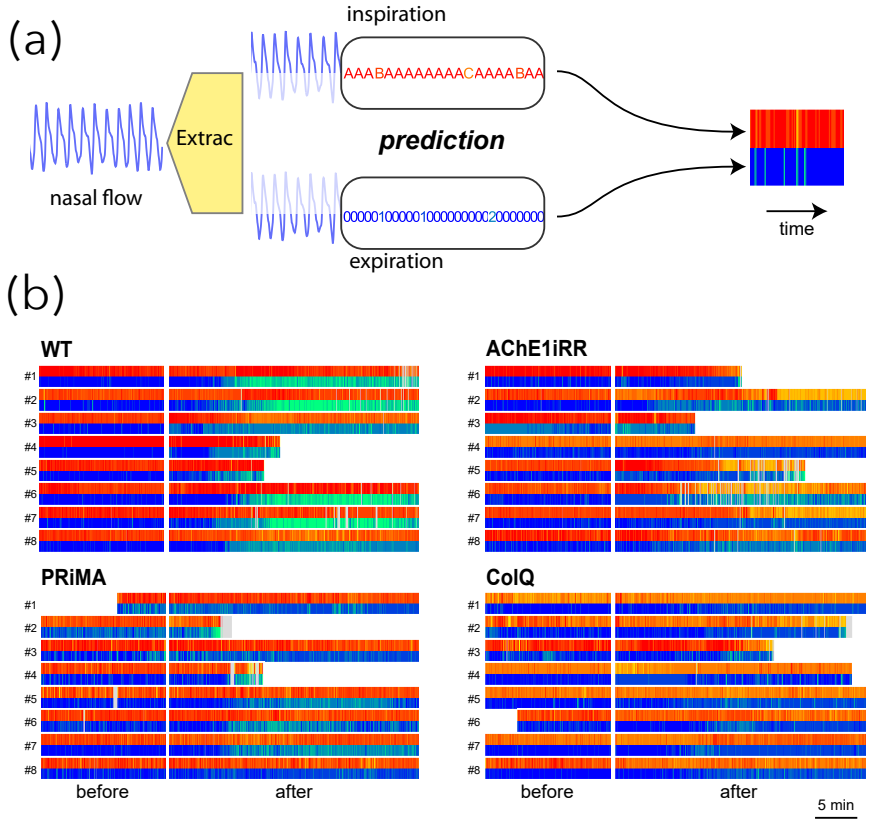


Figure 5.4 Respiratory bar codes: Respiratory bar codes are gathered by genotype: (top,left): WT, (bottom,left): PRIMA, (top,right): AChE1iRR, (bottom,right): ColQ. Numbers to the left of bar codes correspond to the mouse id. For each genotype, the left section corresponds to barcodes before drug injection and the right section to bar codes after injection. Grey areas in bar codes like mouse PRIMA-2 correspond to unpredictable cycles. Some experiments were shorter than others resulting in shorter bar codes.

Time line representation of respiratory cycle categories (bar codes). Previous representations give an overview of the ventilation modality of a mouse or a population. Nonetheless, they do not offer insights into the temporal evolution of a mouse's ventilation modality when facing a stressor. This evolution can be read from the symbolic representation with proper visualization.

To that aim, we construct a respiratory bar code for each mouse that includes the

time information, as presented in Figure 5.4a. The respiratory bar code is composed of two lines, the upper line represents the inspirations, and the lower line representing the expirations. The central white area corresponds to the period of inhibitor injection, and the light grey area corresponds to unpredictable cycles. Each line is composed of rectangles whose color refers to the associated reference sequence and whose length is proportional to the duration of the associated respiratory cycle.

Figure 5.4b presents respiratory bar codes of all mice in the data set. They are gathered by genotype, and mouse identification numbers are on the left of the bar codes. For each genotype, the left section corresponds to bar codes before injection and the right section to bar codes after injection.

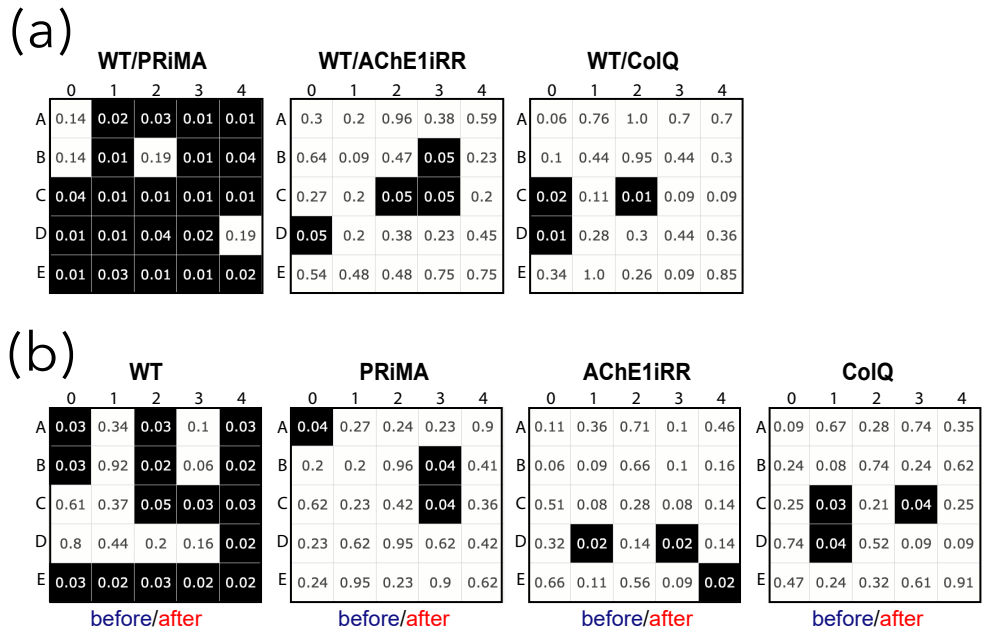


Figure 5.5 Multiple testing scheme with a false discovery rate (FDR) correction of 5%, performing a Mann-Whitney U test for each type of respiratory cycle. A cell is colored black if the unit null hypothesis is rejected after FDR correction and includes the corrected p-value of the associated unit test. **(a)** Statistical tests comparing the distribution of respiratory cycles of control (WT) and AChE-deficient (PRIMA, AChE1iRR, ColQ) mice before drug injection. **(b)** Statistical tests comparing the distribution of respiratory cycles before and after drug injection for each genotype.

Statistical analysis of respiratory cycle categories. RC maps provide visual comprehension of the heterogeneity in ventilation modalities and changes due to the presence of a stressor. In complement to the visual presentation, we provide a statistical analysis that compares the ventilation modalities between genotypes and the breathing responses to the presence of a stressor.

The first statistical test compares the respiratory cycle distribution of AChE-deficient mice (PRIMA, AChE1iRR, ColQ) with that of control mice (WT). The null hypothesis is that the cohort of AChE-deficient mice has the same respiratory cycle distribution

as the cohort of control mice. The alternative hypothesis is different respiratory cycle distributions.

The second statistical test compares the distribution of respiratory cycles for each genotype before and after drug injection. For the cohort of a given genotype, the null hypothesis is to have the same distribution of respiratory cycles before and after drug injection. The alternative hypothesis is different respiratory cycle distributions.

In both cases, we implemented a multiple testing scheme with a false discovery rate (FDR) correction of 5%, performing a Mann-Whitney U test for each type of respiratory cycle. Application of this test gives a map where each cell represents a type of respiratory cycle, with the row corresponding to the type of inspiration and the column to the type of expiration. A cell is colored black if the unit null hypothesis is rejected after FDR correction. In each cell, we also displayed the corrected p-value of the associated unit test.

All tests are rejected, Figure 5.5a, and the number of unit tests rejected at 5% is for WT vs. PRIMA: 21, WT vs. AChE1iRR: 4, WT vs. ColQ: 3. Similarly, all tests are rejected, Figure 5.5b, and the number of unit tests rejected at 5% is for WT: 15, PRiMA: 3, AChE1iRR: 3, ColQ: 3.

5.5 Discussion

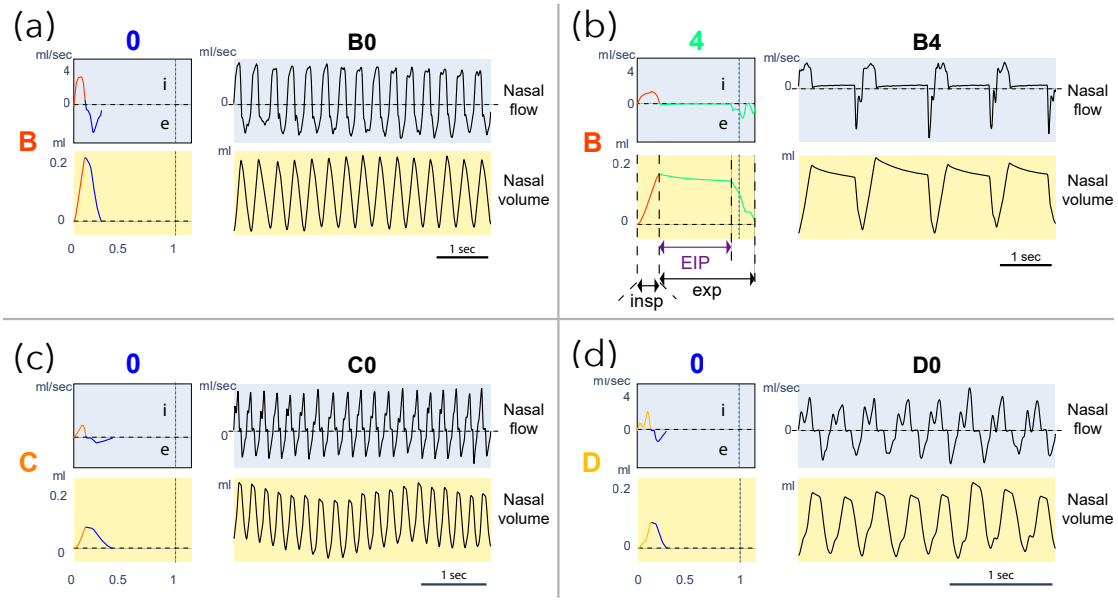


Figure 5.6 Examples of typical ventilation modalities. For each panel, the left column represents the referent cycle, and the right column is an extract from a recording of up to 5 seconds where the reference cycle is repeated continuously. Charts with a blue background are expressed in nasal airflow, and charts with a yellow background are expressed in nasal volume. **(a)** Referent cycle `B0`. **(b)** Referent cycle `B4`. Inspiration, expiration, and end-inspiratory pause (EIP) duration are illustrated. **(c)** Referent cycle `C0`. **(d)** Referent cycle `D0`.

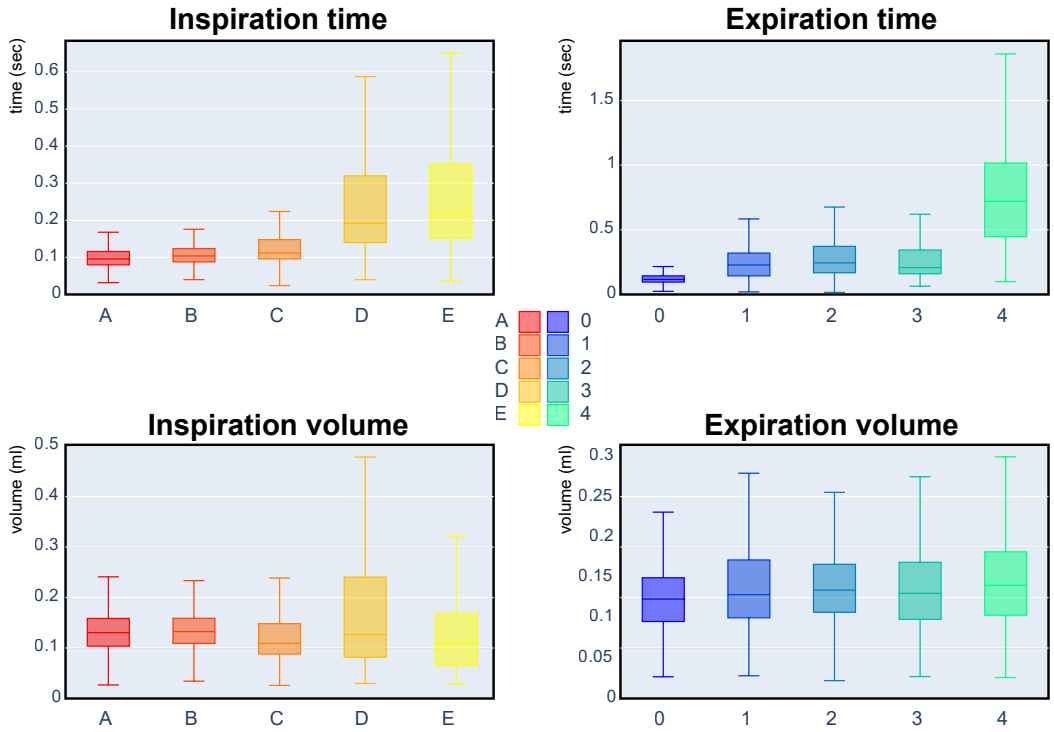


Figure 5.7 Box plots of the respiratory cycle descriptors: inspiration/expiration time and inspiration/expiration volume. Each box plot represents a referent sequence. A box represents the first quartile (Q1), median, and third quartile (Q3). The lower whisker corresponds to the minimum value observed, and the upper whisker is above the third quartile by 1.5 interquartile range (IQR: Q3-Q1).

This chapter presents a new method to compare and quantify cyclic signals that may be particularly appropriate for biological investigations, such as respiratory signals. Rather than comparing cycles based on the ventilation descriptors, cycles' shapes are compared to shape representations of most typical cycles. We will discuss the contributions and limitations of this new strategy by analyzing a part of recordings previously published [Ner+19].

5.5.1 Inspiration and expiration classes fit respiratory physiological control

The classes learned with the new approach represent various respiratory profiles that carry biological meaning. We illustrate some respiratory profiles through their classes in Figure 5.6. The last 15 minutes before physostigmine injection represents mice's baseline ventilation modalities. The control mice (WT) breathe with cycles of type `A0` and `B0`. Figure 5.6a shows 5 consecutive seconds of a raw signal with respiratory cycles of `B0`. After injection of physostigmine, the inspiration classes (`A` and `B`) are not changed for the control mice (WT), as shown with the polar plot (Figure 5.3c). However, the expiration class changes from type `0` to type `2,3,4`. Raw signals of 5 consecutive seconds of classes

B4 are presented in Figure 5.6b. The profile of these classes shows a long pause when the lungs are inflated. They correspond to post-inspiratory pauses. They were analyzed in [Ner+19], and the authors quantified the duration of these pauses. The new approach captures significant ventilation modalities making previous results apparent with the new representation: for control mice (WT), post-inspiratory pauses appear after inhibitor injection.

The approach also presents details about the inspiration dynamic of ColQ mice. Indeed, the cycles of ColQ mice before injection are grouped into types C0 and D0, which we present in Figure 5.6cd. Inspiratory classes C and D are characterized by a nasal airflow that enters in two phases. The two phases in class D are distinctive. Compared to D, the separation between phases is less visible in C. The ColQ mouse is a model of congenital myasthenic syndrome with AChE deficit at the neuromuscular junctions. This mouse shows an impairment of motor control, which could be reflected during the motor control required for a smooth inspiration.

Bar codes (Figure 5.3b) also validate inspiration and expiration classes. A bar code represents the symbolization of a raw signal as a timeline where inspirations and expirations are colored accordingly to their classes. Bar codes reveal the dynamics of ventilation modalities and their changes. For example, inspiration classes for control mice (WT) after physostigmine injection are almost unchanged. On the contrary, their expiration classes change significantly after a latent period. This dynamic is consistent with results in [Ner+19] where the mean frequency per minute of respiratory cycle decreases after the injection of physostigmine for control mice (WT). The frequency decrease corresponds to an increase in the duration of the post-inspiration pauses per min. Through the bar codes, it is possible to visualize the appearance of expiration classes 3 and 4 after injection with remarkable precision.

The inspiration and expiration classes have been constructed without prior knowledge of mice's ventilation modalities. Nonetheless, the classes present differences that can be interpreted in terms of physiological modifications. For instance, some of the expiration classes represent post-inspiratory pauses. New inspiration classes have also been described, probably related to the motor controls dynamics during the active ventilation phase.

5.5.2 Classes reveal heterogeneity: an observation masked by classical ventilation descriptors

Analyses on a small cohort can be biased if individual responses are heterogeneous. Unfortunately, it is often difficult to recognize this heterogeneity through some descriptors. The new symbolization, based on typical inspiration/expiration, the visualization and the quantification tools we proposed, offer perspectives on this critical issue in biology. For example, it is apparent on individual RC maps and bar codes that control mice (WT) present homogeneous respiration; the respiration cycle types are A0 and B0. After injection of AChE inhibitor, the RC maps and bar codes of control mice WT-1,2,6,7 show that they follow the same evolutionary dynamics. Nevertheless, mice WT-3,8 present different dynamics, and mice WT-4,5 died during the experiment. Thus, we can conclude that mice adapt differently to cholinesterase inhibition by physostigmine. In addition, the tests highlight changes that are significantly different.

Boudinot et al. [Bou+09] and Nervo et al. [Ner+19] proposed that mice with partial AChE deficiency were remarkably adapted to AChE deficit in the brain, autonomic nervous systems, and muscles. Indeed, the most frequent respiratory cycles before injection are composed with the inspiration of type `A,B,C` and the expiration of kind `0,1,2`. Looking at Figure 5.7, these reference sequences share similar duration and volume. Therefore, it is impossible to differentiate the genotypes based on inspiration/expiration duration or volume.

The present study shows that the distributions of inspiration and expiration classes on AChE1iRR mice are similar. AChE1iRR mice do not have AChE in skeletal muscle. These mice show a high homogeneity of adaptation despite muscle weakness. In contrast, PRiMA mice, which have AChE deficiency in the brain and autonomic nervous systems, adapted well to AChE inhibition, but showed heterogeneous ventilation modality. The heterogeneity is apparent in inspiration and expiration classes, which suggests the possibility of different ventilation modalities to cope with AChE deficit in the nervous system. The cohort of ColQ mice also presents heterogeneity in ventilation modalities, specifically for inspiration. As discussed, the inspiration of ColQ mice is characterized by types `C` and `D`. In contrast, the inspiration of other genotypes is characterized by types `A` and `B`. While ColQ and AChE1iRR mice have similar AChE deficiency in neuromuscular junctions, AChE1iRR mice adapt better than ColQ mice which also have AChE deficit in other tissues. This result suggests that AChE deficit in skeletal muscle is insufficient to affect these mice's inspiration.

If the respiratory adaptations are different, it is not surprising that the consequences of the injection of physostigmine are so variable. Visualization of inspiration and expiration classes, either in RC maps or bar code, makes it possible to account for this diversity. After injection of physostigmine, the changes tend to affect inspiration in AChE1iRR and ColQ mice, whereas expiration is more affected in WT and PRiMA mice.

In summary, representing respiratory cycles by classes sharing similar shapes reveals a diversity of unsuspected ventilation modalities that were not identifiable with descriptors deduced from the airflow. This rich information is synthesized in graphical representations highlighting how mice respond differently to cholinesterase deficits or inhibition.

5.5.3 Inspiration and expiration classes evoke distinct biological processes.

Inspiration and expiration classes are defined without prior knowledge of underlying biological processes. Inspiration classes `A` and `B` represent a regular inspiration phase, while classes `C` and `D` represent an inspiration phase with a more or less significant pause. The pauses in category `C` are very short and always during inspiration; they probably correspond to a motor impairment during lung inflation (the main action of the diaphragm, a powerful muscle) or by a fine control of the glottis. The longer pauses of class `D` may occur during the air inflow and are probably similar, in nature, to class `C`. In contrast, the long pauses of Class `E` correspond to a sort of pause before the air enters the lungs. From a physiological point of view, these pauses could correspond to a delay in the glottis's active opening, which is required to allow air to enter into the trachea. Two situations can lead to the glottis remaining closed: the cessation of muscle contractions that control the glottis opening or the spasm (cramp) of the muscles that control the closing of the glottis.

Expiration class `0` represents a regular and probably passive phase of expiration. Classes `2,3` and `4` start with a post-inspiratory pause whose duration increases progressively from category 2 to category 4. These post-inspiratory pauses are well described in the literature and appear in different physiological conditions. They appear when it is necessary to increase the air pressure in the lungs (short pauses) or as reflexes (long pauses), such as those resulting from inhaling molecules that irritate the upper airways [Dut+14].

From these results we can conclude that inspiration and expiration classes learned from a subset of recordings selected from [Ner+19] carry interpretable physiological meaning. It is important to note that these classes are specific to the experiment. For instance, applying our method to a set of signals presenting bronchoconstrictions will likely lead to classes differentiating the severity/variety of constrictions in a finer way than using the EF50 metric [GB21].

5.6 Conclusion

This chapter introduced a baseline embedding method for analyzing mice's ventilation. It is a shape-based approach that creates a symbolic embedding of respiratory cycles with a DTW-based Kmeans algorithm, a tool from machine learning for time series. This simple and effective method surpasses current approaches by discovering ventilation modalities untractable with classical descriptors. Specifically, the resulting symbolic representation allows the characterization of genotype-related adaptation to ChE deficiency and reveals heterogeneous responses after drug exposure.

Chapter 6

Deformation-based embedding

Key points:

1. This chapter proposes an unsupervised shape-based method, named TS-LDDMM, for embedding time series of **variable length** and potentially **irregularly sampled** by fixed-size vectors. Each vector encodes the unique deformation mapping a referent time series to the observed one.

2. This method is built upon Large Deformation Diffeomorphic Metric Mapping (LDDMM), a framework from shape analysis that learns a unique deformation mapping two geometrical objects by integrating ordinary differential equations.

3. Deformations learned with LDDMM needed to be specified to ensure that the deformed time series remains a time series throughout the integration of the differential equations.

Contributions:

1. Section 6.3 describes a class of deformations preserving the graph structure of time series while ensuring a transitive action (Theorem 3). Lemma 1 describe suitable Reproducible Kernel Hilbert spaces for encoding such deformations.

2. Appendix B.5 demonstrates the identifiability of the model by estimating the true generating parameter of synthetic data, and we highlight the sensitivity of our method concerning its hyperparameters.

3. Appendices B.6 and B.7 illustrate the quantitative interest of such representation on classification tasks on real shape-based datasets with regular and irregular sampling.

4. Section 6.5.2 showcases the interpretability of TS-LDDMM embedding on the analysis of mice ventilation.

Associated paper:

- Thibaut Germain et al. "Shape analysis for time series". In: *Advances in neural information processing systems* (2024)

Contents

6.1 Introduction

This chapter describes an unsupervised shape-based method for embedding time series of variable length and potentially irregularly sampled by fixed-size vectors and tailored for any subsequent statistical analysis.

From distance to deformations. The approach presented in the previous chapter ensures the learning of meaningful clusters by relaxing the DTW invariance to time warping through restrictions the time warping set with the Sakoe-Chiba constraint. It means that when comparing a respiratory cycle to a reference sequence, the distance is no longer invariant to time warping; instead, it evaluates, to some degree, the warping deformations. Roughly speaking, the clustering algorithm learns the clusters by quantifying the respiratory cycles' deformations to the reference sequences.

To clarify, Figure 6.1 illustrates how a time series f is mapped onto another time series g through a distortion h and a time parameterization γ^{-1} by the relation $(f+h)\circ\gamma^{-1} = g$. It corresponds to a deformation caused by the group action outlined in Equation (1.9). Building on the previous observation, an alternative approach for embedding time series consists of embedding the deformation (h, γ^{-1}) mapping a referent time series f_0 to any other time series f. This approach raises identifiability concerns, addressed in this chapter, where we propose an unsupervised method representing respiratory cycles through the vectorized embedding of deformations. This method draws on Large Deformation Diffeomorphic Metric Mapping (LDDMM), a framework from shape analysis.

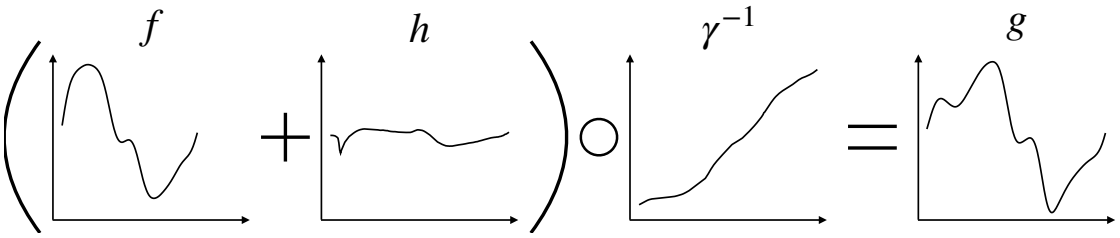


Figure 6.1 Illustration of a relevant deformation mapping f on g with the distortion h and the time parametrization γ^{-1} by the relation: $(f + h) \circ \gamma^{-1}$.

A deformation-based embedding. With the mice experiment as an illustration, we first represent a respiratory cycle signal by its graph, i.e., $\mathsf{G}(\mathsf{I}, f) = \{(t, f(t)) \mid t \in \mathsf{I}\}$. The proposed method learns parametric deformations $(\phi_{\alpha_j})_{j\in[\![1,N]\!]}$ that map a reference respiratory cycle $\mathsf{G}(\mathsf{I}_0, f_0)$ to a set of respiratory cycles $(\mathsf{G}(\mathsf{I}_j, f_j))_{j\in[\![1,N]\!]}$, i.e $\phi_{\alpha_j} \cdot \mathsf{G}(\mathsf{I}_0, f_0) \sim \mathsf{G}(\mathsf{I}_j, f_j)$ where $\phi \cdot \mathsf{G}(\mathsf{I}, f) = \{\phi(t, f(t)) \mid t \in \mathsf{I}\}$. Importantly, the learning procedure is designed so that there exists a unique set of parameters $\boldsymbol{\alpha}_j$ that permits the mapping between $\mathsf{G}(\mathsf{I}_0, f_0)$ and $\mathsf{G}(\mathsf{I}_j, f_j)$, guaranteeing the identifiability of $\mathsf{G}(\mathsf{I}_j, f_j)$ by $\boldsymbol{\alpha_j}$. The resulting set $(\boldsymbol{\alpha}_j)_{j\in[\![1,N]\!]}$ are vectorized embeddings of the respiratory cycles, which can be used in subsequent statistical analysis. Figure 6.2 illustrates the embedding workflow with mice respiratory cycles.

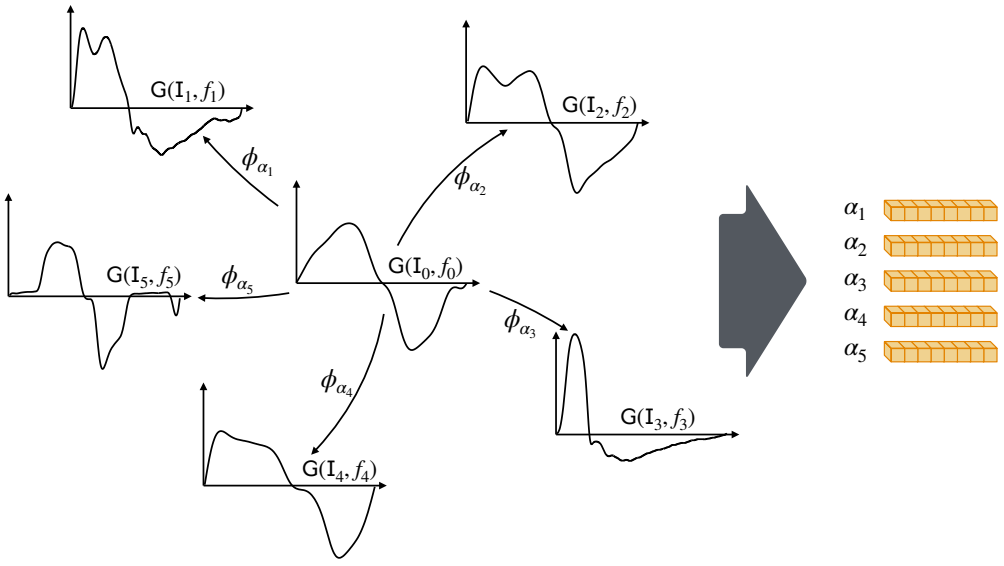


Figure 6.2 Illustrations of the deformation-based embedding workflow on mice respiratory cycles. The deformation ϕ_{α_j} mapping the referent respiratory cycle $\mathsf{G}(\mathsf{I}_0, f_0)$ to an observed respiratory cycle $\mathsf{G}(\mathsf{I}_j, f_j)$ is learned, and its parametrization $\boldsymbol{\alpha_j}$ provides an embedding of the corresponding cycle.

Refining the LDDMM framework. The parametric deformations are diffeomorphisms, i.e. topology-preserving smooth maps with smooth inverse, learned by leveraging

the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework [Beg+05; Vai+04]. Specifically, a diffeomorphic deformation ϕ_{α_j} is learned by integrating an ordinary differential equation parametrized in a Reproducing Kernel Hilbert Space (RKHS). Notably, LDDMM provides a sparse and interpretable parameterization of diffeomorphisms $(\boldsymbol{\alpha}_j)_{j\in[\![1,N]\!]}$, which can be identified with respiratory cylcles $(\mathsf{G}(\mathsf{I}_j, f_j))_{j\in[\![1,N]\!]}$. motivating our choice to propose a method built upon this framework. This characteristic makes it a compelling foundation for the method we propose.

Initially developed for computational anatomy, LDDMM was designed to address spatial geometrical objects, such as images of organs and bones, where diffeomorphisms correspond to anatomically meaningful deformations. However, the action of a general diffeomorphism on graph time series does not necessarily result in a graph time series, see, e.g., Figure 6.3, as a graph time series has more structure than a simple curve [Gla+08]. Our contributions arise from this observation: we specify the class of diffeomorphisms preserving the time series graph structure and show how to learn them. This change is fruitful in representing time series deformations as illustrated in Figure 6.4.

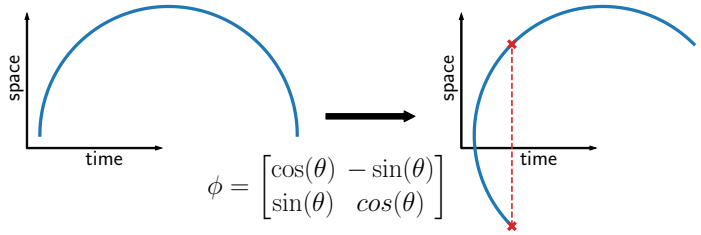


Figure 6.3 A time series' graph $\mathsf{G} = \{(t, f(t)) \mid t \in \mathsf{I}\}$ can lose its structure after applying a general diffeomorphism $\phi \cdot \mathsf{G}$: a time value can be related to two values on the space axis. ϕ is a rotation in this illustration.

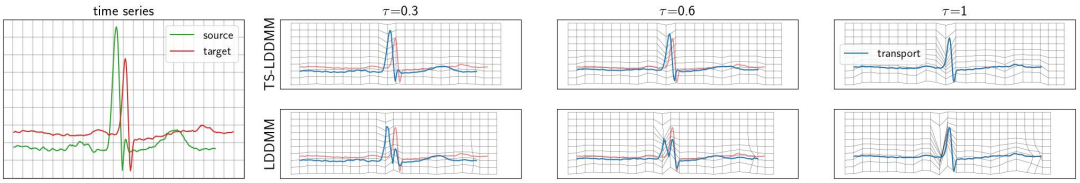


Figure 6.4 LDDMM and TS-LDDMM (our approach) are applied to ECG data. We observe that LDDMM, using a general Gaussian kernel, does not learn the time translation of the first spike but changes the space values, i.e., one spike disappears before emerging at a translated position. At the same time, TS-LDDMM handles the time change in the shape. This difference in deformations implies differences in feature representations.

Notation. We denote by $\mathsf{C}^m(\mathsf{U},\mathsf{E})$ the set of m-times continously differentiable functions defined on an open set U to a normed vector space E, by $\mathsf{D}(\mathsf{O})$ the set of diffeomorpshim defined on an open set O to O, and by $||u||_\infty = \sup_{x\in\mathsf{U}} |u(x)|$ for any bounded function $u : \mathsf{U} \to \mathsf{E}$.

Chapter overview. First, we provide some background on the LDDMM framework and its computation. Specifically, we describe its procedure for building diffeomorphisms by flowing geodesic shooting equations parametrized in an RKHS and its implications in learning linear representations in a shape space. Secondly, we specify our extension of LDDMM, called TS-LDDMM, to the case of time series by providing a class of diffeomorphisms preserving the graph structure of time series and deriving suitable RKHSs for learning such diffeomorphisms. Thirdly, we present the related work. Finally, we showcase the application of TS-LDDMM with the analysis of mice ventilation. For conciseness, ablation studies and benchmarks are summarized and relegated in appendices.

6.2 Background on LDDMM

This section exposes how to learn the diffeomorphisms ϕ_α with the LDDMM framework initially introduced in [Beg+05]. In a nutshell, ϕ_α corresponds to a differential flow related to a learnable velocity field parametrized by α and belonging to a well-chosen Reproducing Kernel Hilbert Space (RKHS). We invite readers interested in the LDDMM framework to read the book [You10], especially chapters 8 to 12.

Let us consider a source shape $\mathbf{x} = (x_i)_{i\in[\![1,n_x]\!]}$ and a target shape $\mathbf{y} = (y_i)_{i\in[\![1,n_y]\!]}$, living in the ambient space $\mathbb{R}^d$. These sets $\mathbf{x}, \mathbf{y}$ usually refer to meshes of continuous objects, e.g., surfaces, curves, images, etc. The basic problem that we consider in this section is the following. We aim to find a unique diffeomorphism ϕ such that the transformation of the source shape by ϕ, i.e. $\phi \cdot \mathbf{x} = (\phi(x_i))_{i\in[\![1,n_x]\!]}$, and the target shape $\mathbf{y}$ are similar according to a well specified data fidelity function $\mathcal{L}$.

6.2.1 Large diffeomorphic deformations

Building large diffeomorphic deformations. Intuitively, LDDMM builds a large diffeomorphic deformation ϕ^v by concatenating infinitesimal small deformations. Formally, ϕ^v is the flow of a time-varying vector field v in $\mathbb{R}^d$ satisfying some smoothness properties and such that for any $x_0 \in \mathbb{R}^d$ and $\tau \in [0,1]$:

$$\frac{dy(\tau)}{d\tau} = v_\tau(y(\tau)), \quad y(0) = x_0, \quad \phi^v_\tau(x_0) = y(\tau), \quad \phi^v = \phi^v_1, \tag{6.1}$$

where $v : \tau \in [0,1] \mapsto v_\tau \in \mathsf{V}$, and V is a Hilbert space of continuously differentiable vector fields in $\mathbb{R}^d$ vanishing at infinity. Following [Gla05, Theorem 5], to guarantee the existence and diffeomorphic nature of ϕ^v, it should exist $c_\mathsf{V} > 0$, such that for any $u \in \mathsf{V}$, $\|u\|_\infty + \|du\|_\infty \leq c_\mathsf{V}\|u\|_\mathsf{V}$, and $v \in \mathsf{L}^2([0,1],\mathsf{V})$ should be square integrable, i.e. $\int_0^1 \|v_\tau\|^2_\mathsf{V} d\tau < \infty$.

If the smoothness conditions hold, the application $\tau \in [0,1] \mapsto \phi^v_\tau$ is a flow of diffeomorphisms with $\phi^v_0 = Id$, $\phi^v_\tau(x_0)$ is the position at τ of the particle that was at the

position x_0 at time $\tau = 0$ by moving along v, and $\phi^v = \phi_1^v$ the diffeomorphic deformation of interest. This procedure offers a general recipe to construct diffeomorphism given a well-specified functional space V. This transport is illustrated in Figure 6.4.

With this in mind, the velocity field v could be estimated by minimizing the functional:

$$v \in \mathsf{L}^2([0,1],\mathsf{V}) \mapsto \mathcal{L}(\phi^v \cdot \mathbf{x}, \mathbf{y}) \in \mathbb{R} \ . \tag{6.2}$$

However, two computational challenges arise. First, this optimization problem is ill-posed; there is no guarantee of the solution uniqueness as several time-varying vector fields v may lead to the same diffeomorphism ϕ mapping $\mathbf{x}$ to $\mathbf{y}$. In addition, a parametric family $\mathsf{V}_\Theta \subset \mathsf{L}^2([0,1],\mathsf{V})$, parameterized by Θ, is sought to efficiently solve this minimization problem.

A group of diffeomorphisms with a right-equivariant metric. Interestingly, the set of diffeomorphisms of $\mathbb{R}^d$:

$$\mathsf{D}_\mathsf{V} = \{\phi^v \mid v \in \mathsf{L}^2([0,1],\mathsf{V})\} \ , \tag{6.3}$$

is a group for which we can define a right-equivariant metric by first defining the application that for any $\phi \in \mathsf{D}_\mathsf{V}$:

$$d(Id,\phi) = \inf_{v \in \mathsf{L}^2([0,1],\mathsf{V})} \left\{ \left(\int_0^1 \|v_\tau\|_\mathsf{V}^2 d\tau \right)^{\frac{1}{2}} \mid \phi^v = \phi \right\} \ . \tag{6.4}$$

The previous infimum is reach for a certain $v^* \in \mathsf{L}^2([0,1],\mathsf{V})$, and the application $d_\mathsf{D}(\phi,\phi') \in \mathsf{D}_\mathsf{V}^2 \mapsto d(Id, \phi' \circ \phi^{-1}) \in \mathbb{R}_+$ is a right-equivariant metric on D_V, leading to the complete metric space $(\mathsf{D}_\mathsf{V}, d_\mathsf{D})$, (proofs [Gla05, Chapter 1]). In addition, the curve $\tau \mapsto \phi_\tau^{v^*}$ can be understood as a geodesic in D_V going from Id to ϕ and conserving its kinetic energy along time [Gla05, Chapter 1]:

$$\forall \tau \in [0,1], \quad \|v_\tau^*\|_\mathsf{V} = \|v_0^*\|_\mathsf{V} \ . \tag{6.5}$$

To summarize, geodesics are minimizers of the total kinetic energy:

$$\inf_{v \in \mathsf{L}^2([0,1],\mathsf{V})} \left\{ \frac{1}{2} \int_0^1 \|v_\tau\|_\mathsf{V}^2 d\tau \mid \phi^v = \phi \right\} \ . \tag{6.6}$$

Therefore, by deriving differential constraints related to the minimum of (6.6) and using Cauchy-Lipschitz conditions, geodesics can be defined solely by giving the initial velocity $v_0 \in \mathsf{V}$ [MTY06]. Denoting by $\tau \mapsto \rho_{v_0}(\tau) \in \mathsf{D}_\mathsf{V}$ the geodesic starting from the Id with initial velocity $v_0 \in \mathsf{V}$, we define the exponential map as

$$\exp_{Id} : v_0 \in \mathsf{V} \mapsto \rho_{v_0}(1) \in \mathsf{D}_\mathsf{V} \ . \tag{6.7}$$

Using $\exp_{Id}(v_0)$ instead of ϕ^v, the previous matching problem becomes a geodesic shooting problem:

$$\inf_{v_0 \in \mathsf{V}} \left\{ \mathcal{L}(\exp_{Id}(v_0) \cdot \mathbf{x}, \mathbf{y}) + \lambda \|v_0\|_\mathsf{V}^2 \right\} \ , \tag{6.8}$$

where $\lambda > 0$ is a balancing factor between the data fidelity term and the "amount" of deformations, as $d(Id, \exp_{Id}(v_0)) = \|v_0\|_{\mathsf{V}}$ by combining (6.5) and (6.4). The existence of a solution is guaranteed under mere conditions on $\mathcal{L}$ by [Cha13, Theorem 1.3.1] and using $\exp_{Id}(v_0)$ instead of ϕ^v for any $v \in \mathsf{L}^2([0,1], \mathsf{V})$ regularizes the problem and induces a sparse representation of the learned diffeomorphisms. The regularizing factor $\|v_0\|_{\mathsf{V}}$ plays an important role in the presence of noisy data to prevent the repercussion of the noise' perturbations on the diffeomorphisms representations, i.e., preventing overfitting. Moreover, by setting V as an RKHS, the geodesic shooting problem has a unique solution and becomes tractable, as described in the next section.

6.2.2 Discrete parametrization of diffeomorpshim.

In this part, V is chosen as an RKHS [BT11] generated by a smooth kernel K (e.g., Gaussian). We follow [DAJ13] and define a discrete parameterization of the velocity fields to perform geodesics shooting (6.8). The initial velocity field v_0 is chosen as a finite linear combination of the RKHS basis vector fields, n_0 control points $\mathbf{c}_0 = (c_{k,0})_{k\in[\![1,n_0]\!]} \in (\mathbb{R}^d)^{n_0}$ and momentum vectors $\boldsymbol{\alpha}_0 = (\alpha_{k,0})_{k\in[\![1,n_0]\!]} \in (\mathbb{R}^d)^{n_0}$ are defined such that for any $x \in \mathbb{R}^d$:

$$v_0(x) = \sum_{k=1}^{n_0} K(x, c_{k,0})\alpha_{k,0} \ . \tag{6.9}$$

In our applications, the control points $(c_{k,0})_{k\in[\![1,n_0]\!]}$ can be understood as the discretized graph $(t_k, f_0(t_k))_{k\in[\![1,n_0]\!]}$ of a starting time series (I, f). With this parametrization of v_0, [MTY06] show that the velocity field v of the solution of (6.8) keeps the same structure along time, meaning that for any $x \in \mathbb{R}^d$ and $\tau \in [0,1]$:

$$v_\tau(x) = \sum_{k=1}^{n_0} K(x, c_k(\tau))\alpha_k(\tau) \ , \tag{6.10}$$

In addition, the system of differential equations governing the geodesic shooting are derived from the Hamiltonian:

$$H : (\mathbf{c}, \boldsymbol{\alpha}) \in \mathbb{R}^{n_0\times d} \times \mathbb{R}^{n_0\times d} \mapsto \sum_{k,l=1}^{n_0} \alpha_k^\top K(c_k, c_l)\alpha_l \in \mathbb{R} \ , \tag{6.11}$$

such that the velocity norm is preserved $||v_\tau||_{\mathsf{V}} = ||v_0||_{\mathsf{V}}$ for any $\tau \in [0,1]$.

$$\begin{cases} \frac{dc_k(\tau)}{d\tau} = v_\tau(c_k(\tau)) \\ \\ \frac{d\alpha_k(\tau)}{d\tau} = -\sum\limits_{k=1}^{n_0} d_{c_k(\tau)}K(c_k(\tau), c_l(\tau))\alpha_l(\tau)^\top\alpha_k(\tau) \end{cases} , \tag{6.12}$$

with initial conditions $c_k(0) = c_{k,0}$, $\alpha_k(0) = \alpha_{k,0}$ for any k in $[\![1, n_0]\!]$.

By (6.12), the velocity field related to a geodesic v^* is fully parametrized by its initial control points and momentum $(x_{k,0}, \alpha_{k,0})_{k\in[\![1,n_0]\!]}$.

A tractable geodesic shooting problem. Assuming a source shape $\mathbf{x} = (x_i)_{i\in[\![1,n_x]\!]}$ and a target shape $\mathbf{y} = (y_i)_{i\in[\![1,n_y]\!]}$, living in the ambient space $\mathbb{R}^d$, a RKHS's kernel $K : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}^{d\times d}$, a data fidelity term on sets $\mathcal{L}$, a numerical integration scheme of ODE and a penalty factor $\lambda > 0$, the basic geodesic shooting step minimizes the following function using a gradient descent method:

$$\mathcal{F}_{\mathbf{x},\mathbf{y}} : \boldsymbol{\alpha} \in (\mathbb{R}^d)^{n_x} \mapsto \mathcal{L}\left(\exp_{Id}(v_0)\cdot\mathbf{x}, \mathbf{y}\right) + \lambda||v_0||^2_{\mathsf{V}} \in \mathbb{R} \ , \tag{6.13}$$

where v_0 is defined by (6.9) and $\exp_{Id}(v_0)\cdot\mathbf{x}$ is the result of the numerical integration of (6.12) using control points $\mathbf{x}$ and initial momentums $\boldsymbol{\alpha}$.

6.2.3 Atlas estimation

Atlas estimation is at the heart of statistical shape analysis by extending the notion of mean and variance to the case of shape [AAT07; Vai+04].

From a computational perspective, let us consider a population of N sampled shapes $(\mathbf{y}^j)j \in [\![1, N]\!]$ living in the ambient space $\mathbb{R}^d$ and potentially of different sampling sizes. The goal of atlas estimation is to learn a referent shape $\mathbf{x}_0 \in (\mathbb{R}^d)^{n_0}$ representing the average shape and the parameterization $(\boldsymbol{\alpha}_0^j)_{j\in[\![1,N]\!]}$ of the diffeomorphisms ϕ^j mapping $\mathbf{x}_0$ to individual shape $\mathbf{y}^j$. The atlas estimation is carried out by solving the minimization problem with gradient descent:

$$\underset{\mathbf{x}_0,\ (\boldsymbol{\alpha}_0^j)_{j\in[\![1,N]\!]}}{\operatorname{argmin}} \quad \sum_{j=1}^{N} \mathcal{F}_{\mathbf{x}_0,\mathbf{y}^j}(\boldsymbol{\alpha}_0^j) \ , \tag{6.14}$$

such that:

$$\mathbf{x}_0 \in (\mathbb{R}^{n_0})^d, \quad \boldsymbol{\alpha}_0^j \in (\mathbb{R}^{n_0})^d \quad \forall j \in [\![1, N]\!] \ .$$

It is important to notice that atlas estimations drastically reduce the complexity of statistical analysis on shapes. Indeed, by solving of (6.14), the non-linear deformations $(\phi_{\alpha_j})_{j\in[\![1,N]\!]}$ mapping the average shape $\mathbf{x}_0$ the observed shapes $(\mathbf{y}^j)_{j\in[\![1,N]\!]}$ are reduced to a linear and identifiable representations $(\boldsymbol{\alpha}_0^j)_{j\in[\![1,N]\!]}$. Therefore, linear statistical and machine learning tools can be leveraged to analyze shapes deformed by non-linear deformations.

Going back to the case of time series, whenever deformations include time warping, they become inevitably non-linear, making statistical analysis at a population level difficult in several cases, including mice ventilation. In such situations, atlas estimation with LDDMM becomes an appealing approach by linearizing complex deformations while guaranteeing the identifiability of the represents. However, LDDMM must be refined to preserve the graph structure of the time series, which will be the topic of the next section.

6.3 Application of LDDMM to time series analysis: TS-LDDMM

This section presents our theoretical contribution: we tailor the LDDMM framework to handle time series data. The reason is that applying a general diffeomorphism ϕ from $\mathbb{R}^{n+1}$ to a time series' graph $\mathsf{G}(\mathsf{I}, f)$ can result in a set $\phi \cdot \mathsf{G}(\mathsf{I}, f)$ that does not correspond

to the graph of any time series, as illustrated in the Figure 6.3. Thus, time series graphs have more structure than a simple curve [Gla+08] and deserve their unique analysis.

To address this challenge, we need to identify an RKHS kernel $K : \mathbb{R}^{n+1} \times \mathbb{R}^{n+1} \to \mathbb{R}^{(n+1)^2}$ that generates deformations preserving the structure of the time series graph. This goal motivates us to clarify, in Theorem 3, a family of diffeomorphisms preserving the graph structure and, subsequently, a class of kernels that produce deformations belonging to this family.

Similarly, selecting a loss function on sets $\mathcal{L}$ that considers the temporal evolution in a time series graph is crucial for meaningful comparisons with time series data. Consequently, we introduce the oriented Varifold distance.

6.3.1 Diffeomorphisms separating space and time.

We prove that two time series graphs can always be linked by a time transformation composed with a space transformation. Moreover, a time series graph transformed by this kind of transformation is always a time series graph. For any $\gamma \in \mathsf{D}(\mathbb{R})$ and $h \in \mathsf{C}^1(\mathbb{R}^{d+1}, \mathbb{R}^d)$, we define the deformations:

$$\begin{cases} \Psi_\gamma \in \mathsf{D}(\mathbb{R}^{d+1}) : (t,x) \in \mathbb{R}^{d+1} \mapsto (\gamma(t), x) \in \mathbb{R}^{d+1} \\ \\ \Phi_h : (t,x) \in \mathbb{R}^{d+1} \mapsto (t, h(t,x)) \in \mathbb{R}^{d+1} \end{cases} . \tag{6.15}$$

As reminder, we denote by $\mathsf{G}(\mathsf{I}, f) = \{(t, f(t)) \mid t \in \mathsf{I}\}$ the graph of a time series $f : \mathsf{I} \to \mathbb{R}^d$ and $\phi \cdot \mathsf{G}(\mathsf{I}, f) = \{\phi(t, f(t)) \mid t \in \mathsf{I}\}$ the action of $\phi \in \mathsf{D}(\mathbb{R}^{d+1})$ on $\mathsf{G}(\mathsf{I}, f)$. We have the following representation theorem.

Theorem 3. *Let $f : \mathsf{J} \mapsto \mathbb{R}^d$ and $f_0 : \mathsf{I}_0 \to \mathbb{R}^d$ be two continuously differentiable time seriess with I_0, J two intervals of $\mathbb{R}$. There exist $h \in \mathsf{C}^1(\mathbb{R}^{d+1}, \mathbb{R}^d)$ and $\gamma \in \mathsf{D}(\mathbb{R})$ such that $\gamma(\mathsf{I}_0) = \mathsf{J}$ and $\Phi_h \in \mathsf{D}(\mathbb{R}^{d+1})$,*

$$\Pi_{\gamma,h} \cdot \mathsf{G}(\mathsf{I}_0, f_0) = \mathsf{G}(\mathsf{J}, f), \ \textit{with} \quad \Pi_{\gamma,h} = \Psi_\gamma \circ \Phi_h \ . \tag{6.16}$$

Moreover, for any $\bar{h} \in \mathsf{C}^1(\mathbb{R}^{d+1}, \mathbb{R}^d)$ and $\bar{\gamma} \in \mathsf{D}(\mathbb{R})$, there exists a continuously differentiable time series $(\bar{\mathsf{I}}, \bar{f})$ such that $\Pi_{\bar{\gamma},\bar{h}} \cdot \mathsf{G}(\mathsf{I}_0, f_0) = \mathsf{G}(\bar{\mathsf{I}}, \bar{f})$.

Proof. Let $f : \mathsf{J} \mapsto \mathbb{R}^d$ and $f_0 : \mathsf{I} \mapsto \mathbb{R}^d$ be two continuously differentiable time series with $\mathsf{I} = (a,b), \mathsf{J} = (\alpha, \beta)$ two intervals of $\mathbb{R}$. By setting $\gamma : t \in \mathbb{R} \mapsto (\beta-\alpha)(t-a)/(b-a)+\alpha \in \mathbb{R}$, we have $\gamma(\mathsf{I}) = \mathsf{J}$ and $\gamma \in \mathsf{D}(\mathbb{R})$. By defining $h : (t,x) \in \mathbb{R}^{d+1} \mapsto x - f_0(t) + f \circ \gamma(t)$, the map $\Phi_h \in \mathsf{D}(\mathbb{R}^{d+1})$, as its inverse is $\Phi_h^{-1} : (t,x) \in \mathbb{R}^{d+1} \mapsto (t, x+f_0(t)-f(t))$ and is continuously differentiable. Moreover, we have $\Pi_{\gamma,h} \cdot \mathsf{G}(f_0) = \{(\gamma(t), f \circ \gamma(t)) \mid t \in \mathsf{I}\} = \mathsf{G}(f)$.

Let $\bar{h} \in \mathsf{C}^1(\mathbb{R}^{d+1}, \mathbb{R}^d)$, $\bar{\gamma} \in \mathsf{D}(\mathbb{R})$ and $f_0 \in \mathsf{C}^1(\mathsf{I}, \mathbb{R}^d)$ with I an interval of $\mathbb{R}$. We have :

$$\Pi_{\gamma,f} \cdot \mathsf{G}(f_0) = \{(\gamma(t), h(t, f_0(t))) \mid t \in \mathsf{I} \} \tag{6.17}$$

$$= \{(t, h\left(\gamma^{-1}(t), f_0(\gamma^{-1}(t))\right)) \mid t \in \gamma(\mathsf{I}) \} \ . \tag{6.18}$$

By defining $\bar{f} : t \in \gamma(\mathsf{I}) \to h\left(\gamma^{-1}(t), f_0(\gamma^{-1}(t))\right)$, we have $\bar{f} \in \mathsf{C}^1(\gamma(\mathsf{I}), \mathbb{R}^d)$ by composition of continuous functions and $\mathsf{G}(\bar{f}) = \Pi_{\gamma,h} \cdot \mathsf{G}(f_0)$ by (6.18), which concludes the proof. □

Remark 3. *Note that for any $\gamma \in \mathsf{D}(\mathbb{R})$, $h \in \mathsf{C}^1(\mathbb{R}^{d+1})$, and $f \in \mathsf{C}^1(\mathsf{I}, \mathbb{R}^d)$:*

$$\Pi_{\gamma,h} \cdot \mathsf{G}(\mathsf{I}, f) = \{(\gamma(t), f(t) + h(t, f(t)) - f(t)) \mid t \in \mathsf{I}\} \tag{6.19}$$

As a result, γ can be understood as a time parametrization and $\tilde{h} : t \in \mathsf{I} \mapsto h(t, f(t)) - f(t) \in \mathbb{R}^d$ as the distortion of the time series (I, f) which is congruent with the group action depicted in Equation (1.9).

6.3.2 Kernels preserving time and space separation

As depicted on Figure 6.3-6.4, we must use specific kernels K to apply the previous methodology when learning deformations on time series graphs. Diffeomorphisms separating time and space preserve the graph structure, and in this section, we describe kernels of the RKHS V generating such diffeomorphism.

We denote the one-dimensional Gaussian kernel by $K_\sigma^{(a)}(x, y) = \exp(-\|x - y\|^2/\sigma)$ for any $(x, y) \in (\mathbb{R}^a)^2$, $a \in \mathbb{N}$ and $\sigma > 0$.

To solve the geodesic shooting problem (6.13) on $\mathbb{R}^{d+1}$, we consider for V the RKHS associated with the kernel defined for any $(t, x), (t', x') \in (\mathbb{R}^{d+1})^2$:

$$K_{\mathsf{V}}((t, x), (t', x')) = \begin{pmatrix} c_0 K_{\text{time}} & 0 \\ 0 & c_1 K_{\text{space}} \end{pmatrix}, \tag{6.20}$$

with:

$$\begin{cases} K_{\text{space}} = K_{\sigma_{T,1}}^{(1)}(t, t') K_{\sigma_x}^{(d)}(x, x') I_{\mathbb{R}^d} \\ K_{\text{time}} = K_{\sigma_{T,0}}^{(1)}(t, t') \end{cases}, \tag{6.21}$$

parametrized by the widths $\sigma_{T,0}, \sigma_{T,1}, \sigma_x > 0$ and the constants $c_0, c_1 > 0$.

Lemma 1. *If we denote by V the RKHS associated with the kernel K_{V}, then for any vector field v generated by (6.12) with v_0 satisfying (6.9), there exist $\gamma \in \mathsf{D}(\mathbb{R})$ and $h \in \mathsf{C}^1(\mathbb{R}^{d+1}, \mathbb{R}^d)$ such that $\phi^v = \Psi_\gamma \circ \Phi_h$.*

Proof. Let v be a vector field generated by (6.12) with v_0 satisfying (6.9). We remark that the first coordinate of the velocity field v_τ denoted by v_τ^{time} only depends on the time variable t for any $\tau \in [0, 1]$. Thus, when computing the first coordinate of the deformation ϕ^v, denoted by γ, we integrate (6.1) with v_τ replaced by v_τ^{time}, thus γ is independant of the variable x. Moreover, $\gamma \in \mathsf{D}(\mathbb{R})$ since a Gaussian kernel induced an Hilbert space V satisfying $\|u\|_\infty + \|du\|_\infty \leq \|u\|_V$ for any $u \in \mathsf{V}$ by [Gla05, Theorem 9]. For the same reason, we have $\phi^v \in \mathsf{D}(\mathbb{R}^{d+1})$, and thus its last coordinates denoted by h belongs to $\mathsf{C}^1(\mathbb{R}^{d+1}, \mathbb{R}^d)$, and by construction $\phi^v = \Psi_\gamma \circ \Phi_h$. □

Instead of Gaussian kernels, other types of smooth kernels can be selected as long as the structure (6.20) is respected.

Remark 4. *With this choice of kernel, the features associated with the time transformation can be extracted from the momentums $(\alpha_{k,0})_{k \in [\![1, n_0]\!]} \in (\mathbb{R}^{d+1})^{n_0}$ in (6.9) by taking the coordinates related to time. However, the features related to the space transformation are not only in the space coordinates since the related kernel K_{space} depends on time as well.*

We provide guidelines for setting the hyperparameters $(\sigma_{T,0}, \sigma_{T,1}, \sigma_x, c_0, c_1)$ in Appendix B.2.

6.3.3 A data fidelity term for time series.

This section specifies the distance function $\mathcal{L}$ introduced in the loss function defined in (6.13).

In practice, we can only access discretized graphs of time series, $(t_i^j, f_i^j)_{i\in[\![1,n_j]\!]}$ for any $j \in [\![1, N]\!]$, potentially of different sizes n_j and sampled at different timestamps $(t_i^j)_{i\in[\![1,n_j]\!]}$ for any $j \in [\![1, N]\!]$. Usual metrics, such as the Euclidean distance, are not appealing as they make the underlying assumptions of equal-size sets and the existence of a pairing between points. Distances between measures on sets (taking the empirical distribution), such as Maximum Mean Discaprency (MMD) [DRG15; Bor+06], alleviate those issues; however, MMD only accounts for positional information and lacks information about the time evolution between sampled points. A classical data fidelity term from shape analysis corresponding to the distance between oriented varifolds associated with curves alleviates this last issue [KCC17]. Intuitively, an oriented varifold is a measure that accounts for positional and tangential information about the underlying curves at sample points. More details and information about oriented varifolds can be found in Appendix B.1.

From a numerical perspective, the oriented varifold measure is embedded in the dual W^* of an RKHS W with a kernel $k : (\mathbb{R}^{d+1} \times \mathbb{S}^d)^2 \mapsto \mathbb{R}$ verifying verifying [KCC17, Proposition 2 & 4]. Given a time series graph set $G = (g_i)_{i\in[\![1,n]\!]} \in (\mathbb{R}^{d+1})^n$, it is map to the set $(l_i, p_i, \overrightarrow{v_i})_{i\in[\![1,n-1]\!]}$ defined by:

$$\left\{\begin{array}{lcll} l_i & = & \|g_{i+1} - g_i\| & \\ p_i & = & (g_i + g_{i+1})/2 & , \quad \forall i \in [\![1, n-1]\!] \ , \\ \overrightarrow{v_i} & = & (g_{i+1} - g_i)/\|g_{i+1} - g_i\| & \end{array}\right. \tag{6.22}$$

and its embedding as oriented varifold is the measure:

$$\mu_G = \sum_{i=1}^{n-1} l_i \delta_{(p_i, \overrightarrow{v_i})} \ . \tag{6.23}$$

Therefore, given two time series graph sets $G_0 \in (\mathbb{R}^{d+1})^{n_0}$ and $G_1 \in (\mathbb{R}^{d+1})^{n_1}$, the data fidelity term is defined as:

$$\begin{array}{rcl} \mathcal{L}_{\mathsf{W}^*}(G_0, G_1) & = & \|\mu_{G_0} - \mu_{G_1}\|^2_{\mathsf{W}^*} \\ & = & \sum\limits_{i,j=1}^{n_0-1} l_i^0 k((p_i^0, \overrightarrow{v_i}^0), (p_j^0, \overrightarrow{v_j}^0)) l_j^0 + \sum\limits_{i,j=1}^{n_1-1} l_i^1 k((p_i^1, \overrightarrow{v_i}^1), (p_j^1, \overrightarrow{v_j}^1)) l_j^1 \\ & & -2 \sum\limits_{i=1}^{n_0-1} \sum\limits_{j=1}^{n_1-1} l_i^0 k((p_i^0, \overrightarrow{v_i}^0), (p_j^1, \overrightarrow{v_j}^1)) l_j^1 \end{array}$$

In practice, we set the kernel k as the product of two anisotropic Gaussian kernels, k_{pos} and k_{dir}, such that for any $(x, \overrightarrow{u}), (y, \overrightarrow{v}) \in (\mathbb{R}^{d+1} \times \mathbb{S}^d)^2$:

$$k((x, \overrightarrow{u}), (y, \overrightarrow{v})) = k_{\text{pos}}(x, y) k_{\text{dir}}(\overrightarrow{u}, \overrightarrow{v}) \ . \tag{6.24}$$

Note that the loss kernel k has nothing to do with the velocity field kernel denoted by K_{V} or other K specified in Section 6.3.2. Finally, we define the data fidelity term, $\mathcal{L}$, as a sum of one ore $\mathcal{L}^2_{\mathsf{W}^*}$ using different kernel's width parameters σ to incorporate multiscale information. $\mathcal{L}$ is indeed differentiable with respect to its first variable. The specific kernels $k_{\text{pos}}, k_{\text{dir}}$ that we use in our experiments are given Appendix B.1. For further readings on curves and surface representation as varifolds, readers can refer to [KCC17; CT13].

6.4 Related Works

The following paragraphs present unsupervised shape-related methods for embedding time series of variable length and sampled irregularly. We present works from both shape analysis and deep learning for time series.

From shape analysis. LDDMM framework is a relevant shape analysis framework to represent curves as depicted in [Gla+08]. However, graphs of time series are a well-structured type of curve due to the inclusion of the temporal dimension that requires specific care (Figure 6.3). Similarly, Qiu *et al* [Qiu+09] proposes a method for tracking anatomical shape changes in serial images using LDDMM. They include temporal evolution, but not for the same purpose: the aim is to perform longitudinal modeling of brain images.

Leaving the LDDMM representation, the results of [Sri+10; Heo+24] address the representation of curves with the Square-Root Velocity (SRV) representation. However, the SRV representation is applied after parametrizing the temporal dimension on the unit length segment. Consequently, the graph structure of the time series is not respected, and the original time evolution of the time series is not encoded in the final representation. Very recently, in a functional data analysis framework, a paper [WHS24] (Shape-FPCA) improved by representing the original time evolution. Nevertheless, this method is made for continuous objects and only applies to time series of the same length, making the estimation more sensitive to noise and interpolation procedures.

From deep learning for time series. Balancing between discrete and continuous elements is a challenging task. In the deep learning literature [Che+18; Kid+20; TR19; JB19; Liu+19; Ans+23], Neural Ordinary Differential Equations (Neural ODEs) [Che+18] learn continuous latent representations using a vector field parameterized by a neural network, serving as a continuous analog to Residual Networks [ZK16]. This approach was further enhanced by Neural Controlled Differential Equations (Neural CDEs) [Kid+20] for handling irregular time series, functioning as continuous-time analogs of RNNs [SP97]. Extending Neural ODEs, Neural Stochastic Differential Equations (Neural SDEs) introduce regularization effects [Liu+19], although optimization remains challenging. Leveraging techniques from continuous-discrete filtering theory, Ansari et al. [Ans+23] applied successfully Neural SDEs to irregular time series. Oh *et al.* [OLK24] improved these results by incorporating the concept of controlled paths into the drift term, similar to how Neural CDEs outperform Neural ODEs.

All these state-of-the-art methods previously mentioned [Gla+08; OLK24; WHS24; Heo+24] are compared to TS-LDDMM in Appendix B.6 and Appendix B.7.

6.5 Experiment

In addition to the study of mice ventilation with the TS-LDDMM framework, we also performed several experiments relegated in appendices and summarized in the next section for conciseness. Specifically, we studied the parameters' influences on the representation identifiability. We also evaluated the robustness of the TS-LDDMM representation to irregular sampling, and we compared classifiers based on TS-LDDMM representations with other shaped-based methods on a classification task.

6.5.1 Summary of additional experiments

1. **TS-LDDMM representation identifiability, Appendix B.5:** On synthetic data, we evaluate the ability of our method to retrieve the parameter v_0^* that encodes the deformation $\exp_{Id}(v_0^*)$ acting on a time series graph G by solving the geodesic shooting problem (6.13) between G and $\exp_{Id}(v_0^*) \cdot \mathsf{G}$. **Results** show that TS-LDDMM representations are identifiable or weakly identifiable depending on the velocity field kernel K_{V} specification.

2. **Robustness to irregular sampling, Appendix B.6:** We compare the robustness of TS-LDDMM representation with 9 URL methods handling irregularly sampled multivariate time series on 15 shape-based datasets (7 univariates & 8 multivariates). We assess methods' classification performances under regular sampling (0% missing rate) and three irregular sampling regimes (30%, 50%, and 70% missing rates), according to the protocol depicted in [Kid+20]. **Results** show that our method, TS-LDDMM, outperforms all methods for sampling regimes with missing rates: 0%, 30%, and 50%.

3. **Classification benchmark on regularly sampled datasets, Appendix B.7:** We compare performances of a kernel support vector machine (SVC) algorithm based on TS-LDDMM representation with 3 state-of-the-art classification methods from shape analysis on 15 shape-based datasets (7 univariates & 8 multivariates). **Results** show that the TS-LDDMM-based method outperforms other methods (best performances over 13 datasets), making TS-LDDMM representation relevant for time series shape analysis.

6.5.2 Application to mice ventilation analysis

A reminder. As a reminder from Chapters 4 and 5, respiration is responsible for supplying O2 and eliminating CO2. Its proper functioning depends on the nervous system's precise and coordinated control of the ventilation muscles. The enzymes AChE and BChE play a crucial role in this system by synchronizing nervous system signals with muscle activity. Certain drugs inhibit AChE/BChE activity, leading to severe respiratory dysfunctions, which can be fatal. The underlying causes of these dysfunctions are not

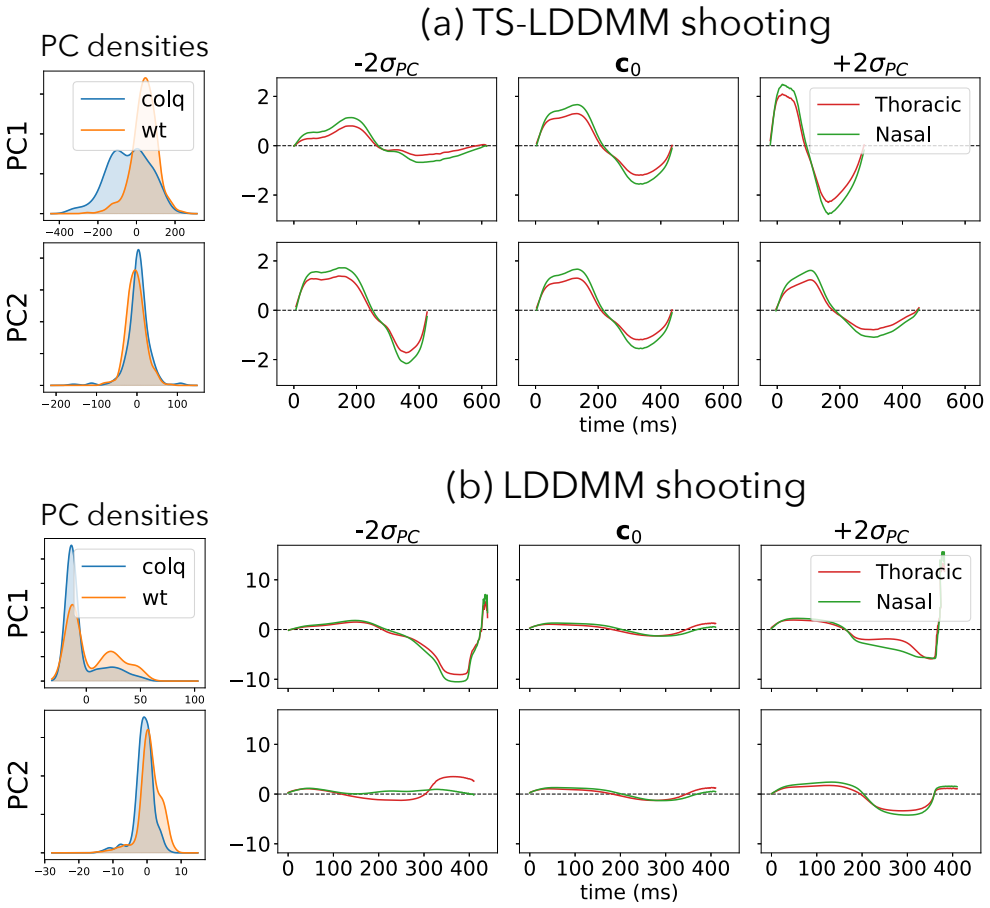
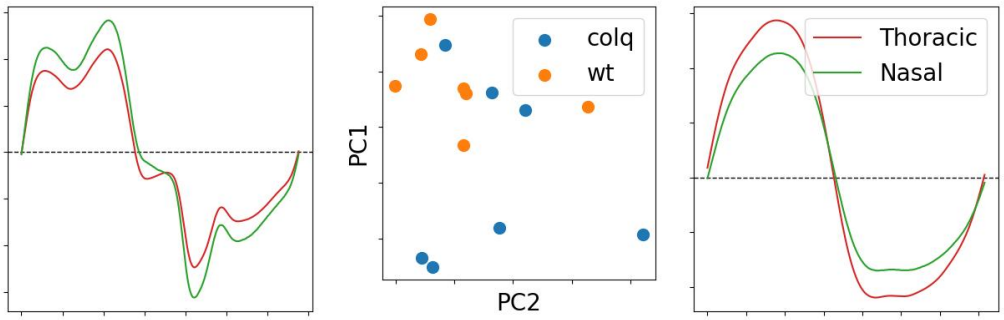


Figure 6.5 Analysis of the two principal components (PC) related to mice ventilation before exposure with TS-LDDMM representations **(a)**, and LDDMM **(b)**. In both cases and for all PC, the left plot displays PC densities according to mice genotype and right plot displays deformations of the reference graph $\mathbf{c}_0$ along each PC.

yet fully understood, making it an active area of research aimed at developing effective treatments. One approach to investigating this complexity involves using genetics to remove AChE/BChE in specific tissues and locations selectively. In the experiment of interest, the respiration of mice with different genotypes is recorded via plethysmography during a control period before exposure to an inhibitor, followed by monitoring the evolution of their respiratory patterns. We aim to use shape-based unsupervised methods to infer distinct ventilation modalities from respiratory cycles. These modalities provide insights, at least partially, into the mechanisms at play at a molecular scale, and by analyzing their evolution over time and across genotypes, we aim to validate or refute various hypotheses regarding the effects of AChE/BChE inhibition.

In the previous chapter, we analyzed plethysmography signals using a symbolic representation of respiratory cycles with a DTW-based clustering approach. As a baseline, this method produced physiologically meaningful results, identifying ventilation modalities



(a) colq cycle **(b)** PC1 vs PC2 **(c)** wt cycle

Figure 6.6 **(a)** an example of ColQ respiratory cycle. **(b)** Referent respiratory cycle of individual mouse $\mathbf{c}_0^j$ in the TS-LDDMM PC1-PC2 coordinates system of $\mathbf{c}_0$. **(c)** an example of WT respiratory cycle.

specific to different genotypes or drug exposure, such as prolonged pauses after inspiration or motor control impairment in ColQ mice. It also revealed heterogeneous adaptation patterns in mutant mice with genotypic ChE deficiencies. In many ways, this initial analysis overcame the limitations of previous methods, which relied solely on basic descriptors that could not capture the diversity of observed modalities. In this chapter, we build upon this foundation by employing vectorized representations of respiratory cycles, focusing on the entire deformation mapping to a reference cycle. This is achieved by adapting the LDDMM framework to the time series case, and in this section, we explore the potential of these representations.

Experimental protocol. We considered two experimental scenarios; the first focuses on mice ventilation before exposure to explore the inter-individual variability and genotype-specific ventilation modalities. The second is similar to the previous chapter and focuses on whole recordings to analyze the evolution of mice's ventilation after exposure to a ChE inhibitor. We only considered two genotypes for both scenarios: the control group (WT) and the mutant ColQ presenting AChE deficiency in neuromuscular junctions. In both cases, the baseline protocol was the following:

1. Creating the dataset by extracting N respiratory cycles with the procedure described in Section 4.1.3.

2. Learning the referent respiratory cycle $\mathbf{c}_0$ and the representations of respiratory cycles $(\boldsymbol{\alpha}_0^j)_{j\in[\![1,N]\!]}$ by solving (6.14) using TS-LDDMM. $\boldsymbol{\alpha}_0^j$ being the momentum of the initial velocity field of the geodesic encodings the diffeomorphisms mapping $\mathbf{c}_0$ to the j^{th} respiratory cycle.

3. Performing a Kernel-PCA on the initial velocity fields (6.9) belonging to V and encoded by the pairs $(\boldsymbol{\alpha}_0^j, \mathbf{c}_0)_{j\in[\![1,N]\!]}$.

In addition, we performed the experimental protocol with LDDMM representation for the first scenario to compare TS-LDDMM with LDDMM. The first experiment includes

$N_1 = 700$ respiratory cycles collected before exposure. The second experiment includes $N_2 = 1400$ respiratory cycles with 25% (resp. 75%) before (resp. after) exposure. The signals were down-sampled to 1,000Hz, and Appendix B.8 describes settings for TS-LDDMM and LDDMM. Essentially, varifold losses are identical for both methods, and the velocity field kernels are set to encompass time and space scales.

Geodesic shooting along principal component directions. When performing a kernel-PCA in the space of initial velocity fields V, any principal component (PC) v_0^{pc} is itself an initial velocity field encoded by a pair $(\mathbf{c}_0, \boldsymbol{\alpha}_0^{pc})$. PCs encode the principal axis of deformations, and it is possible to shoot along the geodesic they encode as depicted in (6.12). Performing shootings along PC directions with amplitudes related to their variance permits the interpretation of the main direction of deformations.

Mice ventilation before exposure. We focus on the analysis of the two first Principal Components (PC) for TS-LDDMM (Figure 6.5a) and LDDMM (Figure 6.5b). Looking at the geodesic shooting along PCs, Figure 6.5 shows that principal components learned with TS-LDMM lead to deformations that remain respiratory cycles. In contrast, deformations learned with LDDMM are challenging to interpret as respiratory cycles. The LDDMM velocity field kernel is a Gaussian anisotropic kernel that accounts for time and space scales; however, the entanglement of time and space dimensions in the kernel does not guarantee the graph structure, and it makes the convergence of the method complex (relative varifold loss error: TS-LDDMM: 0.06, LDDMM: 0.11).

Concerning TS-LDDMM Figure 6.5a, its PCs refer to deformations directions carrying different physiological meanings. Indeed, the geodesic shooting along these directions indicates that PC1 accounts for variations of the total duration of a respiratory cycle, while PC2 expresses the trade-off between inspiration and expiration duration. In addition, the distribution of ColQ respiratory cycles along PC1 is wider than in WT mice, which is in congruence with observation of the previous chapter where we have seen that mutant mice, like ColQ, inter-individual adaptation to their ChE deficiency is variable. This observation can also be seen in Figure 6.6b where a referent respiratory cycle $\mathbf{c}_0^j$ is learned by atlas estimation (6.14) for each mouse and encoded in the (PC1,PC2) coordinate system of $\mathbf{c}_0$ by registration (6.8). Indeed, the average respiratory cycles of ColQ mice are more spread out than those of WT mice. Going back to the densities of PC1, ColQ mice distribution has a heavier tail toward negative values compared to WT mice. When shooting in the opposite direction of PC1, we can observe that the inspiration is divided into two steps. As seen in the previous chapter, an inspiration in two steps indicates motor control difficulties specifically for ColQ mice as they have a ChE deficiency in neuromuscular junctions. Figure 6.6a is an example of ColQ respiratory cycle with negative PC1 coordinate.

Mice ventilation evolution after exposure to a ChE inhibitor. This experiment only focuses on the first principal components learned from TS-LDDDM representations of respiratory cycles randomly sampled before and after inhibitor exposure. Figure 6.7a illustrates the geodesic shootings along PC1. Again, PC1 accounts for variations in

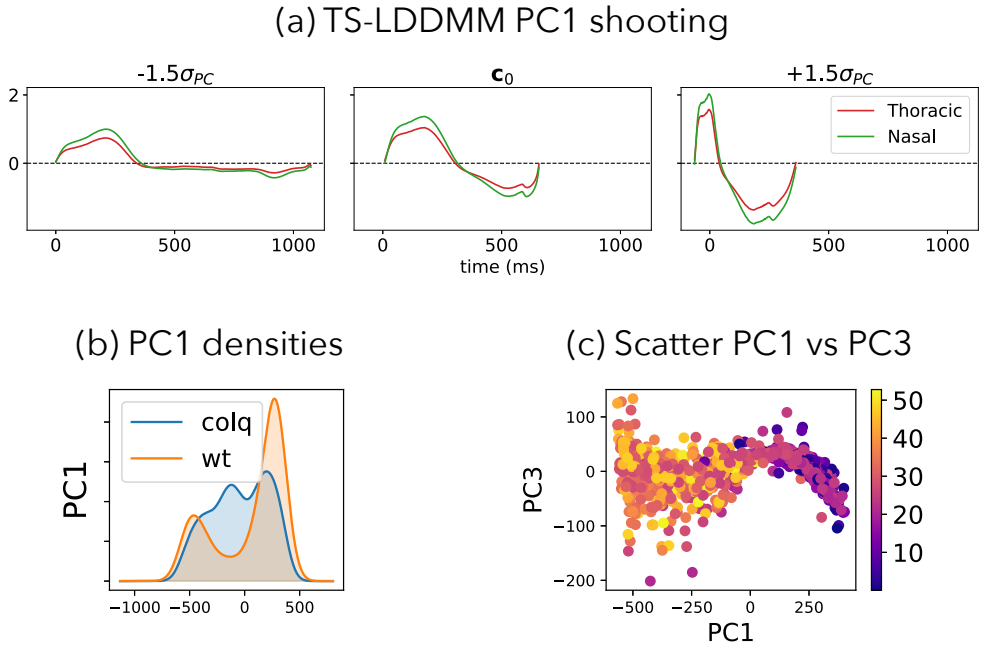


Figure 6.7 Analysis of the first Principal Component (PC1) related to mice ventilation before and after exposure with TS-LDDMM representations. **(a)** displays PC densities per mice genotype, **(b)** illustrates deformations of the reference respiratory cycle $\mathbf{c}_0$ along PC1, and **(c)** displays all respiratory cycles with respect to time in PC1 and PC3 coordinates

respiratory cycle duration, but more importantly, it can be observed on the deformation at -1.5 σ_{PC} the apparition of a long pause after inspiration. This phenomenon was also observed in the previous chapter and was prevalent in WT mice. Congruently, Figure 6.7c indicates that pauses appear after inhibitor exposure as cycles with negative PC1 values mainly occur after 20 minutes and present more variability along PC3. In addition, Figure 6.7b shows a bimodal distribution for WT mice with one of the peaks in the negative values. This peak was not observed in the previous experiment Figure 6.5a. It indicates that pause after inspiration is a prevalent ventilation modality in WT mice after inhibitor exposure.

In the same way, in the previous chapter, we observed that ColQ mice were less affected by inhibitor exposure than WT mice. This difference in reaction is probably due to their habituation to AChE deficiency in neuromuscular junctions. Similarly, the distributions of ColQ mice's respiratory cycles along PC1 in both experiments are similar and account for the same deformation, suggesting that ColQ mice weakly react to the exposure of ChE inhibitors.

Experiment Conclusion. The straightforward analysis of mice ventilation using TS-LDDMM representations highlights the method's ability to facilitate meaningful interaction between experts and the data. Notably, the principal deformations learned through TS-LDDMM in the context of mice ventilation reveal physiologically significant deformations. The statistical and visual interpretation of these deformations enabled

the characterization of some mice genotypes, ventilation modalities, and the effects of inhibitor exposure.

6.6 Conclusion

This chapter introduced TS-LDDMM, an unsupervised shape-based embedding method for time series. An essential contribution was specifying a class of diffeomorphisms that preserve the graph structure of time series, which can be learned using the LDDMM framework. Given a reference time series, a diffeomorphism is fully determined by its initial conditions through geodesic shooting equations. By learning the diffeomorphisms mapping the referent time series to a set of variable length time series and irregularly sampled, the initial conditions provide linear representations of time series. This approach enables the application of linear statistical learning methods in subsequent analysis.

Experiments demonstrated that classifiers based on representations learned with TS-LDDMM outperformed traditional shape-analysis methods, deep learning, and machine learning approaches for shape-related classification tasks involving irregularly sampled time series. Furthermore, applying atlas estimation with TS-LDDMM to analyze mice ventilation yielded promising results, offering statistically grounded and interpretable insights into ventilation modalities, genotype differences, and the effects of inhibitor exposure.

Chapter 7

Conclusion & Perspectives

Conclusion

This thesis tackled challenges related to shape-based comparisons in biomedical temporal data, where deterministic patterns are essential for statistical analysis. The work is divided into two parts: the first focuses on searching and discovering such patterns, while the second concentrates on comparing them.

Searching or discovering patterns. The first part addressed the problem of searching or discovering deterministic patterns in long time series with distances independent of some irrelevant sources of variability modeled with a group of deformations. The presented methods prioritize interpretability, high efficiency, and ease in modeling the group of deformations in order to facilitate meaningful interactions between data and biomedical researchers. To that end, a general framework has been proposed to build deformation-invariant distances, which can be plugged into state-of-the-art algorithms for similarity search and motif discovery without sacrificing efficiency. Specifically, when sources of variability can be modeled by a group of deformations acting on time series as a vector subspace, it is possible to create a deformation-invariant embedding, and the resulting distance is the Euclidean distance between embeddings. This framework generalizes the well-known Z-normalized Euclidean distance and has shown great success in several biomedical use cases.

Additionally, an interpretable and interactive algorithm for motif discovery has been proposed. This algorithm represents a time series through a diagram whose visual interpretation allows the identification and extraction of repeated patterns. This algorithm outperforms existing motif discovery algorithms on a biomedical benchmark, and an application leveraging its interpretability and efficiency has been proposed to allow meaningful interaction between the data and biomedical researchers.

Comparing patterns. The second part addressed the problem of comparing deterministic patterns. The proposed methods were motivated by the problem of comparing mice respiratory cycles recorded by plethysmography and necessitating more complex groups of deformations to handle time warping. The aim was to identify mice's ventilation

modalities and the breathing evolution when mice of different genotypes are exposed to a drug. A first approach, intended as a baseline, compares respiratory cycles with a DTW-based clustering algorithm, leading to a shape-based symbolic representation where each symbol corresponds to a cluster. Experimental results have shown that clusters can be connected to genotype-related ventilation modalities with genotype-dependent response to the drug exposition.

As an improvement toward a more statistically founded method, the second approach built a fixed-size vector representation of irregularly sampled and variable length time series with the vectors parametrizing the deformations mapping a referent time series to the observed time series. This method draws on Large Deformation Diffeomorphic Metric Mapping (LDDMM), a framework from shape analysis. The LDDMM framework was refined to ensure the spatiotemporal nature of deformed time series while guaranteeing the bijectivity of the embedding method. This method offers statistical results and a visual interpretation of shapes and deformations. Conducting simple statistical analysis in the respiratory cycle embedding space has shown that the principal axes of deformations carry physiological meanings that are informative about mice ventilation modalities dependent on the genotype and the drug exposure.

Perspectives

Refining similarity search and motif discovery for multivariate time series

This thesis explores shape-based similarity search and motif discovery for multivariate time series under the simplifying assumption that deterministic patterns are present across all channels. However, this assumption breaks down in several cases, particularly when patterns only appear in a subset of channels, as seen in various biomedical applications [YKK17; Min+07]. Specifically, the definition of shape and deformations must be refined to account for the variability inherent in multivariate time series while focusing on interpretability and efficiency.

Improving mice ventilation analysis with TS-LDDMM

The potential of TS-LDDMM for analyzing mice ventilation can be explored further in two directions:

- **Reducing inter-individual variability:** As we have observed, ventilation varies between mouse and mouse. It can be due to physiological differences or adaptation to some deficiency, like mutant mice. These inter-individual variations can limit the analysis of ventilation modalities. The LDDMM parallel transport along geodesics could be useful for addressing this issue by aligning the data across individuals [Pir+21; Lou+17].

- **Exploring ventilation dynamic from TS-LDDMM representations with time series algorithms:** Unlike symbolic embedding, TS-LDDMM representations retain much information about respiratory cycles in a vectorized way. Once embedded, plethysmography signals result in a set of multivariate time series, which

can be analyzed using a wide range of time series algorithms developed in machine learning. It opens the door to possibilities such as identifying anomalies [SWP22], discovering motifs [TL17], or performing change point detection [TOV20], among many other techniques.

Computational behavior analysis

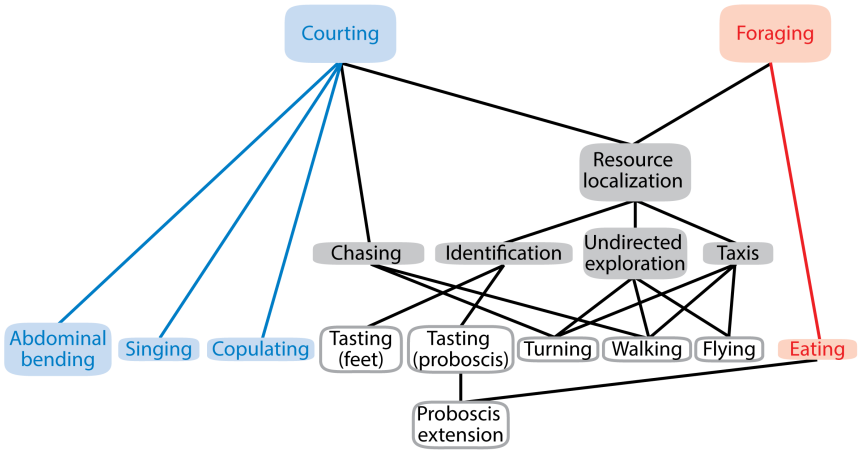


Figure 7.1 Hierarchical organization of Fruit flies' behavior, from [EB16].

Mice ventilation analysis is closely related to the emerging field of computational behavior analysis for animals [EB16; Gom+14]. Rooted in ethology, this discipline quantitatively analyzes animal survival and social behaviors observable from an external eye [Cas+15]. The central concept is that behaviors can be modeled as sequences of simpler actions and organized in a hierarchical order of complexity. The simplest behaviors are basic actions called behavioral primitives. For example, the courtship behavior combines simpler behaviors like chasing or singing, which can also be broken down into even smaller behaviors, as depicted in Figure 7.1.

Computer vision requirements. Animal behavior analysis primarily concentrates on posture-based behaviors, as videos offer a non-invasive and accessible means of observation. For instance, a study [Ber+14] identified sex-specific flying and grooming behaviors of fruit flies by filming them in a controlled environment. An important preprocessing step toward behavior analysis consists of extracting animals' poses from video footage, and several methods have been proposed to tackle this problem with recent advancements in deep learning [Ye+24; Lau+22; MM20; Mat+18].

Behavior analysis methodology. Once the time series of animals' poses have been extracted, the approach to behavior analysis follows a framework identical to the one of mice ventilation analysis Section 4.1 involving time series segmentation, sequence embedding, and analysis of the embedded time series. To leverage the hierarchical

organization of behaviors from primitives, researchers have focused on unsupervised methods that segment a time series in primitive actions and embed them with symbols, i.e., each primitive action corresponds to a symbol. Currently, most behavior studies involving symbolic representations remain at the scale of primitives and analyze their distribution at a cohort level. For instance, the pioneer method Moseq [Wil+20] uses an autoregressive hidden Markov model to infer the primitives and the segmentation of freely moving mice in a closed environment, with the hidden states as behavioral primitives. Mice were exposed to different drugs, and by comparing the primitive distributions, the study reveals that some primitives only manifest when mice are exposed to a specific drug.

Potential benefits from shape analysis and machine learning for time series. Current methods in computational behavior analysis can be enhanced by integrating shape analysis and machine learning techniques for time series in several key areas:

- **Video Embedding:** Poses of subjects are a crucial feature in video-based behavioral studies, typically well-estimated by the latest deep learning algorithms [Ye+24; Zhe+23]. By incorporating shape analysis, these poses can be embedded into a shape space, enabling comparisons independent of irrelevant sources of variability.
- **Discovery of Behavior Primitives:** Machine learning methods for time series, such as anomaly detection [SWP22], segmentation [TOV20], and motif discovery [TL17], can be employed to identify specific patterns and regimes that may hold significance in behavioral analysis. The discovered patterns or regimes can be the seeding datasets to learn estimator for detecting primitives with active learning [Mos+23; KG20; Elh+13].
- **Hierarchical Behavior Analysis:** The dynamic analysis of symbolic representations and the hierarchical organization of behaviors is an active area of research [Zin+24; Wei+24; Mag20]. This field could benefit from advances in symbolic time series representation [Com24; Sen+18], and improvements in T-pattern analysis [Sal+10; TSP08].

Chapitre 8

Introduction (en français)

Points clés :

1. Les séries temporelles sont courantes dans les applications biomédicales où elles présentent souvent des formes déterministes, récurrentes ou anormales précieuses en analyse statistique en raison de leur apparence cohérente entre différents sujets. En revanche, une utilisation précise et efficace de ces formes nécessite des outils mathématiques appropriés.

2. La comparaison des formes temporelles se situe à l'intersection entre apprentissage automatique pour les séries temporelles et analyse des formes où les sources de variabilité sont modélisées comme des déformations d'une forme de référence. Malgré le grand succès des travaux issus de l'apprentissage automatique pour les séries temporelles qui s'apparentent à la notion de forme, ce cas spécifique n'a été que partiellement traité en l'analyse de forme suggérant que les deux communautés pourraient tirer profit l'une de l'autre.

3. Cette thèse vise à tirer parti de l'apprentissage automatique pour les séries temporelles et de l'analyse de formes pour proposer des méthodes adaptées aux recherches biomédicales nécessitant le traitement de données temporelles. Un intérêt particulier est aussi mis sur l'interprétation visuelle des formes et des déformations, une approche cruciale dans de nombreux cas.

Contributions :

1. Dans ce chapitre, un cadre général pour l'analyse des formes temporelles est proposé. Ce cadre constitue les fondations pour les chapitres suivants. En particulier, il définit l'objet séries temporelles, le groupe de déformations qui peuvent agir sur celles-ci ainsi que la manière dont ces déformations affectent les séries temporelles.

Contents

8.1 Motivation

L'analyse des données expérimentales pour valider ou réfuter une hypothèse est un principe fondamental de la science moderne. Cette approche est particulièrement centrale en recherche biomédicale, où elle permet notamment d'approfondir la compréhension des structures et des fonctions biologiques, de mettre au point de nouveaux traitements ou encore d'améliorer les diagnostics et les pratiques médicales. D'autre part, les récentes innovations technologiques ont grandement facilité l'acquisition non invasive de données biomédicales et parmi lesquelles les séries temporelles jouent un rôle important [GKK20 ; Fer17].

Par exemple, les électroencéphalogrammes (EEG), Figure 8.1a, enregistrent l'activité électrique du cerveau avec des électrodes placées autour du crâne. Parmi de nombreuses autres applications, les EEG jouent un rôle important dans le diagnostic de troubles neurologiques tels que l'épilepsie ou la narcolepsie ainsi que dans l'étude de fonctions cérébrales en réponse à divers stimuli externes [SBH74]. De manière similaire, l'électrocardiogramme (ECG), Figure 8.1b, mesure l'activité électrique du coeur à l'aide d'électrodes placées sur le corps. Ces séries temporelles facilitent le diagnostic de plusieurs pathologies cardiaques telles que l'arythmie ou l'évaluation de la réponse cardiaque à divers traitements cliniques [Vic+19]. En revanche, les signaux de marche, Figure 8.1c, mesurent la vitesse angulaire des pas à l'aide d'unités de mesure inertielle. Ces signaux offrent des informations précieuses pour la rééducation de patients à mobilité réduite en raison de pathologies telles que la maladie de Parkinson ou d'accidents vasculaires cérébraux [Bar+15].

La santé humaine étant en jeu, l'analyse des données biomédicales nécessite des outils mathématiquement et statistiquement fondés afin de garantir des interactions significatives entre les données et le personnel qualifié.

Des données structurées. De nombreuses séries temporelles biomédicales présentent des motifs déterministes qui reflètent l'état physiologique d'un sujet. Par exemple, des formes d'EEG spécifiques telles que les K-complexes (pics) et les spindles (motifs sinusoïdaux) sont représentatifs de la deuxième phase du sommeil, comme illustré Figure 8.2. De même, la forme des battements cardiaques enregistrés par ECG peut être modifiée par des conditions physiologiques comme les contractions ventriculaires prématurées (PVC),

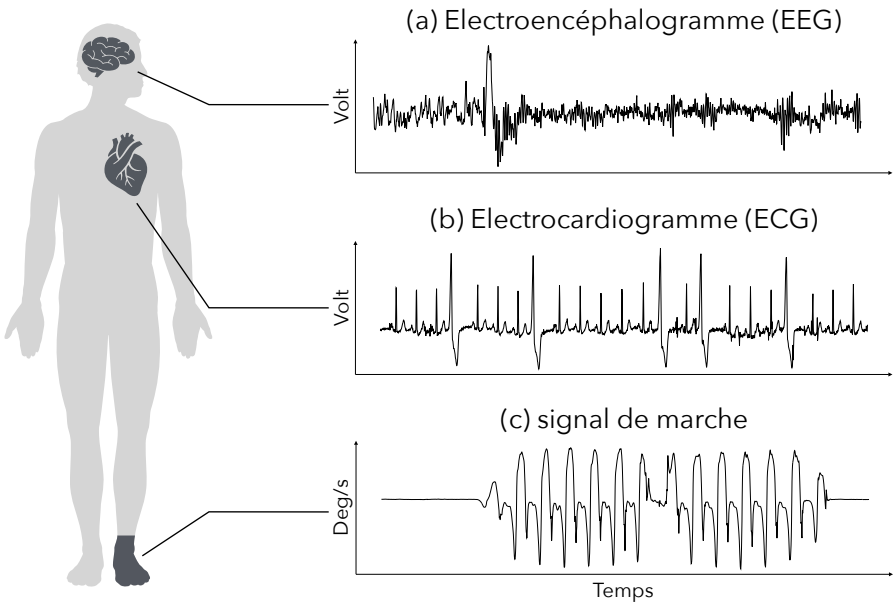


Figure 8.1 Illustrations de séries temporelles biomédicales non invasives. **(a)** Electroencéphalogramme (EEG) mesurant l'activité électrique du cerveau à l'aide d'électrodes, **(b)** Electrocardiogramme (ECG) mesurant l'activité électrique du cœur, et **(c)** Signal de marche mesurant la vitesse angulaire des pas à l'aide d'une unité de mesure inertielle.

Figure 8.2b. De même, les signaux de la marche varient également entre les individus sains et ceux atteints de troubles neurologiques, voir Figure 8.2c.

Ces motifs récurrents sont pertinents en recherche biomédicale puisqu'ils sont observés de manière cohérente chez différents sujets, ce qui en fait des variables robustes pour mener des analyses statistiques. Néanmoins, des outils mathématiques appropriés sont nécessaires pour établir des comparaisons précises de ces formes.

Caractéristiques de formes. Il est intéressant de noter que la comparaison de ces motifs déterministes se résume à la comparaison de leur forme. Historiquement, cela a été fait en comparant des caractéristiques prédéfinies et extraites des motifs, comme illustré dans le cas des battements de coeur par la Figure 8.3 (caractéristiques des battements de coeur). Toutefois, ces caractéristiques ont tendance à être localisées, ce qui peut entraîner une perte d'informations discriminantes. Plus récemment, des algorithmes d'apprentissage automatique et d'apprentissage profond ont été utilisés pour apprendre des caractéristiques directement à partir des données. Cependant, ces méthodes nécessitent souvent de grands ensembles de données, un luxe inabordable dans certains contextes biomédicaux. En outre, garantir la fiabilité et l'interprétabilité des caractéristiques apprises est un domaine de recherche actif et qui est essentiel en recherche biomédicale.

Alors que la première approche risque d'être trop réductionniste et que la seconde tend à être trop paramétrée, une troisième approche s'appuyant sur la notion de forme pourrait être étudiée pour remédier à ces deux limitations.

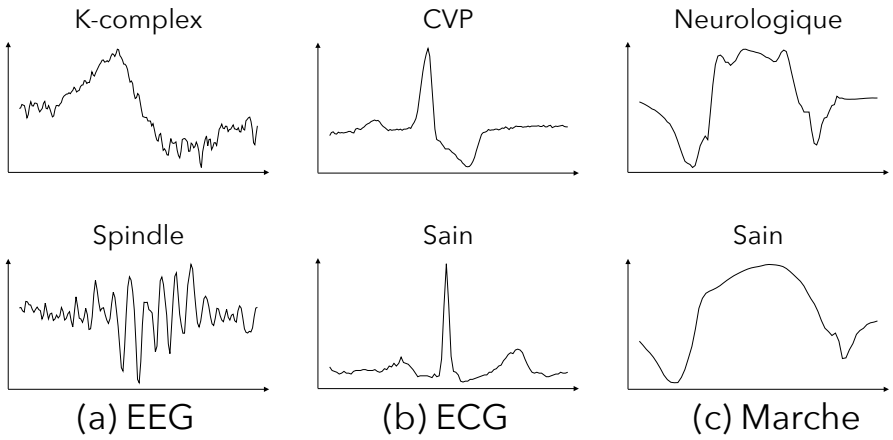


Figure 8.2 Illustrations de motifs déterministes. **(a)** Sur un EEG, le K-complexe et les spindles indiquent un sommeil en phase deux. **(b)** Sur un ECG, les battements de coeur des sujets souffrant de contraction ventriculaire prématurée (CVP) ont un profil différent de celui des sujets sains. **(c)** Dans un signal de marche, le pas d'un sujet atteint d'une pathologie neurologique diffère de celui d'un sujet sain.

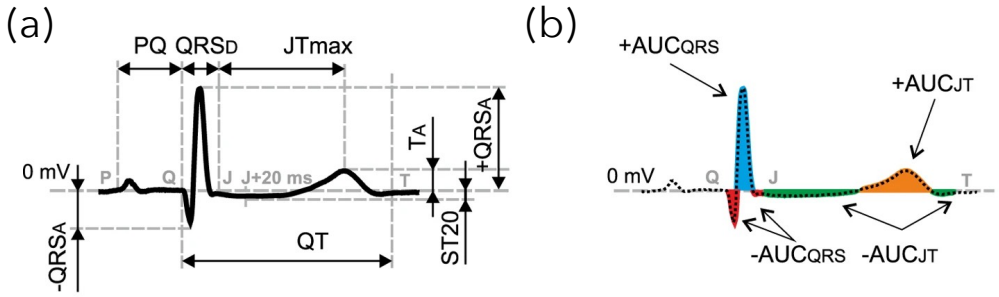


Figure 8.3 De [Mar+17]. Illustrations de caractéristiques standard pour décrire la forme du battement cardiaque à partir de l'ECG en vue de la classification automatique des battements cardiaques ventriculaires prématurés ou ischémiques. Il s'agit notamment de caractéristiques qui rendent compte de **(a)** la durée d'intervalles spécifiques et de l'amplitude de certains pics, ou de **(b)** l'aire sous la courbe sur des intervalles spécifiques.

Formes et séries temporelles. La communauté d'analyse des formes a défini un cadre mathématique pour l'étude des formes d'objets géométriques, dans lequel les concepts de forme et de déformation sont profondément liés. Par exemple, une feuille de papier peut être déformée en la pliant ou en la dépliant. La séparation entre objet (feuille de papier) et déformations (pliures) conduit à deux approches analytiques différentes :

1. **Comparer des objets indépendamment des déformations.** Par exemple, indépendamment du pliage, une feuille de papier plane et une feuille de papier pliée sont considérées comme le même objet.

2. **Comparer les objets en quantifiant les déformations.** Par exemple, un papier plié 4 fois et un papier plat diffèrent par 4 plis.

Par rapport aux approches précédentes, l'analyse de forme ajuste la complexité du problème en incorporant les connaissances d'experts dans la conception de l'ensemble de déformations. Les déformations sont soigneusement sélectionnées pour tenir compte des sources significatives de variabilité, en veillant à ce que les caractéristiques ou les déformations invariantes aient une signification biologique.

Alors que l'analyse de forme se concentre principalement sur l'imagerie médicale pour comparer des organes et des tissus soumis à des déformations spatiales, son application aux séries temporelles (données spatiotemporelles) reste relativement inexplorée. Malgré cela, les méthodes de séries temporelles s'appuyant sur la notion de forme ont démontré un succès significatif dans des tâches telles que la classification et le clustering. Ces méthodes s'appuient généralement sur des distances invariantes par rapport aux déformations courantes des séries temporelles, telles que le changement d'amplitude, le décalage et la reparamétrisation temporelle, comme illustré dans Figure 8.4.

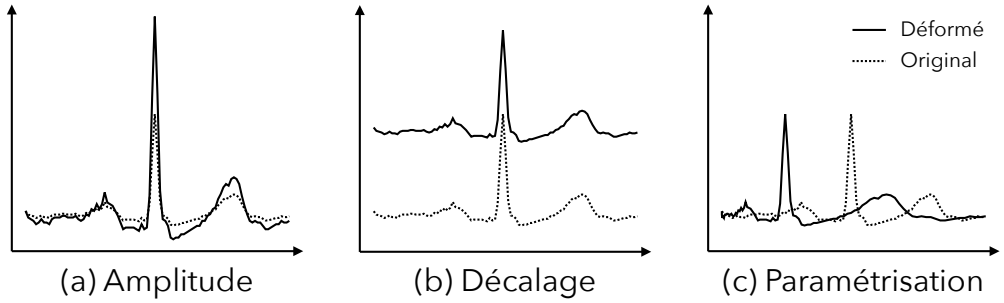


Figure 8.4 Illustration sur un battement de coeur provenant d'un ECG des déformations courantes des séries temporelles, y compris **(a)** le changement d'amplitude, **(b)** le décalage, et **(c)** la reparamétrisation temporelle.

Positionnement de la thèse. Cette thèse a été menée au Centre Borelli[1], un laboratoire de recherche multidisciplinaire qui rassemble des experts de divers domaines, y compris les mathématiques, l'informatique, les neurosciences, la biologie, la médecine et la pratique

[1] `https://centreborelli.ens-paris-saclay.fr/en`

clinique. Cette thèse vise à tirer parti de l'apprentissage automatique pour les séries temporelles et de l'analyse de formes pour proposer des méthodes adaptées aux recherches biomédicales nécessitant le traitement de données temporelles. Un intérêt particulier est aussi mis sur l'interprétation visuelle des formes et des déformations, une approche cruciale dans de nombreux cas.

Cas d'usage. Cette thèse est divisée en deux parties. La première partie se concentre sur la recherche de motifs spécifiques ou la découverte de motifs récurrents dans une longue série temporelle, indépendamment de certaines déformations prédéfinies. Conçues comme des méthodes pratiques, elles sont testées sur plusieurs données temporelles biomédicales, notamment les ECG et les EEG.

La deuxième partie se concentre sur les méthodes de représentation non supervisées de séries temporelles s'appuyant sur la notion de forme. Le développement de ces méthodes est motivé par un projet de recherche mené au Centre Borelli afin de mieux comprendre le rôle d'une enzyme dans la régulation de la respiration [Ner+19]. Une description détaillée de ce projet est fournie dans le Chapitre 4.

Située à l'intersection de l'apprentissage automatique pour les séries temporelles et de l'analyse de forme, la section suivante donne un aperçu des deux communautés de recherche et des fondements mathématiques nécessaires à l'application de l'analyse de forme aux séries temporelles.

8.2 A la croisée des chemins

8.2.1 Apprentissage automatique pour les time series

Les séries temporelles sont abondantes. Les séries temporelles apparaissent dans de nombreux domaines d'application et soulèvent divers défis. Parmi les nombreux exemples en dehors du biomédical, les astronomes s'intéressent à la classification de milliards d'objets astronomiques à partir de séries temporelles photométriques [JB20 ; Lin+12]. Les sismologues cherchent à prédire les tremblements de terre à venir à partir de sismogrammes en temps réel [BAM23]. Les économistes souhaitent détecter les manipulations frauduleuses du marché à partir de séries chronologiques de transactions financières [KG22 ; GZ15]. Les industriels souhaitent rationaliser leur chaîne d'approvisionnement en prévoyant les ventes [RLM21], en contrôlant leur niveau de stock [Avi03], ou en effectuant une maintenance préventive [RBP11].

Face à une telle diversité de contextes et de problèmes, la recherche en apprentissage automatique pour les séries temporelles s'est organisée autour de tâches transversales telles que la classification ou la prévision, ainsi que de critères d'évaluation des méthodes.

Des critères fondamentaux. Dans la littérature, les algorithmes sont généralement évalués sur la base de trois critères qui englobent les difficultés rencontrées dans la plupart des applications :

- **Efficacité :** Pour traiter des ensembles de séries temporelles potentiellement volumineux, les algorithmes doivent être efficaces à la fois en termes de temps de calcul et d'utilisation de la mémoire.

- **Interprétabilité :** Motivés par des applications dans des domaines tels que l'industrie ou la médecine, où les décisions algorithmiques peuvent affecter la santé et le bien-être des personnes, les algorithmes doivent présenter certaines garanties, notamment en terme d'interprétabilité afin d'expliquer la décision algorithmique à partir des données d'entrée.

- **Performances :** Pour motiver la création d'algorithmes performants dans de multiples domaines d'application, les chercheurs ont établi des mesures de performances spécifiques à chacune des tâches [SR24 ; JPJ24 ; Tat+18] et ils ont aussi proposé plusieurs jeux de données [Pap+22a ; God+21 ; Dau+19 ; Bag+18].

Les algorithmes proposés durant cette thèse seront évalués à la lumière de ces critères.

Des tâches transversales. Dans plusieurs applications, les mêmes tâches doivent être effectuées, et de nombreux chercheurs dans le domaine des séries temporelles ont axé leurs travaux autour de celles-ci [EA12b ; Fu11]. Dans ce qui suit, de brèves descriptions des tâches les plus courantes sont données :

- **Détection d'anomalies [SWP22] :** Détection de parties anormales dans une série temporelle. Le comportement normal/anormal peut être appris avec ou sans supervision.

- **Classification [Bag+17] :** Prédiction de la classe de séries temporelles suite à l'entraînement d'un algorithme à partir de sériés temporelles labélisées.

- **Clustering [ASW15] :** Regroupement de séries temporelles en ensembles homogènes en fonction d'une mesure de similarité et sans supervision.

- **Représentation [Li+17] :** Réduire la dimension des séries temporelles dans le temps ou dans l'espace pour gagner en performance et en efficacité sur les tâches en aval.

- **Prédiction [LZ21] :** Prédire l'avenir à partir d'observations passées en s'appuyant sur les propriétés statistiques du processus sous-jacent.

- **Découverte de motifs [TL17] :** Détection et localisation de motifs locaux qui se répètent dans une série temporelle.

- **Segmentation [TOV20] :** Division d'une série temporelle en segments homogènes à partir d'une mesure de similarité ou d'un entraînement.

- **Recherche de similarité [Pat+02] :** Recherche des occurrences de formes spécifiques au sein d'une unique série temporelle d'un ensemble de séries temporelles.

Dans cette thèse, des contributions ont été apportées à la recherche de similarités dans le Chapitre 2, à la découverte de motifs dans le Chapitre 3, et à la représentation de séries temporelles dans les Chapitres 5 et 6.

Deux échelles. La plupart des tâches liées aux séries temporelles s'effectuent sur à l'une de deux échelles : locale ou globale. Certaines tâches, comme le clustering, se concentrent sur l'échelle globale ; elles comparent des séries temporelles appartenant à un ensemble de données. D'autres, comme la découverte de motifs, se concentrent sur l'échelle locale ; elles recherchent des événements locaux dans une longue série temporelle unique. Dans certaines situations, la tâche fait référence aux deux échelles. Par exemple, la détection d'anomalies fait référence à la détection de séries temporelles anormales dans un ensemble de données mais aussi à la détection d'événements anormaux locaux dans une série temporelle. La Table 8.1 détaille l'échelle à laquelle chaque tâche opère.

En outre, il est possible de passer d'une tâche à l'échelle locale à une tâche à l'échelle globale avec un algorithme de segmentation approprié. Par exemple, avec un algorithme de segmentation des battements du coeur (Figure 8.2b), un ECG (Figure 8.1b) peut être décomposé en un ensemble de battements de coeur qui peuvent être comparés à l'aide de méthodes opérant à l'échelle globale.

Dans cette thèse, la Partie I se concentre sur les tâches à l'échelle locale, et la Partie II se concentre sur les tâches à l'échelle globale.

Table 8.1 Echelle opérationnelle des tâches courantes sur les séries temporelles.

Tâche	Locale	Globale
Détection d'anomalies	✓	✓
Classification		✓
Clustering		✓
Représentation		✓
Prédiction		✓
Découverte de motifs	✓	
Segmentation	✓	
Recherche de similarité	✓	✓

La jungle des distances. La plupart des algorithmes qui traitent les tâches mentionnées ci-dessus s'appuient sur des distances entre séries temporelles. Comme elles sont facilement interchangeables, de nombreuses distances ont été proposées pour améliorer les performances dans divers contextes. Face à la jungle des distances, plusieurs évaluations expérimentales ont été menées au fil des annéess [HMB24 ; Pap+20 ; AML19 ; Din+08]. Par exemple, une étude récente compare 71 distances sur 128 ensembles de données [Pap+20]. La majorité des distances se répartissent en deux familles :

- **Distances à pas fixe :** Elles comparent des séries temporelles de même longueur et supposent un appariement bijectif entre les échantillons. Elles sont connues pour leur efficacité en termes de calcul.

- **Distances élastiques :** Elles peuvent comparer des séries temporelles de longueurs différentes et, compte tenu de la distance entre les échantillons, elles trouvent l'appariement optimal entre échantillons qui minimise la distance globale. Elles sont connues pour leur robustesse face aux déformations de reparamétrisation temporelle.

Dans cette thèse, des contributions aux deux familles ont été faites et sont présentées dans le Chapitre 2 pour les distances à pas fixe et Chapitre 6 pour les distances élastiques.

Les distances s'appuyant sur la notion de forme se distinguent. Parmi toutes les distances, deux sont bien établies et considérées comme la distance de référence par de nombreux algorithmes : la distance euclidienne Z-normalisée [GK95] et la distance nommée "Dynamic Time Warping" (DTW) [SC78].Ces deux distances sont en fait des distances qui s'appuient sur la forme. Plus précisément, elles comparent des séries temporelles indépendamment de certaines déformations.

Appartenant à la famille des distances à pas fixe, la distance euclidienne Z-normalisée est invariante aux changements d'amplitude et aux décalages, voir Figure 8.4ab. Bien qu'élémentaires, ces déformations sont omniprésentes dans les séries temporelles et l'invariance devient cruciale dans de nombreuses applications. La distance euclidienne Z-normalisée $\mathbf{x} \in \mathbb{R}^n$ et $\mathbf{y} \in \mathbb{R}^n$ est définie par :

$$d_Z(\mathbf{x}, \mathbf{y}) = \left\| \frac{\mathbf{x} - \mu_x \mathbf{1}}{\sigma_x} - \frac{\mathbf{y} - \mu_y \mathbf{1}}{\sigma_y} \right\| , \tag{8.1}$$

avec $\mu_x = \frac{1}{n}\sum_{i=1}^n x_i$, $\sigma_x^2 = \frac{1}{n}\sum_{i=1}^n (x_i - \mu_x)^2$ et $\mathbf{1} = (1, \ldots, 1) \in \mathbb{R}^n$. Traitées comme des échantillons gaussiens, la moyenne et l'écart-type sont retirés des échantillons de sorte que la série temporelle devient invariante aux déformations de décalage et d'amplitude. La distance Z-normalisée bénéficie d'un calcul efficace [ZM24] et elle a connu un grand succès, en particulier dans la recherche de similarités et la découverte de motifs [ZM24 ; Yeh+16].

Appartenant à la famille des distances élastiques, la DTW est invariante à une source commune de variabilité interindividuelle : la paramétrisation temporelle de la série temporelle, voir Figure 1.4c. Dans sa forme originale [SC78], la DTW entre $\mathbf{x} \in \mathbb{R}^m$ et $\mathbf{y} \in \mathbb{R}^n$ est définie par :

$$dtw(\mathbf{x}, \mathbf{y}) = \min_{A \in \mathsf{A}_{m,n}} \langle A, \Delta \rangle_F, \quad \text{where :} \quad \Delta_{ij} = \|x_i - y_j\|^2 , \tag{8.2}$$

avec $\mathsf{A}_{m,n} \subset \{0,1\}^{m \times n}$ étant l'ensemble des matrices de chemins qui relient le coin supérieur gauche au coin inférieur droit [CB17].

De nombreuses variantes de la DTW ont été proposées au fil des ans, certaines visent notamment à améliorer sa robustesse au bruit [ZI18 ; CB17]. Récemment, la DTW a été combinée au transport optimal pour comparer des séries temporelles prenant valeur dans des espaces hétérogènes afin d'aborder des questions d'adaptation de domaine tout en garantissant l'invariance aux reparamétrisations temporelles [Pai+23 ; Coh+21 ; JCG20]. De même, un travail récent [Vay+20] a proposé une distance basée sur DTW également invariante aux déformations globales appartenant aux variétés de Stiefel. Par exemple, cette distance est bien adaptée à la comparaison de séries temporelles d'enregistrements

de mouvements où l'angle de la caméra peut différer d'un enregistrement à l'autre. Il convient de noter qu'il ne s'agit pas d'une métrique, car elle ne garantit pas l'inégalité triangulaire, et que son temps de calcul est quadratique. Cependant, les distances fondées sur la DTW sont performantes dans de nombreuses tâches sur des ensembles de séries temporelles courtes [Wan+13].

Les distances s'appuyant sur la notion de forme ont connu un grand succès et constituent un premier choix pour de nombreuses applications. Cependant, la notion de forme dans les séries temporelles n'a pas été entièrement explorée et des améliorations sont encore possibles en s'inspirant de la littérature en analyse de formes.

8.2.2 Analyse de formes

Comparer des formes. L'analyse de forme fait référence aux méthodes qui comparent des objets géométriques tels que des surfaces ou des courbes, en accordant une attention particulière à la modélisation de la variabilité inter-objets. Comme l'illustre Figure 8.5, les premières applications ont été faites en biologie [DM16], où les chercheurs s'intéressaient aux différences anatomiques entre les espèces indépendamment d'une source de variabilité modélisée par des dilatations, des translations ou des rotations. Une telle analyse est connue sous le nom d'analyse de Procuste ordinaire [HC62].

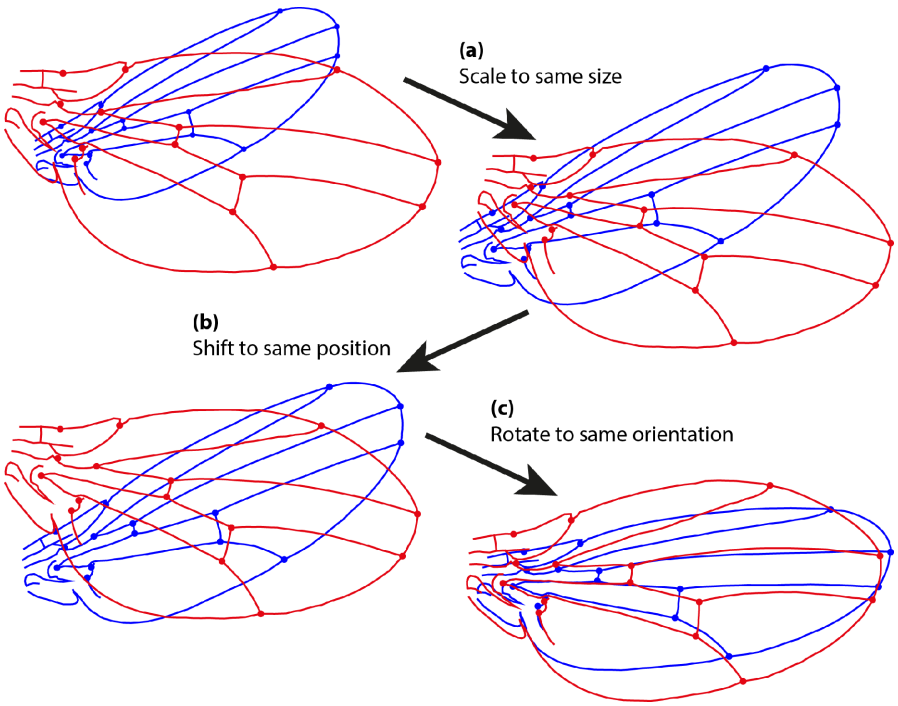


Figure 8.5 A partir de [Kli15], les formes des ailes d'insectes sont comparées grâce à l'analyse de Procrustes ordinaire. **(a)** les deux ailes sont mises à la même échelle, ce qui élimine la variabilité de dilatation, **(b)** le barycentre des deux ailes est translaté vers l'origine, ce qui élimine la variabilité de translation, et **(c)** l'aile bleue est orientée similairement à l'aile rouge, ce qui élimine la variabilité de rotation. Enfin, la distance entre les ailes est la somme des distances euclidiennes entre les points de repère.

Vers des méthodes statistiques. Plus récemment, l'analyse de forme a été appliquée dans des domaines tels que la vision par ordinateur [Wei18 ; WM18 ; You12], l'imagerie médicale [Sto+24 ; Dub+18 ; Mor+08], ou l'anatomie computationnelle [Gas+22 ; MTY02 ; GM98], où les méthodes statistiques jouent un rôle central dans le processus scientifique. Par exemple, plusieurs études ont porté sur la relation entre la forme de l'hippocampe et la maladie d'Alzheimer [Wen+20 ; Chu+09 ; Wan+07], et d'autres sur la relation entre la forme du cœur et certains dysfonctionnements [Gua+24 ; Man+11 ; Hel+05].

Malheureusement, les méthodes statistiques classiques ne sont pas adaptées aux espaces de forme, car ceux-ci ne sont généralement pas dotés d'une structure vectorielle. Par exemple, la somme pixellisée de deux IRM cérébrales ne donne pas une IRM cérébrale. Le développement de méthodes statistiques dédiées aux espaces de forme est devenu un sujet de recherche actif au cours des deux dernières décennies [Fey20].

Métrique sur un espace de formes. S'il est difficile de définir une structure vectorielle appropriée sur un espace de formes, il est plus facile de quantifier la différence entre les formes. Un grand nombre de travaux se sont concentrés sur la définition d'une structure métrique sur les espaces de forme qui sont évalués autour de trois critères :

- **Pertinence pour le domaine d'application :** englobe les sources de variabilité en fonction de leur effet sur les formes.
- **Fondement mathématique :** Hérite de propriétés mathématiques pertinentes pour les méthodes en aval, notamment les méthodes statistiques.
- **Efficacité informatique :** Adaptable à de grands ensembles de données.

Déformation et action de groupe. Une approche conceptuelle pour définir une métrique sur l'espace des formes a été introduite [Gre94]. Plus précisément, les sources de variabilité sont modélisées comme des déformations de l'espace ambiant auquel appartiennent les objets géométriques. L'ensemble des déformations est doté d'une structure de groupe et son action sur les objets géométriques est décrite par une action de groupe.

Définition 1 (Action de groupe)**.** *Un groupe* G *de neutre e agit part la gauche sur l'ensemble* M*, s'il existe une fonction* $a : \mathsf{G} \times \mathsf{M} \mapsto \mathsf{M}$ *qui vérifie :*

$$\begin{aligned} &1)\ a(e, m) = m, \quad \forall m \in \mathsf{M} \\ &2)\ a(g, a(h, m)) = a(gh, m), \quad \forall (g, h) \in \mathsf{G}^2, \forall m \in \mathsf{M}. \end{aligned}$$

Remarque 1. *L'action à droite peut également être définie ; il suffit de remplacer la deuxième propriété par* $a(g, a(h, m)) = a(hg, m)$*. Pour simplifier les notations, les actions gauches (resp. droites) sont notées* $g \times m \mapsto g \cdot m$ *(resp.* $g \times m \mapsto m \cdot g$*).*

Pour un groupe G qui agit à gauche sur un ensemble M, l'orbite d'un élément $m \in \mathsf{M}$ est l'ensemble $[m] = \{g \cdot m \mid g \in \mathsf{G}\}$. L'action de G sur M est dite transitive si pour tout $m \in \mathsf{M}$ son orbite est l'ensemble entier : $[m] = \mathsf{M}$. Différentes stratégies de définition de métriques doivent être envisagées selon que la propriété de transitivité s'applique ou non.

Action non transitive. Pour les actions non transitives, les métriques sont conçues pour comparer les formes indépendamment de l'ensemble des déformations. Formellement, l'ensemble des orbites indépendantes, noté M/G, appelé espace quotient, n'est pas réduit à un singleton. Chaque orbite représente une forme, et l'espace quotient M/G doit être doté d'une structure métrique.

Théorème 1. *Soit (M, d) un espace métrique et G un groupe qui agit non transitivement sur la gauche sur M. La fonction $\tilde{d}$ définie par :*

$$\tilde{d}([m],[m']) = \inf_{(g,g')\in\mathsf{G}^2} d(g\cdot m, g'\cdot m')$$

est une métrique sur M/G, si les orbits sont des fermés de M pur la topologie induite par d.

De plus, si d est G-équivariante, ie $d(g\cdot m, g\cdot m') = d(m,m')$, $\tilde{d}$ vérifie aussi :

$$\tilde{d}([m],[m']) = \inf_{g\in\mathsf{G}} d(m, g\cdot m')$$

Démonstration. Voir chapitre 12 du livre *Shapes and diffeomorphisms*, [You10]. □

Exemple 1 (Invariance aux rotations et translations)**.** *Une distance invariante aux rotations et translations est une application directe du Théorème 1.*

Formellement, supposons deux ensembles de repères appariés $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ et $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$ vivant dans l'espace ambiant $\mathbb{R}^d$. Les objets $\mathbf{x}$ et $\mathbf{y}$ appartiennent à la même orbite s'il existe une rotation $R \in \mathsf{SO}(d)$ et une translation $\tau \in \mathbb{R}^d$ telle que :

$$\mathbf{y} = R\mathbf{x} + \tau \qquad i.e. \qquad \forall i \in [\![1, N]\!],\ \mathbf{y}_i = R\mathbf{x}_i + \tau. \tag{8.3}$$

Si $\mathbf{x}$ et $\mathbf{y}$ ne sont pas ramenés à un seul point, par équivariance de translation et de rotation de la distance euclidienne, la distance invariante est définie comme suit :

$$d_{RT}(\mathbf{x},\mathbf{y}) = \inf_{(R_1,\tau_1,R_2,\tau_2)} \|(R_1\mathbf{x}+\tau_1) - (R_2\mathbf{y}+\tau_2)\| = \inf_{(R,\tau)} \|(R\mathbf{x}+\tau) - \mathbf{y}\| \tag{8.4}$$

Exemple 2 (Invariance aux paramétrisations temporelles : le cadre Square Root Velocity (SRV))**.** *Issu de l'analyse des formes, le cadre Square Root Velocity [Sri+10] vise à comparer les courbes indépendamment de leur paramétrage temporel. Il propose une distance invariante à la paramétrisation temporelle construite à travers la stratégie du Théorème 1.*

Formellement, prenons $\mathsf{M} \subset \mathsf{L}^2([0,1],\mathbb{R}^d)$ comme l'ensemble des courbes ouvertes intégrables qui sont différentiables, avec une dérivée première également intégrable, et telles que pour tout $c \in \mathsf{M}$, $c(0) = 0$. Le but est de définir une distance entre les courbes qui soit invariante par rapport à l'action du groupe $\mathsf{G} = \{\gamma \in C^1([0,1],[0,1]) \mid \gamma(0) = 0,\ \gamma(1) = 1,\ \gamma'(t) > 0\ \forall t\}$.

Pour ce faire, considérons la fonction de représentation bijective F telle que pour toute courbe $c \in \mathsf{M}$, $F(c)$ est la courbe définie comme :

$$F(c) : t \mapsto \begin{cases} c'(t)/\sqrt{\|c'(t)\|},\ if\ c'(t) \neq 0 \\ 0,\ else \end{cases}, \tag{8.5}$$

et la distance d sur M :

$$d:\ (c_1, c_2) \in \mathsf{M} \times \mathsf{M} \mapsto \int_0^1 \|F(c_1)(t) - F(c_2)(t)\|^2 dt\ , \tag{8.6}$$

la distance d est G-équivariante, ce qui signifie que pour toutes courbes c_1, c_2 and paramétrisation temporelle γ, $d(c_1 \circ \gamma, c_2 \circ \gamma) = d(c_1, c_2)$. D'après le Théorème 1, la fonction :

$$\tilde{d}:\ ([c_1], [c_2]) \in \mathsf{M}/\mathsf{G} \times \mathsf{M}/\mathsf{G} \mapsto \inf_{\gamma \in \mathsf{G}} \int_0^1 \|F(c_1 \circ \gamma)(t) - F(c_2)(t)\|^2 dt\ , \tag{8.7}$$

est une pseudo-distance qui compare les courbes jusqu'à leur paramétrage temporel, et avec quelques considérations techniques [Sri+10], elle définit une distance sur M/G.

Action transitive. Avec une action transitive, il est toujours possible de trouver une déformation qui fait correspondre un objet géométrique à un autre. La déformation déforme l'espace ambiant du premier objet pour le faire correspondre au second. L'intérêt de l'action transitive réside dans la possibilité de décrire la transformation d'un objet en un autre à une échelle globale et locale et pour tout point de l'espace ambiant. Malheureusement, la stratégie décrite dans le cas non transitif n'est pas transférable au cas présent. Cependant, la définition d'une distance sur l'espace des formes M est toujours possible si le groupe G peut être doté d'une structure métrique. Intuitivement, les distances définies par le théorème suivant quantifient « combien » l'objet source doit être déformé pour être mis en correspondance avec l'objet cible.

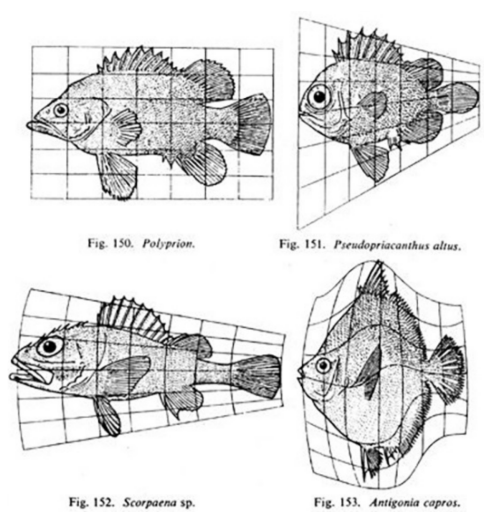
Théorème 2. *Soit (G, e) un groupe munit d'un métrique d_{G} et qui agit transitivement à gauche sur l'ensemble M. Si d_{G} est une métrique droite-équivariante sur G, ie $d_{\mathsf{G}}(gh, g'h) = d_{\mathsf{G}}(g, g')$, alors $\tilde{d}$ définie par :*

$$\tilde{d}(m, m') = \inf_{g \in \mathsf{G}} \{d_{\mathsf{G}}(e, g) \mid g \cdot m = m'\}$$

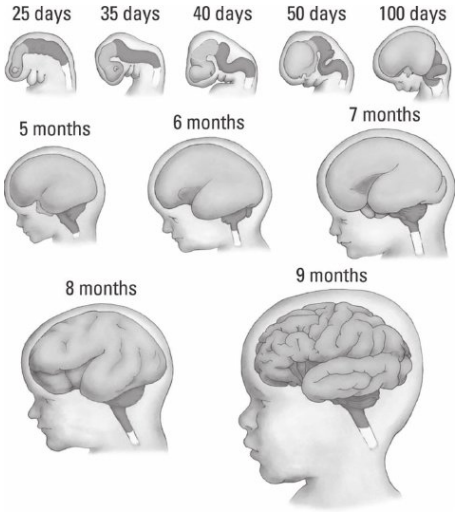
est une métrique sur M si $\{g \in \mathsf{G} \mid g \cdot m_0 = m_0\}$ est fermé pour la topology induite par d_{G} et pour un élément fixé $m_0 \in \mathsf{M}$.

Démonstration. Voir chapitre 12 du livre *Shapes and diffeomorphisms*, [You10]. □

Enraciné dans les travaux du biomathématicien D'Arcy Thompson [Tho17] qui a décrit pour la première fois le passage d'une espèce à une autre par le biais d'une déformation géométrique, voir Figure 8.6a, le groupe des difféomorphismes a fait l'objet d'une attention particulière dans l'analyse des formes pour les actions transitives. Intuitivement, les difféomorphismes sont des applications bijectives qui, elle-même et son inverse, sont différentiables et de différentielles continue. Ces déformations peuvent être générées par des équations différentielles ordinaires, ce qui rend les distances induites par ce groupe pertinentes pour toute application biomédicale dans laquelle la déformation d'une forme évolue de manière régulière dans le temps. Par exemple, le cerveau d'un enfant se forme progressivement pendant la grossesse, voir Figure 8.6b, et l'évolution de la forme du cerveau peut être comparée au niveau d'une population dans l'objectif de fournir des informations précieuses aux cliniciens et des conseils aux parents [GBA21].



(a) De [Tho17], correspondance par déformation entre poissons.



(b) De [KF09], développement du cerveau de l'enfant durant la grossesse.

Figure 8.6 (a) Le poisson dans le coin supérieur gauche est déformé pour représenter d'autres poissons. L'hypothèse sous-jacente de D'Arcy Thompson est que les déformations entre espèces étroitement apparentées doivent être « faibles ». **(b)** Description schématique du développement du cerveau d'un enfant pendant la grossesse. L'absence de développement de certaines parties du cerveau de l'enfant pendant la grossesse peut entraîner des dysfonctionnements cognitifs. Il est essentiel pour les cliniciens de détecter ces anomalies et de les différencier d'éventuels retards de développement [GBA21].

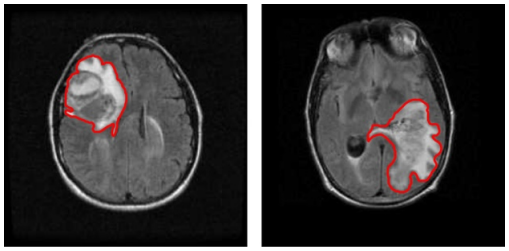
Les séries temporelles ne sont pas des courbes. Les séries temporelles et les courbes réfèrent au même objet mathématique : une application d'un intervalle fermé $\mathsf{I} \subset \mathbb{R}$ prenant une valeur dans un espace E. Cependant, leur différence vient des défis soulevés par les divers champs d'application qui sont traités par deux communautés différentes.

Les courbes sont souvent issues d'applications en vision par ordinateur ou en imagerie médicale où elles font fréquemment référence à des objets détourés, voir Figure 8.7a. Ce type de courbes a été largement étudié dans l'analyse des formes [Bau+21 ; You10]. Ici, l'aspect temporel des courbes se réfère simplement à la paramétrisation de l'objet détouré et n'apporte aucune information significative. Par conséquent, toutes les courbes sont définies sur un même intervalle fermé I, et l'accent est mis sur la comparaison de la forme des courbes indépendamment de toute paramétrisation temporelle.

Pour opposer les séries temporelles aux courbes, prenons un exemple pratique. La bradycardie est une maladie dans laquelle les sujets ont un rythme cardiaque anormalement bas, ce qui entraîne un manque d'oxygène. La différence entre les sujets sains et les sujets malades peut être observée sur les électrocardiogrammes (ECG), voir Figure 8.7b. Par rapport aux sujets sains, le cycle cardiaque des sujets souffrant de bradycardie présente une longue pause à la fin de la contraction du coeur. Il est surprenant de constater que les cycles des patients sains et malades sont identiques lorsqu'ils sont comparés indépendamment de toute paramétrisation temporelle. L'information discriminante réside dans la paramétrisation temporelle du cycle cardiaque, qui devrait être incluse dans la

notion de forme des séries temporelles.

L'exemple précédent montre qu'une application directe des méthodes conçues pour les courbes aux séries temporelles est restrictive dans certaines situations. D'autre part, l'important et fructueux corpus de travaux autour de la notion de forme doit être ajouté à la communauté des séries temporelles. Cette remarque motive le positionnement de cette thèse pour étendre une certaine notion d'analyse de forme au contexte des séries temporelles.



(a) De [Bha+18], IRM du cerveau.

(b) ECG du site internet de Cleveland clinic.

Figure 8.7 (a) Courbes représentant des tumeurs cérébrales détourées à partir d'IRMs. **(b)** Différence d'électrocardiogramme (ECG) entre un sujet sain et un sujet souffrant de bradycardie, une maladie où le coeur a une vitesse de contraction lente. Alors que les contractions du coeur sont identiques, les cycles des battements cardiaques présentent de longues pauses après la contraction en cas de bradycardie. À l'échelle des cycles cardiaques individuels, les méthodes de comparaison des courbes indépendamment de la paramétrisation temporelle ne permettront pas de différencier les sujets sains des sujets souffrant de bradycardie. Extrait du site internet : `https://my.clevelandclinic.org/health/diseases/17841-bradycardia`.

8.2.3 Un cadre général pour l'analyse des formes des séries temporelles

D'après la section précédente, trois choses doivent être définies pour établir une notion de forme : un ensemble d'objets géométriques, un groupe de déformations et l'action du groupe sur l'ensemble. Les paragraphes suivants présentent, de la manière la plus générique possible, ces ensembles et l'action de groupes dans le contexte des séries temporelles. La définition de métriques s'appuyant sur la notion de forme sera au centre des prochains chapitres où le cadre générique sera décliné à des cas plus spécifiques.

Représentation de séries temporelles. Dans la littérature [Bau+21 ; Wil17], les séries temporelles sont généralement représentées de deux manières :

- **La représentation fonctionnelle :** Une série temporelle est une fonction f d'un interval fermé $\mathsf{I} \subset \mathbb{R}$ à valeur dans $\mathbb{R}^d$.
- **La représentation discrète :** Une série temporelle est une séquence $(f(t_1), \ldots, f(t_n)) \in \mathbb{R}^{n \times d}$ échantillonnée aux temps $t_1 < \ldots < t_n \in \mathsf{I}$.

Alors que la représentation fonctionnelle d'une série temporelle est exacte, sa contrepartie discrète est une approximation dont l'erreur dépend de l'échantillonnage. En ce qui

concerne l'analyse de forme, la représentation fonctionnelle des séries temporelles est plus attrayante pour le mathématicien car elle est exacte et évite les problèmes liés à l'échantillonnage. Cependant, dans la pratique, nous n'avons accès qu'à des représentations de séries temporelles discrètes. Cela motive le besoin de passerelles entre les représentations fonctionnelles et discrètes afin de combler le fossé entre la théorie et les applications.

Cette thèse s'intéressera principalement à la représentation fonctionnelle pour définir des distances basées sur la forme entre les séries temporelles. En outre, des efforts seront faits pour décliner ces distances au cas discret et pour fournir des garanties de convergence vers la distance exacte au fur et à mesure que la discrétisation s'affine.

Ensembles de séries temporelles admissibles. Par rapport à l'ensemble des courbes qui correspondent à toutes les fonctions continues sur le même intervalle fermé $\mathsf{I} \subset \mathbb{R}$ à valeur dans $\mathbb{R}^d$, l'ensemble admissible des séries temporelles diffère de deux façons. L'hypothèse de continuité doit être révoquée car elle n'est pas valable dans plusieurs domaines d'application. Par exemple, la consommation électrique des appareils se comporte souvent comme un signal binaire.

Plus important encore, la restriction consistant à définir les fonctions sur le même intervalle fermé I doit également être supprimée. En effet, pour revenir à l'exemple de la bradycardie, lorsque l'on compare des cycles cardiaques, l'information discriminante réside dans la paramétrisation temporelle et en particulier dans la longueur de l'intervalle sur lequel la fonction est définie.

L'ensemble des séries temporelles admissibles doit englober ces différences et peut être défini au sens le plus général comme l'union suivante :

$$\mathsf{F} = \{(\mathsf{I}, f) \mid \mathsf{I} \in \mathcal{I} \text{ and } f \in \mathsf{M}(\mathsf{I}, \mathbb{R}^d)\} \ , \tag{8.8}$$

où $\mathcal{I}$ est l'ensemble des intervalles fermés de $\mathbb{R}$ et $\mathsf{M}(\mathsf{I}, \mathbb{R}^d)$ est l'ensemble des fonctions Borel mesurables de I dans $\mathbb{R}^d$. On notera que ce vaste ensemble englobe la plupart des séries temporelles rencontrées dans les applications. Cependant, cet ensemble est peu structuré et les prochains chapitres se concentreront sur des sous-ensembles qui présentent plus de structure afin de faciliter la définition des métriques.

Action de groupe admissible pour les séries temporelles. Les actions de groupe ont été introduites en analyse des formes pour modéliser l'action d'une déformation sur un objet géométrique. Si l'on considère une série temporelle $f : \mathsf{I} \mapsto \mathbb{R}^d$, une déformation qui aurait un sens est une combinaison d'une distorsion $h : \mathsf{I} \mapsto \mathbb{R}^d$ et d'une paramétrisation temporelle $\gamma : \mathsf{I} \mapsto \mathsf{J}$ qui conduirait à la série temporelle déformée : $g = (f + h) \circ \gamma^{-1}$.

Pour correctement définir une action de groupe sur l'ensemble admissible des séries temporelles F, modélisons les distorsions par l'ensemble des fonctions Borel mesurables $\mathsf{M}(\mathbb{R}, \mathbb{R}^d)$ et la paramétrisation temporelle par l'ensemble des homéomorphismes strictement croissants $\mathsf{H}^+(\mathbb{R})$. L'ensemble $\mathsf{M}(\mathbb{R}, \mathbb{R}^d) \rtimes \mathsf{H}^+(\mathbb{R})$ avec la règle de composition : $(h_2, \gamma_2) \rtimes (h_1, \gamma_1) = (h_1 + h_2 \circ \gamma_1, \gamma_2 \circ \gamma_1)$ forme un groupe qui peut agir sur F par l'action à gauche :

$$(h, \gamma) \cdot (\mathsf{I}, f) = \left(\gamma(\mathsf{I}), (f + h) \circ \gamma^{-1}\right) \ . \tag{8.9}$$

Notons que cette action est transitive, ce qui signifie que pour toute série temporelle admissible (I, f) et (J, g), il existe une déformation (h, γ) telle que $(h, \gamma) \cdot (\mathsf{I}, f) = (\mathsf{J}, g)$. Plus encore, il existe une multitude de déformations qui font correspondre (I, f) à (J, g) car pour toute paramétrisation temporelle γ, la distorsion dont la restriction sur I est égale à $g \circ \gamma - f$ assure la correspondance. En résumé, cette action sur les séries temporelles est très expressive et offre de nombreuses façons de modéliser des déformations pertinentes pour les cas d'applications.

En termes de notations, le groupe $\mathsf{M}(\mathbb{R}, \mathbb{R}^d)$ seul réfère aux **déformations rigides**, tandis que les groupes $\mathsf{H}^+(\mathbb{R})$ et $\mathsf{M}(\mathbb{R}, \mathbb{R}^d) \rtimes \mathsf{H}^+(\mathbb{R})$ réfèrent aux **déformations élastiques**.

Pour revenir à la littérature en analyse des formes, l'action du groupe admissible relève de la notion des formes fonctionnelles [CCT17 ; CT14], un problème émergent en anatomie computationnelle [MQ09].

Simplification par cas d'usage. La simplification de l'action de groupe admissible se fait au prix d'une restriction de l'ensemble des séries temporelles et des déformations pour un gain de structure supplémentaire ce qui permet de définir des métriques sur des espaces de formes. En fonction de l'application, les simplifications peuvent être effectuées selon deux stratégies :

- **Invariance à un ensemble de déformations :** L'objectif est de comparer des séries temporelles indépendamment de certaines déformations telles que l'amplitude, le décalage, la paramétrisation temporelle, etc. La simplification de l'action de groupe conduit à une action non transitive qui, selon le théorème 1, conduit à une métrique invariante par rapport aux déformations, à condition que l'ensemble des séries temporelles puisse être doté d'une métrique équivariante par rapport au groupe de déformations. De telles métriques seront étudiées dans le Chapitre 2. Il sera notamment démontré que ce cadre inclut la distance euclidienne Z-normalisée (Equation 8.1) ainsi que d'autres distances intéressantes.

- **Quantification de déformations significatives :** L'objectif est de comparer des séries temporelles par la déformation qui fait correspondre une série temporelle à une autre. Le groupe de déformations est choisi pour donner un sens à l'application, et l'action de groupe qui en résulte satisfait la propriété de transitivité. Selon le théorème 2, une métrique liée à la déformation entre les séries temporelles peut être établie si le groupe de déformations peut être doté d'une métrique équivariante à lui-même. Cette stratégie sera explorée dans le Chapitre 6 en ajustant des méthodes bien établies de l'analyse de formes au cas des séries temporelles.

Conclusion. Les paragraphes précédents présentent un cadre général pour l'analyse de formes appliquée au cas des séries temporelles, notamment en tirant parti d'une action de groupe suffisamment expressive. En ce qui concerne son application à différents cas d'usage, deux stratégies ont été présentées pour créer des métriques fondées sur la notion de forme en séries temporelles. Ce cadre constitue les fondations sur lesquelles cette thèse est menée.

8.3 Déroulé de la thèse

La thèse est organisée comme suit :

La **première partie** se concentre sur la recherche de formes spécifiques ou la découverte de motifs récurrents dans une longue série temporelle, indépendamment de certaines déformations prédéfinies. Elle traite spécifiquement des tâches à échelle locale sur une unique série temporelle, comprenant la recherche de similarités et la découverte de motifs. De même, elle aborde l'invariance par rapport à des groupes prédéfinis de déformations rigides.

- **Chapitre 2** aborde le problème de la recherche d'occurrences (répétitions) de formes prédéfinies dans une longue série temporelle. La première section passe en revue les travaux relatifs à la recherche de similarités dans des séries temporelles, en mettant particulièrement l'accent sur les méthodes exactes qui utilisent des distances à pas fixé. Les propriétés algorithmiques contribuant à leur efficacité sont détaillées. En s'appuyant sur ces propriétés, la deuxième section introduit un cadre général pour la construction de distances invariantes par rapport à des ensembles de déformations définis par l'utilisateur, tout en garantissant une complexité de temps de calcul équivalente à celle des méthodes les plus efficaces. La dernière section applique ce cadre pour développer une distance invariante au changement d'amplitude, au décalage et à la tendance linéaire. Cette distance s'avère précieuse dans les cas où les séries temporelles sont affectées par des déformations induites par une tendance, comme démontré expérimentalement.
- **Chapitre 3** présente un nouvel algorithme pour la découverte de motifs. Après une revue complète de la littérature en découverte de motifs, la deuxième section présente l'algorithme proposé, appelé PEPA, qui permet de découvrir des motifs de longueur variable. Cet algorithme transforme une série temporelle en un graphe et utilise l'homologie persistante pour résumer le graphe en un diagramme. Les motifs sont ensuite identifiés par une interprétation visuelle du diagramme. Bien que PEPA demande à l'utilisateur de spécifier le nombre de motifs à découvrir, une version adaptative, A-PEPA, utilisant une simple heuristique pour déduire ce nombre est également introduite. La section suivante évalue les performances des algorithmes sur des bases de données labéllisées agrégées au cours de cette thèse et des études de sensibilités sont menées. Une application web est présentée dans la dernière section, elle démontre comment l'efficacité de l'algorithme et l'interprétation visuelle du diagramme peuvent être combinées pour permettre une découverte interactive de motifs.

La **seconde partie** se concentre sur les méthodes de représentation non supervisées de séries temporelles s'appuyant sur la notion de forme. Plus précisément, cette tâche, qui agit à l'échelle globale, est abordée en considérant des groupes de déformations élastiques pouvant être restreints ou bien quantifiés.

- **Chapitre 4** présente l'application biomédicale qui a motivé le développement des méthodes proposées dans les chapitres suivants. En bref, une enzyme joue un

rôle important dans la régulation de l'activité musculaire et de la transmission des signaux au sein du système nerveux. Certains médicaments inhibent l'action de cette enzyme pouvant engendrer des conséquences graves, notamment sur la respiration, et qui ne sont pas encore totalement comprises. Pour étudier les conséquences de l'inhibition, la respiration de souris de différents génotypes est mesurée par pléthysmographie une fois les souris exposées à des inhibiteurs. La première section présente les outils de mesure (pléthysmogramme) et souligne les limites des méthodes existantes pour l'analyse de ces signaux. En outre, un algorithme permettant de segmenter les signaux de pléthysmographie en ensembles de données de cycles respiratoires (inspiration et expiration) est présenté. La deuxième section décrit le contexte biologique et le protocole expérimental.

- **Chapitre 5** présente, dans la première section, une nouvelle méthode non supervisée de référence pour l'analyse des signaux de pléthysmographie. Cette méthode utilise un algorithme de clustering combiné à la distance DTW pour apprendre une représentation symbolique des cycles respiratoires. La symbolisation des signaux de pléthysmographie se traduit par des séquences de symboles faisant référence à des formes caractéristiques. Les résultats et la discussion qui suivent illustrent en particulier l'interprétabilité de la méthode en présentant des correspondances entre les symboles et les fonctions physiologiques. Parmi plusieurs découvertes, les représentations symboliques ont mis en évidence des modalités respiratoires dépendantes du génotype et une réponse physiologique hétérogène suite à l'exposition aux inhibiteurs.

- **Chapitre 6** présente une méthode, appelée TS-LDDMM, qui représente une série temporelle par le vecteur paramétrant la déformation qui met en correspondance une série temporelle de référence avec la série temporelle observée. Cette méthode s'appuie sur le cadre "Large Deformation Diffeomorphic Metric Mapping" (LDDMM) qui est issu de l'analyse de formes et présenté dans la première section. LDDMM apprend des déformations difféomorphiques en résolvant des équations différentielles spécifiques. La deuxième section adapte le cadre LDDMM aux séries temporelles en établissant des conditions suffisantes sur le système différentiel pour garantir que les déformations apprises préservent la structure spatio-temporelle des séries temporelles. Par souci de concision, les études de performances et de sensibilités de l'algorithme proposées sont incluses en annexe. Dans ce chapitre, la section expérimentale se concentre sur l'étude de la ventilation chez la souris. Cette section démontre comment les représentations TS-LDDMM capturent des déformations physiologiquement significatives dont l'interprétabilité est accrue en analysant les résultats statistiques associés. Plus précisément, les représentations TS-LDDMM ont permis de caractériser les génotypes des souris, les modalités de ventilation et les effets de l'exposition aux inhibiteurs.

8.4 Contributions

Chapitre 2 :

1. Un cadre général est présenté pour construire des distances invariantes par rapport à des déformations rigides spécifiques et qui peuvent être intégrées dans les algorithmes de recherche de similarité les plus récents sans en compromettre l'efficacité. Plus précisément, lorsque les sources de variabilité peuvent être modélisées comme un groupe de déformations agissant sur les séries temporelles en tant que sous-espace vectoriel, il est possible de créer une représentation des séries temporelles invariante par rapport à la déformation, où la distance entre les représentants est simplement la distance euclidienne. Ce cadre étend la célèbre distance euclidienne Z-normalisée.

2. Pour illustrer cette extension, la distance euclidienne LT-normalisée, invariante au changement d'amplitude, au décalage et à la tendance linéaire, est présentée. Cette distance est localement robuste aux déformations causées par une tendance et elle a fait ses preuves dans plusieurs cas d'utilisation biomédicale.

Chapitre 3 :

1. Ce chapitre présente un algorithme appelé PersistentPattern (PEPA) qui permet de découvrir des motifs de longueur variable sans avoir besoin d'une connaissance préalable de la similarité entre les occurrences des motifs. PEPA fonctionne en transformant une série temporelle en un graphe et en la représentant à l'aide de l'homologie persistante, un outil d'analyse topologique des données. Les motifs pertinents sont ensuite identifiés à partir du résumé du graphe.

2. Une version adaptative de l'algorithme qui déduit le nombre de motifs à découvrir à partir du résumé du graphe est également présentée.

3. Un benchmark de 9 jeux de données labélisées, dont 6 jeux de données issus de cas réels, est introduit pour la découverte de motifs. Les évaluations empiriques montrent que PEPA surpasse de manière significative les algorithmes existants.

Chapitre 4 :

1. Ce chapitre présente un nouvel algorithme permettant de segmenter les cycles respiratoires des souris (inspiration et expiration) à partir des signaux de pléthysmographie. En intégrant des contraintes physiologiques, la méthode détecte avec précision le début de l'inspiration et de l'expiration, offrant une plus grande robustesse aux variations respiratoires par rapport aux approches précédentes.

Chapitre 5 :

1. Ce chapitre présente une méthode de référence qui compare les cycles respiratoires à l'aide d'un algorithme de clustering combiné à la distance DTW. Cet algorithme donne une représentation symbolique des cycles respiratoires s'appuyant sur la notion de forme et où chaque symbole représente un groupe. Le suivi de ces symboles dans le temps aboutit à une représentation symbolique des signaux de pléthysmographie.

2. Cette approche facilite la découverte de diverses modalités de ventilation qui ne sont pas prises en compte par les descripteurs conventionnels. La représentation

symbolique permet notamment d'identifier les adaptations spécifiques au génotype en cas de déficience enzymatique et révèle diverses réponses à l'exposition aux inhibiteurs.

Chapitre 6 :

1. Section 6.3 décrit une classe de déformations préservant la structure de graphe des séries temporelles tout en garantissant une action transitive (théorème 3). Le lemme 1 décrit des espaces de Hilbert à noyau reproductible appropriés pour coder ces déformations.

2. Annexe B.5 démontre l'identifiabilité du modèle en estimant le véritable paramètre générateur des données synthétiques, et illustre la sensibilité de la méthode en ce qui concerne ses hyperparamètres.

3. Annexe B.6 et B.7 illustrent l'intérêt quantitatif des représentations TS-LDDMM pour les tâches de classification sur différents ensembles de séries temporelles avec un échantillonnage régulier ou irrégulier.

4. Section 6.5.2 montre l'interprétabilité des représentations TS-LDDMM sur l'analyse de la ventilation des souris.

8.5 Publications publiées

Chapitre 2 :

- Thibaut Germain, Charles Truong et Laurent Oudre. "Linear-trend normalization for multivariate subsequence similarity search". In : *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*. IEEE. 2024, p. 167-175

Chapitre 3 :

- Thibaut Germain, Charles Truong et Laurent Oudre. "Persistence-based motif discovery in time series". In : *IEEE Transactions on Knowledge and Data Engineering* (2024)

- Thibaut Germain, Charles Truong et Laurent Oudre. "Interactive motif discovery in time series with persistent homology". In : *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2024, p. 383-387

Chapitre 5 :

- Thibaut Germain et al. "Unsupervised classification of plethysmography signals with advanced visual representations". In : *Frontiers in Physiology* 14 (2023), p. 781

- Thibaut Germain et al. "Unsupervised study of plethysmography signals through DTW clustering". In : *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE. 2022, p. 3396-3400

Chapitre 6 :

- Thibaut Germain et al. “Shape analysis for time series”. In : *Advances in neural information processing systems* (2024)

Appendices

Appendix A

Local scale tasks & rigid deformations appendix

A.1 Datasets

The benchmark is composed of 10 datasets: 6 real-world datasets and 4 synthetic datasets of increasing order of complexity for the task of similarity search and motif discovery. Table A.1 presents the main features of the datasets.

Real-world datasets. We have considered the following real-world datasets:

(R-1) **mitdb-1** [Gol+00; MM01]: The MIT-BIH Arrhythmia Database contains 48 half-hour recordings of two-channel ambulatory electrocardiograms (ECGs) sampled at $360Hz$. Cardiologists annotated the heartbeats according to 19 categories[1]. We divided all recordings into time series of 1 minute and kept the first channel. We selected time series of healthy subjects (id: 100, 101, 103, 117, 122, according to [Sac+22]) that contains only normal heartbeats, and randomly selected 100 time series.

(R-2) **mitdb-2**: We randomly selected 100 one-minute time series in from MIT-BIH dataset. This dataset is more challenging than the previous one as it contains unhealthy heartbeats. Each time series has 1 to 4 patterns, each with several occurrences.

(R-3) **mitdb800** [GPM90]: This database includes 78 half-hour ECG recordings sampled at $120Hz$ with heartbeat annotations (19 categories). We divide all recordings into three-minute time series and keep the first channel. We randomly select 100 time series, and the number of repeated patterns varied between 1 and 4.

(R-4) **ptt-ppg** [Meh+22]: Pule-Transit-Time PPG dataset consists of time series recorded with multiple sensors (sampled at $500Hz$) from healthy subjects performing physical activities. Heartbeats are also annotated. We randomly select a hundred 40-second long signals from the photoplethysmogram (PPG) first channel during the "run" activity.

[1] `https://archive.physionet.org/physiobank/annotations.shtml`

(R-5) **refit** [MSS17]: The original dataset provides aggregate and individual appliance load curves at 8-second sampling intervals from 20 houses in the United Kingdom, recorded over two years. We selected 10 houses and aggregated recordings of the appliances available: dishwasher, food mixer, washing machine, and tumble dryer. The recordings were down-sampled to 32-second intervals and divided into time series of one week. We kept 10 time series for each house in which the appliances were not used simultaneously. This resulted in a dataset of 100 univariate time series with a maximum of 3 motif sets.

(R-6) **arm-coda** [Com+24] is a dataset of 240 multivariate time series collected using 34 Cartesian Optoelectronic Dynamic Anthropometers (CODA) placed on the upper limbs of 16 healthy subjects, each of whom performed 15 predefined movements such as raising their arms or combing their hair. Each sensor records its position in 3D space. To construct the dataset, we kept the left (resp. right) forearm sensor of id 29 (resp. 20) and 5 of the predefined movements: 0,1,4,6,8 (resp. 0,1,4,5,7). We selected the first two occurrences of all movements in the x and y dimensions. Then, the occurrences of the 5 movements were randomly placed along the time axis for each subject, sensor, and dimension. The distance between two consecutive occurrences is sampled uniformly over $[50, 450]$. A Gaussian noise with a signal-to-noise ratio of 0.01 was added to all time series. This resulted in a dataset of 64 univariate time series.

Synthetic datasets. We have generated datasets based on four scenarios of increasing complexity:

(S-1) **pair**: There is 1 pattern of length 100 that repeats twice.

(S-2) **single**: There is 1 pattern of length 100 that repeats 50 times.

(S-3) **fixed**: There are 5 patterns of length 100. For each pattern, the number of occurrences is sampled uniformly between 2 and 10.

(S-4) **variable**: There are 5 patterns with length uniformly sampled between 100 and 200. For each pattern, the number of occurrences is sampled uniformly between 2 and 10.

All time series are generated using the same protocol: occurrences of the N repeated patterns are randomly placed on top of a random walk, and Gaussian noise is added to the resulting time series. For the motif pair dataset, we generated 200 time series for each random walk variance step between 0 and 0.5 by steps of 0.01, and the interval between the occurrences is uniformly sampled over $[100, 900]$. For other scenarios, the amplitude of the random walk (resp. Gaussian noise) is set to 0.2 (resp. 0.1). The interval between two consecutive occurrences is also uniformly sampled over $[10, 90]$ for the single/fixed scenarios and $[20, 180]$ for the variable-length scenario. In all cases, given a length of $l_0 \in \mathbb{N}^*$ and a fundamental frequency of $4Hz$, a pattern is generated as the sum of the sine function of the l_0 first harmonics, with the phases and the amplitudes are uniformly sampled over $[-\pi, \pi]$ and $[-1, 1]$.

Table A.1 N number of repeated patterns, if $< k$, there are at most k patterns. μ_l average pattern length, σ_l standard deviation of pattern length, min/max minimum/maximum pattern length, n time series length, # number of time series.

Type	Name	N	μ_l	σ_l	min/max	n	#
real	(R-1) mitdb-1	1	320	60	215/461	20k	100
	(R-2) mitdb-2	< 4	280	70	69/496	20k	100
	(R-3) mitdb800	< 4	95	25	24/165	20k	100
	(R-4) ptt-ppg	1	325	45	201/461	20k	100
	(R-5) refit	< 3	100	20	47/143	20k	100
	(R-6) arm-coda	5	525	105	272/886	8k	64
synthetic	(S-1) pair	1	100	0	100/100	1k	100
	(S-2) single	1	100	0	100/100	8k	100
	(S-3) fixed	5	100	0	100/100	3k	100
	(S-4) variable	5	150	30	100/200	4k	100

A.2 Metrics: Precision, Recall and F1-score for motif discovery in time series

Motif discovery in time series is an unsupervised event detection task. Like other time series event-based tasks, we evaluate performance with precision, recall, and f1-score metrics [Tat+18]. However, compared to supervised tasks, the computation of these metrics requires the additional step of pairing real and predicted motif sets. In what follows, we propose a resolution of the motif sets assignment problem and detail the metrics' computation.

A.2.1 Motif sets assignment problem

Pairing real and predicted motifs sets is a two-level assignment problem: predicted motif sets must be assigned to real motif sets, and predicted occurrences must be assigned to real ones between paired motif sets. We compute all pairings simultaneously by maximizing the total overlapping between real and predicted motif sets. Technically, let $R = (R_i)_{1 \leq i \leq |R|}$ the real motif sets such that $R_i = (R^s_{i,u}, R^e_{i,u})_{1 \leq u \leq |R_i|}$ is the list of starting and ending sample location of occurrences of the i^{th} motif. Likewise, we define the predicted motif sets $((P^s_{j,v}, P^e_{j,v})_{1 \leq v \leq |P_j|})_{1 \leq j \leq |P|}$ and Σ_N the permutation group of the sequence $(1, \ldots, N)$. Note that we do not enforce the number of motif sets and occurrences to be identical between real and predicted labels. The total overlapping between real and predicted motif sets is defined by:

$$\text{total-overlapping}(R, P) = \max_{(\sigma, \sigma') \in \Sigma_{|R|} \times \Sigma_{|P|}} \sum_{i=1}^{\min(|R|, |P|)} C(R_{\sigma(i)}, P_{\sigma'(i)}) \tag{A.1}$$

where:

$$C(R_i, P_j) = \max_{(\pi,\pi') \in \Sigma_{|R_i|} \times \Sigma_{|P_j|}} \sum_{u=1}^{\min(|R_i|,|P_j|)} overlap(R_{i,\pi(u)}, P_{j,\pi'(u)}) \tag{A.2}$$

and:

$$overlap(R_{i,u}, P_{j,v}) = \max(\min(R^e_{i,u}, P^e_{j,v}) - \max(R^s_{i,u}, P^s_{j,v}), 0) \tag{A.3}$$

Optimal pairings, $(\sigma, \sigma') \in \Sigma_{|R|} \times \Sigma_{|P|}$ and

$$\left\{(\pi_{i,j}, \pi'_{i,j}) \mid \exists u \; s.t \; (i,j) = (\sigma(u), \sigma'(u)),\; \pi_{i,j} \in \Sigma_{|R_i|},\; \pi'_{i,j} \in \Sigma_{|P_i|}\right\} \;,$$

can be efficiently retrieved with the the Hungarian matching algorithm [Kuh55; Sar+21].

A.2.2 Metrics computation

Precision, recall, and f1-score computations rely on the optimal pairings and a threshold $\tau \in [0, 1]$ that controls the overlapping ratio. The metrics average elementary metrics computed between paired motif sets; it can be a macro average with weights $w_i = 1/|R|$ or a weighted average with weights $w_i = |R_i| / \sum_{j=1}^{|R|} |R_j|$. In what follows, (σ, σ') is the optimal pairing between the motif sets of R and P, (π, π') is the optimal pairing between occurrences of R_i and P_j, and $\mathbf{1}$ is the indicator function.

Precision

$$\text{precision}(R, P; \tau) = \sum_{i=1}^{\min(|R|,|P|)} w_{\sigma(i)} * \text{indv-precision}(R_{\sigma(i)}, P_{\sigma'(i)}; \tau),$$

with

$$\text{indv-precision}(R_i, P_j; \tau) = \frac{1}{|P_i|} \sum_{u=1}^{\min(|R_i|,|P_j|)} \mathbf{1}\left(overlap(R_{i,\pi(u)}, P_{j,\pi'(u)}) \geq \tau(P^e_{i,\pi'(u)} - P^s_{i,\pi'(u)})\right) \;.$$

Recall

$$\text{recall}(R_i, P_j) = \sum_{i=1}^{\min(|R|,|P|)} w_{\sigma(i)} * \text{indv-recall}(R_{\sigma(i)}, P_{\sigma'(i)}; \tau)$$

with

$$\text{indv-recall}(R_{\sigma(i)}, P_{\sigma'(i)}; \tau) = \frac{1}{|R_i|} \sum_{u=1}^{\min(|R_i|,|P_j|)} \mathbf{1}\left(overlap(R_{i,\pi(u)}, P_{j,\pi'(u)}) \geq \tau(R^e_{i,\pi'(u)} - R^s_{i,\pi'(u)})\right) \;.$$

F1-score

$$\text{f1-score}(R, P; \tau) = \frac{2 * \text{precision}(R, P; \tau) * \text{recall}(R, P; \tau)}{\text{precision}(R, P; \tau) + \text{recall}(R, P; \tau)}$$

Appendix B

Deformation-based embeddings appendix

B.1 Oriented varifold

Oriented varifold for curves. In this section, we introduce the *oriented varifold* associated with curves. For further readings on curves and surfaces representation as varifolds, readers can refer to [KCC17; CT13]. We associate to $\gamma \in \mathsf{C}^1((a,b),\mathbb{R}^{d+1})$ an *oriented varifold* μ_γ, i.e. a distribution on the space $\mathbb{R}^{d+1}\times\mathbb{S}^d$ defined as follows, for any smooth test function $\omega : \mathbb{R}^{d+1}\times\mathbb{S}^d \to \mathbb{R}$,

$$\mathbb{E}_{Y\sim\mu_\gamma}\left[\omega(Y)\right] = \mu_\gamma(\omega) = \int_a^b \omega\left(\gamma(t), \frac{\dot\gamma(t)}{|\dot\gamma(t)|}\right)|\dot\gamma(t)|dt \ . \tag{B.1}$$

Denoting by W the space of smooth test function, we have that μ_γ belongs to its dual W^*. Thus, a distance on W^* is sufficient to set a distance on oriented varifolds associated to curve and thus on $\mathsf{C}^1((a,b),\mathbb{R}^{d+1})$ by the identification $\gamma \to \mu_\gamma$. Remark that in (TS-LDDMM), γ should be the parametrization of a time series' graph $\mathsf{G}(\mathsf{I}, f)$, i.e. $\gamma : t \in \mathsf{I} \to (t, f(t)) \in \mathbb{R}^{d+1}$ denoting by $f : \mathsf{I} \to \mathbb{R}^d$ the time series. However, in practice, we work with discrete objects. That is why, we set W as an RKHS to use its representation theorem. More specifically [KCC17, Proposition 2 & 4] encourages us to consider a kernel $k : (\mathbb{R}^{d+1}\times\mathbb{S}^d)^2 \to \mathbb{R}$ such that there exist two positive and continuously differentiable kernels k_{pos} and k_{dir}, such that for any $(x,\overrightarrow{u}),(y,\overrightarrow{v}) \in (\mathbb{R}^{d+1}\times\mathbb{S}^d)^2$

$$k((x,\overrightarrow{u}),(y,\overrightarrow{v})) = k_{pos}(x,y)k_{dir}(\overrightarrow{u},\overrightarrow{v}) \ , \tag{B.2}$$

with moreover $k_{dir} > 0$ and k_pos which admits an RKHS W_{pos} dense in the space of continous function on $\mathbb{R}^{d+1}$ vanishing at infinite [Car+10].

Given such a kernel $k : (\mathbb{R}^{d+1}\times\mathbb{S}^d)^2 \to \mathbb{R}$ verifying [KCC17, Proposition 2 & 4], we have that for any $(x,v) \in \mathbb{R}^{d+1}\times\mathbb{S}^d$, $\delta_{(x,\overrightarrow{v})}$ belongs to W^* as a distribution and that the dual metric $\langle\cdot,\cdot\rangle_{\mathsf{W}^*}$ satisfies for any $(x_1,v_1),(x_2,v_2) \in \left(\mathbb{R}^{d+1}\times\mathbb{S}^d\right)^2$,

$$\langle\delta_{(x_1,\overrightarrow{v}_1)},\delta_{(x_2,\overrightarrow{v}_2)}\rangle_{\mathsf{W}^*} = k((x_1,\overrightarrow{v}_1),(x_2,\overrightarrow{v}_2)) \ . \tag{B.3}$$

Thus, given two sets of triplets $X = (l_i, x_i, \overrightarrow{v}_i)_{i\in[\![1,T_0-1]\!]} \in (\mathbb{R}\times\mathbb{R}^{d+1}\times\mathbb{S}^d)^{T_0-1}, Y = (l'_i, y_i, \overrightarrow{w}_i)_{i\in[\![1,T_1-1]\!]} \in (\mathbb{R}\times\mathbb{R}^{d+1}\times\mathbb{S}^d)^{T_1-1}$ and denoting by

$$\mu_X = \sum_{i=1}^{T_0-1} l_i \delta_{(x_i,\overrightarrow{v}_i)}, \quad \mu_Y = \sum_{i=1}^{T_1-1} l'_i \delta_{(y_i,\overrightarrow{w}_i)} \ , \tag{B.4}$$

we have,

$$\begin{array}{rl} |\mu_X-\mu_Y|^2_{\mathsf{W}^*} = & \sum_{i,j=1}^{T_0-1} l_i k((x_i,\overrightarrow{v_i}),(x_i,\overrightarrow{v_i}^0))l_j + \sum_{i,j=1}^{T_1-1} l'_i k((y_i,\overrightarrow{w_i}),(y_i,\overrightarrow{w_i}))l'_j \\ & -2\sum_{i=1}^{T_0-1}\sum_{j=1}^{T_1-1} l_i k((x_i,\overrightarrow{v_i}),(y_i,\overrightarrow{w_i}))l'_j \end{array} \tag{B.5}$$

Then, using the identification $X \mapsto \mu_X, Y \mapsto \mu_Y$, we can define a distance on sets of triplets as $d_{\mathsf{W}^*,3}(X,Y) = |\mu_X-\mu_Y|^2_{\mathsf{W}^*}$.

Now, we aim to discretize the oriented varifold μ_{G} related to a time series' graph $\mathsf{G}(\mathsf{I},f)$ by using a set of triplets. This is carried out by using a discretized version of $\mathsf{G}(\mathsf{I},f)$, i.e. $\tilde{\mathsf{G}} = (g_i = (t_i, f(t_i)))_{i\in[\![1,T]\!]} \in (\mathbb{R}^{d+1})^T$, in the following way: For any $i \in [T-1]$, denoting the center and length of the i^{th} segment $[g_i, g_{i+1}]$ by $c_i = (g_i+g_{i+1})/2$, $l_i = \|g_{i+1}-g_i\|$, and the unit norm vector of direction $\overrightarrow{g_ig_{i+1}}$ by $\overrightarrow{v_i} = (g_{i+1}-g_i)/l_i$, we define the set of triplets $X(\tilde{\mathsf{G}}) = (l_i, c_i, \overrightarrow{v_i})_{i\in[T-1]}$ and its related oriented varifold $\mu_{X(\tilde{\mathsf{G}})} = \sum_{i=1}^{T-1} l_i \delta_{c_i,\overrightarrow{v_i}}$ as in (B.4). This is a valid discretization of the oriented varifold μ_{G} according to [KCC17, Proposition 1]: $\mu_{X(\tilde{\mathsf{G}})}$ converges towards μ_{G} as the size of the descretization mesh $\sup_{i\in[T-1]}|t_{i+1}-t_i|$ converges to 0.

Finally, we define a distance on discretized time series' graphs $\tilde{\mathsf{G}}_1, \tilde{\mathsf{G}}_2$ as $d_{\mathsf{W}^*}(\tilde{\mathsf{G}}_1,\tilde{\mathsf{G}}_2) = d_{\mathsf{W}^*,3}(X(\tilde{\mathsf{G}}_1), X(\tilde{\mathsf{G}}_2))$.

Varifold kernels. Denote the one-dimensional Gaussian kernel by $K^{(a)}_\sigma(x,y) = \exp(-|x-y|^2/\sigma)$ for any $(x,y) \in (\mathbb{R}^a)^2$, $a \in \mathbb{N}$ and $\sigma > 0$. In the implementation, we use the following kernels, for any $((t_1,x_1),(t_2,x_2)) \in (\mathbb{R}^{d+1})^2, ((w_1,v_1),(w_2,v_2)) \in (\mathbb{S}^d)^2$,

$$\begin{cases} k_{pos}(x,y) = K^{(1)}_{\sigma_{pos,t}}(t_1,t_2)K^{(d)}_{\sigma_{pos,x}}(x_1,x_2) \\ k_{pos}(x,y) = K^{(1)}_{\sigma_{dir,t}}(w_1,w_2)K^{(d)}_{\sigma_{dir,x}}(v_1,v_2) \end{cases} \tag{B.6}$$

where $\sigma_{pos,t}, \sigma_{pos,x}, \sigma_{dir,t}, \sigma_{dir,x} > 0$ are hyperparameters. In practice, we select $\sigma_{pos,x} \approx \sigma_{dir,x} \approx 1$ when the times series are centered and normalized. Otherwise we select $\sigma_{pos,x} \approx \sigma_{dir,x} \approx \bar{\sigma}_s$ with $\bar{\sigma}_s$ the average standard deviation of the time series. We choose $\sigma_{pos,t} \approx \sigma_{dir,t} = mf_e$ with f_e the sampling frequency of the time series and $m \in [5]$ an integer depending on the time change between the starting and the target time series graph. The more significant the time change, the higher m should be. The intuition comes from the fact that the width $\sigma_{pos,t}, \sigma_{dir,t}$ rules the time windows used to perform the comparison, and $\sigma_{pos,x}, \sigma_{dir,x}$ affects the space window. The size of the windows should be selected depending on the variations in the data.

B.2 Tuning the hyperparameters of the TS-LDDMM velocity field kernel

The parameter $\sigma_{T,0}$ should be chosen *large* compared the sampling frequency f_e and compared to average standard deviation $\bar{\sigma}_s$ of the time series, e.g $\sigma_{T,0} = 100$ as $\bar{\sigma}_s \approx f_e \approx 1$.

It makes the time transformation smoother. If $\sigma_{T,0}$ is too small, for instance, $\sigma_{T,0} = f_e$, the effect of the time deformation is too localized, and there are not enough samples to make it visible.

The parameter $\sigma_{T,1}$ should be of the same order as f_e: two different points in time can have various space transformations. σ_x should be of the same order of $\bar{\sigma}_s$: two points with a big difference regarding space compared to $\bar{\sigma}_s$ can have very different space transformations.

We take $c_0 \approx 10c_1$, we want to encourage time transformation before space transformation. We take $(c_0, c_1) = (1, 0.1)$ in all experiments.

B.3 Experimental settings

All experiments were performed on a Debian 6.1.69-1 server with NVIDIA RTX A2000 12GB GPU, Intel(R) Xeon(R) Gold 5220R CPU @ 2.20GHz, and 250 GB of RAM. The source code will be available on Github.

We implemented TS-LDDMM in Python with the JAX library [1].

Initialization. As initialization of (6.14), all momentum parameters are set to 0, and the initial graph of reference is picked from the dataset such that its length is equal to the median length observed in the dataset.

Gradient descent. The chosen gradient descent method is "adabelief" [Zhu+20] implemented in the OPTAX library [2]. The gradient descent has two main parameters: the number of steps (nb_steps) and the maximum stepsize value (η_M). The stepsize has a scheduling scheme:

- Warmup period on 0.1× nb_steps steps: the stepsize increases linearly from 0 to η_M. The goal is to learn progressively the parameters. If the step size is too large at the start, smaller steps at the end cannot make up for the mistakes made at the beginning.

- Fine tuning periode on 0.9× nb_steps : the stepsize decreases from η_M to 0 with a cosine decay implemented in the OPTAX scheduler, i.e. the decreasing factor as the form $0.5(1 + \cos(\pi t/T))$.

By default, we set nb_steps to 400 and η_M to 0.1.

B.4 Datasets

Shape-based UCR/UEA time series classification datasets. We selected 15 shape-based datasets (7 univariates and 8 multivariates) from the from the University of East Anglia (UEA) and the University of California Riverside (UCR) Time Series

[1] https://github.com/google/jax
[2] https://optax.readthedocs.io/en/latest/

Classification Repository[3] [Dau+19; Bag+18]. All datasets were downloaded with the python package aeon[4]. Essential datasets information are summarized in Table B.1 and further can be found in [Dau+19; Bag+18].

Table B.1 UCR/UEA shape-based time series datasets for classification.

	Dataset	Size	Lengh	Number of classes	Number of dimensions	Type
Univariate	ArrowHead	211	251	3	1	IMAGE
	BME	180	128	3	1	SIMULATED
	ECG200	200	96	2	1	ECG
	FacesUCR	2250	131	14	1	IMAGE
	GunPoint	200	150	2	1	MOTION
	PhalangesOutlinesCorrect	2658	80	2	1	IMAGE
	Trace	200	275	4	1	SENSOR
Multivariate	ArticularyWordRecognition	575	144	25	9	SENSOR
	Cricket	180	1197	12	6	MOTION
	ERing	60	65	6	4	SENSOR
	Handwriting	1000	152	26	3	MOTION
	Libras	360	45	15	2	VIDEO
	NATOPS	360	51	6	24	MOTION
	RacketSports	303	30	4	6	SENSOR
	UWaveGestureLibrary	240	315	8	3	SENSOR

B.5 TS-LDDMM representation identifiability

In this experiment, we evaluate the ability of TS-LDDMM to retrieve the parameter v_0^* that encodes the deformation $\exp_{Id}(v_0^*)$ acting on a time series graph G by solving the geodesic shooting problem (6.13) between G and $\exp_{Id}(v_0^*) \cdot \mathsf{G}$. Parameter identifiability is an important property for subsequent statistical analysis. Results show that TS-LDDMM representations are identifiable or weakly identifiable depending on the velocity field kernel K_G specification.

Settings. This experiment only involves the TS-LDDMM method in two different settings:

- **The velocity field kernel K_G is well-specified:** The velocity field kernel K_G is set to $(c_0, c_1, \sigma_{T,0}, \sigma_{T,1}, \sigma_x) = (1, 0.1, 100, 1, 1)$, the varifold loss kernels (k_{pos}, k_{dir}) are set to $(\sigma_{pos,t}, \sigma_{pos,t}, \sigma_{dir,t}, \sigma_{dir,x}) = (2, 1, 2, 0.6)$, and the optimizer has 400 steps with a maximum stepsize η_M of 0.05.

- **The velocity field kernel K_G is missspecified:** The velocity field kernel K_G is set with $(c_0, c_1, \sigma_{T,1}) = (1, 0.1, 1)$, $\sigma_{T,0}$ ranging in $(1, 5, 10, 50, 100, 200, 300)$, and σ_x ranging in $(0.1, 1, 10, 100)$. The varifold loss kernels (k_{pos}, k_{dir}) are set to $(\sigma_{pos,t}, \sigma_{pos,t}, \sigma_{dir,t}, \sigma_{dir,x}) = (2, 1, 2, 0.6)$, and the optimizer has 400 steps with a maximum stepsize η_M of 0.05.

[3] https://timeseriesclassification.com
[4] https://www.aeon-toolkit.org/en/stable/

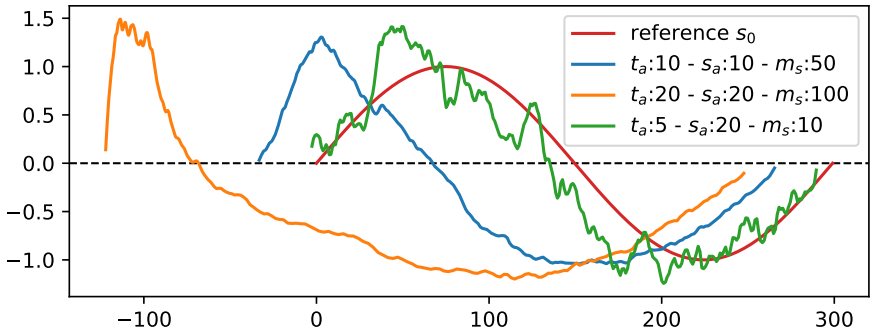


Figure B.1 Plots of $\exp_{Id}(v_0(\alpha^*, \mathbf{x})) \cdot \mathbf{x}$ for different values of α^* according to its sampling parameter t_a, s_a, m_s, taking $\mathbf{x} = \mathsf{G}(s_0)$ with $s_0 : k \in [300] \to \sin(2\pi k/300)$.

Table B.2 Values of $\mathcal{L}(\exp_{Id}(v_0(\alpha^*, \mathbf{x})) \cdot \mathbf{x}, \exp_{Id}(v_0) \cdot \mathbf{x})$ as α^* is sampled according to Gen(10,10,50) and $\hat{v}_0$ is estimated using K_{G} with varying parameters $\sigma_{T,1}, \sigma_x$.

$\sigma_{T,0} \backslash \sigma_x$	1	10	50	100	200	300
0.1	2e+0	3e-4	1e-5	4e-6	7e-4	4e-3
1	4e-2	1e-4	1e-5	4e-6	7e-4	4e-3
100	4e-2	2e-4	1e-5	4e-6	7e-4	4e-3

provided that the hyperparameters and the reference graph are wisely selected, i.e., the parameter v_0^* generating a deformation $\exp_{Id}(v_0^*)$ of a time series graph G can be estimated from the data $\mathsf{G}, \exp_{Id}(v_0^*) \cdot \mathsf{G}$ by solving the geodesic shooting problem (6.13).

The velocity field kernel K_G is well specified. First, we show the model identifiability when the kernel K_G is well specified: the estimated parameter is a good approximation of the generating parameter when the generation and the estimation procedure use the same hyperparameters for the RKHS kernel K_{G}. All the hyperparameter values for generation and estimation are given in Appendix B.5.

We fix the initial control points as $\mathbf{x} = (x_k = (k, \sin(2\pi k/300)))_{k \in [\![1,300]\!]}$. Given $m_s \in \mathbb{N}^*$ and $t_a, s_a > 0$, we randomly generate initial momentums $\alpha^* = (\alpha_k^*)_{k \in [\![1,n_0]\!]}$ with the following sampling, called Gen(m_s, t_a, s_a): For any $k \in [\![1, n_0]\!]$, α'_k is sampled according to a Gaussian normal distribution $\mathcal{N}(0_{d+1}, I_{d+1})$. Then, $(\alpha'_k)_{k \in [\![1,n_0]\!]}$ is regularized by a rolling average of size m_s, we get $\bar{\alpha}' = (\bar{\alpha}'_k)_{k \in [\![1,n_0]\!]}$. Finally, we normalize $\bar{\alpha}'$ to derive α^* such that $|([\alpha_k^*]_t)_{k \in [\![1,n_0]\!]}| = t_{\text{amp}}$ and $|([\alpha_k^*]_s)_{k \in [\![1,n_0]\!]}| = s_{\text{amp}}$ for any $k \in [\![1, n_0]\!]$, denoting by $[\alpha_k^*]_t, [\alpha_k^*]_s$ the time and space coordinates of α_k^* respectively. Note that the regularizing step $(\alpha'_k)_{k \in [\![1,n_0]\!]} \to \bar{\alpha}'$ is necessary to obtain realistic deformations which take into account the regularity induced by the RKHS V.

Then, using $v_0(\alpha^*, \mathbf{x})$ as defined in (6.9) with initial momentums α^* and control points $\mathbf{x}$, we apply the induced deformation $\exp_{Id}(v_0)$ by (6.12) to $\mathbf{x}$ and obtain $\exp_{Id}(v_0) \cdot \mathbf{x}$. Finally, we solve (6.13) to recover an estimation $\hat{\alpha}$ of α^* and report the average relative error (ARE) $|v_0(\hat{\alpha}, \mathbf{x}) - v_0(\alpha^*, \mathbf{x})|_{\mathsf{V}} / |v_0(\alpha^*, \mathbf{x})|_{\mathsf{V}}$ on 50 repetitions. This procedure is

performed for any $m_s, t_a, s_a \in \{10, 50, 100\} \times \{5, 10, 15, 20\}^2$. Mean, standard deviation, and maximum of the ARE on all these hyperparameters choices are respectively $\mathbf{0.10}, \mathbf{0.03}, \mathbf{0.17}$. Therefore, the estimation procedure (6.13) offers a good approximation of the true parameter when the kernel K_{G} is well specified. We observe that the estimation is difficult when $t_a \ll s_a$ because the time series can be very noisy as illustrated in Figure B.1: this impacts the Varifold loss which is sensitive to tangents.

The velocity field kernel K_G is misspecified. We demonstrate a weak identifiability when the kernel K_{G} is misspecified: we can reconstruct the graph time series' after deformations even if the hyperparameters of K_{G} are different during the generation and the estimation. The hyperparameters of K_{G} during generation are $(c_0, c_1, \sigma_{T,0}, \sigma_{T,1}, \sigma_x) = (1, 0.1, 100, 1, 1)$ and we fix $\sigma_{T,1}, c_0, c_1 = (1, 1, 0.1)$ for K_{G} during estimation. We aim to understand the impact of $\sigma_{T,1}, \sigma_x$ on the reconstruction since they are encoding the smoothness of the transformation according to time and space.

For any choice of the hyperparameters $\sigma_{T,1}, \sigma_x \in \{1, 10, 50, 100, 200, 300\} \times \{0.1, 1, 100\}$ related to K_{G} in the estimation, we average $\mathcal{L}(\exp_{Id}(v_0(\alpha^*, \mathbf{x})) \cdot \mathbf{x}, \exp_{Id}(\hat{v}_0) \cdot \mathbf{x})$ on 50 repetitions when α^* is sampled according to $\mathrm{Gen}(10, 10, 50)$ and $\hat{v}_0 = v_0(\hat{\alpha}, \mathbf{x})$ denoting by $\hat{\alpha}$ the result of the minimization (6.13). We observe in Table B.2 that the reconstruction is almost perfect except in the case when $\sigma_{t,0} = 1$ during estimation, while $\sigma_{t,0} = 100$ during generation. Compared to $\sigma_{T,0}$, σ_x has nearly no impact on the reconstruction. In Appendix B.1-B.2, we propose guidelines to drive future hyperparameters tuning and further discussions related to $\sigma_{T,1}, c_0, c_1$.

B.6 Robustness to irregular sampling

This experiment is inspired by [OLK24] where the authors perform an extensive comparison of Neural Ordinary Differential Equations (Neural ODEs) methods [Kid+20]. We assess the classification performances of several methods under regular sampling (0% missing rate) and three irregular sampling regimes on 15 shape-based datasets (7 univariate & 8 multivariate). Methods and training strategy are taken from its associated Github[5] and described in what follows. We conclude with the results, which show that our method, TS-LDDMM, outperforms all methods for sampling regimes with missing rates: 0%, 30%, and 50%.

B.6.1 Benchmark methods

In related work, we give an overview of Neurals ODEs methods and their relation with TS-LDDMM.

- RNN-based methods: Baseline reccurent neural networks including RNN [MJ99], LSTM [HS97], and GRU [Chu+14].

[5] `https://github.com/yongkyung-oh/Stable-Neural-SDEs`

- Attention-based methods: Multi-Time Attention Networks (MTAN) [SM21] and Multi-Integration Attention Module (MIAM) [Lee+22]. Both handle multivariate time series irregularly sampled with attention mechanisms.
- Neural ODEs: ODE-LSTM [LH20] a form of Neural-ODEs used to learn continuous latent representations.
- Neural SDEs: Neural SDE [Liu+19] and Neural LNSDE [OLK24] have been proposed to model randomness in time-series using drift and diffusion terms as an extension of Neural-ODEs.
- Shape-Analysis methods: TS-LDDMM (ours) and LDDMM [Gla+08]. From shape analysis, both methods learn representations by solving ODEs parametrized with Kernels. While both methods handle multivariate signals irregularly sampled, TS-LDDMM is specifically designed for time series.

B.6.2 Model settings

Neural ODEs methods. As depicted in [OLK24], any Neural ODEs layer in Appendix B.6.1 is followed by an MLP with two fully connected layers with `ReLU` activations. The risk of overfitting and the model regularization are handled with a dropout rate of 10% and an early-stopping mechanism, ceasing the training when the validation loss does not improve for 10 successive epochs.

For each method and dataset, the learning rate, the hidden vector dimensions, and the number of layers are optimized to minimize the CrossEntropy loss on a validation set using the `Ray` [6] Python library. The learning rate varies from 10^{-4} to 10^{-1} using log uniform search, the hidden vector dimension ranges from $16, 32, 64, 128$ using grid search, and the number of layers ranges from $1, 2, 3, 4$ using grid search. The batch size was selected from $16, 32, 64, 128$ according to the size of the dataset. All methods were trained for 100 epochs, and the best method was selected based on the lowest validation loss.

TS-LDDMM and LDDMM. Representations learned with TS-LDDMM or LDDMM by solving the atlas estimation problem (6.14) are fed to a Support Vector Classifier (SVC) from `scikit-learn` [7]. All SVC's hyperparameters are set to default except the regularization term C, which is set through grid search on a validation set with the macro f1-score [8].

To learn TS-LDDMM (resp. LDDMM) representations, the velocity field kernel K_G is set to $(c_0, c_1, \sigma_{T,0}, \sigma_{T,1}, \sigma_x) = (1, 0.1, 0.33\bar{l}, 1, n_d)$, (resp. $(\sigma_T, \sigma_x) = (0.33\bar{l}, n_d)$) where $\bar{l}$ is the average time series length and n_d the number of dimensions. For both methods and all datasets, the varifold loss kernels (k_{pos}, k_{dir}) are identical and set to $(\sigma_{pos,t}, \sigma_{pos,t}, \sigma_{dir,t}, \sigma_{dir,x}) = (2, n_d, 2, n_d)$. For TS-LDDMM (resp. LDDMM), the optimizer is set with 400 epochs (resp. 400) and a maximum learning rate $\eta_M = 0.1$ (resp.

[6] `https://github.com/ray-project/ray`
[7] `https://scikit-learn.org/stable/`
[8] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html`

$\eta_M = 0.01$). In all cases, the initial reference graph is selected in the dataset as a time series with the median length.

B.6.3 Protocol

In this experiment, we investigate the robustness to missing samples and the classification performance of TS-LDDMM compared to Neural ODEs on 15 datasets described in Appendix B.4. For fairness between methods of different architectures, the evaluation protocol on each dataset and method is as follows:

1. Spilt the dataset in train 75%, validation 15%, and test 15%.
2. Tune hyperparameters with train and validation sets and a missing rate of 0%.
3. For each missing rate in [0%,30%,50%,70%]
    - Remove samples in time series in the train and test sets according to the missing rate and the drop procedure described in [Kid+20].
    - Train the model on the train set
    - Evaluate the macro f1-score on the test set

B.6.4 Results

In this experiment, we investigate the robstuness to missing samples and the classification performance of TS-LDDMM representations. We compare TS-LDDMM with LDDMM and 8 neural ODEs networks. Performances are evaluated in terms of average macro f1-score and rank on four different regimes of missing rate 0%,30%,50%, and 70%. Results are aggregated in Table B.3.

On three out of four regimes (0%,30%, and 50%) TS-LDDMM classifier is the best performer in terms of f1-score and rank. For missing rates of 0% and 30%, the score increases by 10% compared to the second-best performer, LDDMM. However, LDDMM is not the second-best performer in rank (Neural LNSDE), showing its sensitivity to parameterization, unlike TS-LDDMM, which remains consistent. Performances of Neural LNSDE remain constant with the increase of the missing rate as observed in [OLK24], and it becomes the best performer for missing rate 70%. The decrease in TS-LDDMM performances with the increasing missing rate is due to the varifold loss, which poorly approximates the time series shape. Other losses might be more relevant for high missing rates.

Overall, TS-LDDMM is a relevant and consistent shape-based representation for irregularly sampled multivariate time series for missing rates up to 50% .

B.7 Classification benchmark on regularly sampled datasets

In this section, we compare the classification performances of TS-LDDMM with other methods from shape analysis on 15 shape-based datasets of time series regularly sampled. TS-LDDMM outperforms other methods on 12 out of 15, highlighting its relevance for shape analysis when dealing with time series.

Table B.3 Comparison of average macro f1-score and rank as the sample dropping rate increases. **First** & second best performers. TS-LDDMM is the best performer on three out of four regimes.

Methods	Regular		30 % dropped		50 % dropped		70 % dropped	
	F1-score	Rank	F1-score	Rank	F1-score	Rank	F1-score	Rank
RNN (1999)	0.64 ± 0.21	6.2	0.53 ± 0.23	6.6	0.48 ± 0.21	7.2	0.44 ± 0.21	6.07
LSTM (1997)	0.61 ± 0.29	6.0	0.57 ± 0.29	6.27	0.53 ± 0.25	6.07	0.51 ± 0.29	5.27
GRU (2014)	0.71 ± 0.26	4.2	0.68 ± 0.28	4.27	0.66 ± 0.28	3.73	$\underline{0.59 \pm 0.28}$	$\underline{3.67}$
MTAN (2021)	0.59 ± 0.28	7.13	0.58 ± 0.28	5.8	0.54 ± 0.29	5.33	0.51 ± 0.28	5.0
MIAM (2022)	0.48 ± 0.35	6.93	0.42 ± 0.33	8.27	0.47 ± 0.31	6.93	0.35 ± 0.31	7.6
ODE-LSTM (2020)	0.63 ± 0.24	6.0	0.57 ± 0.25	6.53	0.51 ± 0.24	7.27	0.45 ± 0.23	6.73
Neural SDE (2019)	0.48 ± 0.28	7.67	0.47 ± 0.26	7.47	0.45 ± 0.27	7.13	0.45 ± 0.25	6.0
Neural LNSDE (2024)	0.7 ± 0.27	$\underline{3.87}$	0.68 ± 0.29	$\underline{4.0}$	$\underline{0.67 \pm 0.25}$	$\underline{3.53}$	$\mathbf{0.66 \pm 0.23}$	**2.47**
LDDMM (2008)	$\underline{0.72 \pm 0.2}$	4.53	$\underline{0.7 \pm 0.21}$	4.2	0.57 ± 0.25	5.0	0.4 ± 0.25	7.13
TS-LDDMM (ours)	$\mathbf{0.83 \pm 0.18}$	**2.93**	$\mathbf{0.8 \pm 0.18}$	**2.07**	$\mathbf{0.7 \pm 0.26}$	**3.33**	0.51 ± 0.27	5.67

B.7.1 Benchmark methods

- SRV-based method: we include TCLR [Heo+24] a logistic regression on the tangent space of the Frechet mean with Square Root Velocity (SRV representation). We also include Shape-FPCA [WHS24] that encodes both the time series and its time parameterization.

- LDDMM-Based : TS-LDDMM (ours) and LDDMM [Gla+08]. Both methods learn representations by solving ODEs parametrized with Kernels. While both methods handle multivariate signals, TS-LDDMM is specifically designed for time series.

B.7.2 Model settings

TCLR & Shape-FPCA. Shape-FPCA is available in the Python library `FDASRSF` [9]. Once the shape-FPCA representations are learned, they are fed to an SVC from `scikit-learn`. `FDASRSF` provides SRV representation methods that we combined with a logistic regression from `scikit-learn` to implement TCLR. For both methods, the number of steps to learn the Frechet mean is set to 50, and the regularization hyperparameter C is set through grid search on a validation set with the macro f1-score. Other parameters are set to default.

TS-LDDMM & LDDMM. Representations learned with TS-LDDMM or LDDMM by solving the atlas estimation problem (6.14) are fed to an SVC from `scikit-learn`. All SVC's hyperparameters are set to default except the regularization term C, which is set through grid search on a validation set with the macro f1-score.

To learn TS-LDDMM (resp. LDDMM) representations, the velocity field kernel K_G is set to $(c_0, c_1, \sigma_{T,0}, \sigma_{T,1}, \sigma_x) = (1, 0.1, 0.33\bar{l}, 1, n_d)$, (resp. $(\sigma_T, \sigma_x) = (0.33\bar{l}, n_d)$) where $\bar{l}$ is the average time series length and n_d the number of dimensions. For both methods and all datasets, the varifold loss kernels (k_{pos}, k_{dir}) are identical and set to $(\sigma_{pos,t}, \sigma_{pos,t}, \sigma_{dir,t}, \sigma_{dir,x}) = (2, n_d, 2, n_d)$. For TS-LDDMM (resp. LDDMM), the optimizer is set with 400 epochs (resp. 400) and a maximum learning rate $\eta_M = 0.1$ (resp.

[9] https://fdasrsf-python.readthedocs.io/en/latest/

$\eta_M = 0.01$). In all cases, the initial reference graph is selected in the dataset as a time series with the median length.

Protocole. For each dataset and method, the evaluation protocol is a simple train,validation test with hyperparameter tuning:

1. Split The dataset in train 75%, validation 15%, and test 15%.
2. Training and hyperparameters tuning with train and validation sets
3. Evaluate the macro f1-score on the test set

B.7.3 Results

In this experiment, we investigate the classification performances of several methods from shape analysis on 15 shape-based time series datasets (7 univariate and 8 multivariate). The performances are evaluated in terms of macro f1-score. Results are aggregated in Table B.4.

The TS-LDDMM-based classifier outperforms other methods on 12 out of 15 datasets. TCLR is the second-best performer on univariate datasets; however, its current implementation with `FDASRSF` does not extend to the multivariate case, which limits usage. LDDMM performances are lower than TCLR, and Shape-FPCA is the worst performer.

Overall, TS-LDDMM representations are well suited for shape-based time series classification, and its extension to multivariate irregularly sampled time series makes it a relevant option for time series shape analysis.

Table B.4 F1-score comparison between methods from shape analysis on 15 datasets. **First** and second best performers.

	Dataset	Shape-FPCA (2024)	TCLR (2024)	LDDMM (2008)	TS-LDDMM (ours)
Univariate	ArrowHead	0.18	0.75	0.84	**0.91**
	BME	0.16	1.00	0.82	**1.00**
	ECG200	0.40	0.67	**0.81**	0.79
	FacesUCR	0.08	0.73	0.69	**0.86**
	GunPoint	0.93	0.97	0.83	**1.00**
	PhalangesOutlinesCorrect	0.39	**0.63**	0.53	0.52
	Trace	0.55	1.00	0.46	**1.00**
Multivariate	ArticularyWordRecognition	–	–	0.98	**1.00**
	Cricket	–	–	0.77	**0.93**
	ERing	–	–	0.95	**0.98**
	Handwriting	–	–	0.22	**0.44**
	Libras	–	–	0.56	**0.60**
	NATOPS	–	–	0.82	**0.82**
	RacketSports	–	–	**0.83**	0.79
	UWaveGestureLibrary	–	–	0.72	**0.81**

B.8 Mice ventilation analysis with TS-LDDMM

Settings. This experiment involves TS-LDDMM and LDDMM [Gla+08] methods. Both methods are run twice, first on respiratory cycles before exposure to the irritant molecule to capture mice breathing behavior at rest and on all respiratory cycles to capture the

influence of the irritant molecule. Exposure to the irritant molecule leads to significant shape deformation in the respiratory cycles, and the terms must be added to the varifold loss to capture deformations at a large time scale.

TS-LDDMM parameters.

- **Before exposure:** The velocity field kernel K_G is set to $(c_0, c_1, \sigma_{T,0}, \sigma_{T,1}, \sigma_x) = (1, 0.1, 150, 1, 2)$. The varifold loss is the sum of three varifolds to capture shapes variations at different scales with parameters: (Varifold 1,Varifold 2,Varifold 3): $((5, 2, 5, 1), (2, 1, 2, 0.6), (1, 0.6, 1, 0.6))$ and the mapper $(\sigma_{pos,t}, \sigma_{pos,t}, \sigma_{dir,t}, \sigma_{dir,x})$. The optimizer has 800 steps with a maximum stepsize η_M of 0.3.
- **Before/after exposure:** The velocity field kernel K_G is set to $(c_0, c_1, \sigma_{T,0}, \sigma_{T,1}, \sigma_x) = (1, 0.1, 220, 1, 2)$. The varifold loss is the sum of four varifolds to capture shapes variations at different scales with parameters: (Varifold 1,Varifold 2,Varifold 3, Varifold 4): $((30, 2, 30, 1), (5, 2, 5, 1), (2, 1, 2, 0.6), (1, 0.1, 1, 0.1))$ and the mapper $(\sigma_{pos,t}, \sigma_{pos,t}, \sigma_{dir,t}, \sigma_{dir,x})$. The optimizer has 800 steps with a maximum stepsize η_M of 0.3.

LDDMM parameters. Note that varifold losses are unchanged between TS-LDDMM and LDDMM. Compared to TS-LDDMM, the convergence of LDDMM is more sensitive to the maximum stepsize η_m, which must remain small for LDDMM to guarantee the convergence.

- **Before exposure:** The velocity field kernel K_G is an anysotropic Gaussian kernel with parameters $\sigma_T = 150$ for the time dimension and $\sigma_x = 2$ for space dimensions. The varifold loss is the sum of three varifolds to capture shapes variations at different scales with parameters: (Varifold 1,Varifold 2,Varifold 3): $((5, 2, 5, 1), (2, 1, 2, 0.6), (1, 0.6, 1, 0.6))$ and the mapper $(\sigma_{pos,t}, \sigma_{pos,t}, \sigma_{dir,t}, \sigma_{dir,x})$. The optimizer has 800 steps with a maximum stepsize η_M of 0.01.
- **Before/after exposure:** The velocity field kernel K_G is an anysotropic Gaussian kernel with parameters $\sigma_T = 220$ for the time dimension and $\sigma_x = 2$ for space dimensions. The varifold loss is the sum of four varifolds to capture shapes variations at different scales with parameters: (Varifold 1,Varifold 2,Varifold 3, Varifold 4): $((30, 2, 30, 1), (5, 2, 5, 1), (2, 1, 2, 0.6), (1, 0.1, 1, 0.1))$ and the mapper $(\sigma_{pos,t}, \sigma_{pos,t}, \sigma_{dir,t}, \sigma_{dir,x})$. The optimizer has 800 steps with a maximum stepsize η_M of 0.01.

Bibliography

[AML19] Amaia Abanda, Usue Mori, and Jose A Lozano. "A review on distance based time series classification". In: *Data Mining and Knowledge Discovery* 33.2 (2019), pp. 378–412.

[Abd+20] Alireza Abdoli et al. "Fitbit for chickens? Time series data mining can increase the productivity of poultry farms". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 3328–3336.

[ASW15] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. "Time-series clustering–a decade review". In: *Information systems* 53 (2015), pp. 16–38.

[AFS93] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. "Efficient similarity search in sequence databases". In: *Foundations of Data Organization and Algorithms: 4th International Conference, FODO'93 Chicago, Illinois, USA, October 13–15, 1993 Proceedings 4*. Springer. 1993, pp. 69–84.

[AKK20] Sara Alaee, Kaveh Kamgar, and Eamonn Keogh. "Matrix profile XXII: exact discovery of time series motifs under DTW". In: *2020 IEEE international conference on data mining (ICDM)*. IEEE. 2020, pp. 900–905.

[AAT07] Stéphanie Allassonnière, Yali Amit, and Alain Trouvé. "Towards a coherent statistical framework for dense deformable template estimation". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 69.1 (2007), pp. 3–29.

[AM14] Joakim Andén and Stéphane Mallat. "Deep scattering spectrum". In: *IEEE Transactions on Signal Processing* 62.16 (2014), pp. 4114–4128.

[Ans+23] Abdul Fatir Ansari et al. "Neural continuous-discrete state space models for irregularly-sampled time series". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 926–951.

[AV07] David Arthur and Sergei Vassilvitskii. "K-Means++: The Advantages of Careful Seeding". In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '07. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. ISBN: 9780898716245.

[Avi03] Yossi Aviv. "A time-series framework for supply-chain inventory management". In: *Operations Research* 51.2 (2003), pp. 210–227.

[BHL14] Anthony Bagnall, Jon Hills, and Jason Lines. "Finding motif sets in time series". In: *arXiv preprint arXiv:1407.3685* (2014).

[Bag+17] Anthony Bagnall et al. "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances". In: *Data mining and knowledge discovery* 31 (2017), pp. 606–660.

[Bag+18] Anthony Bagnall et al. "The UEA multivariate time series classification archive, 2018". In: *arXiv preprint arXiv:1811.00075* (2018).

[Bar+15] Rémi Barrois et al. "Quantify osteoarthritis gait at the doctor's office: a simple pelvis accelerometer based method independent from footwear and aging". In: *Computer Methods in Biomechanics and Biomedical Engineering* 18.sup1 (2015), pp. 1880–1881.

[BT70] D. Bartlett and SM. Tenney. "Control of breathing in experimental anemia". In: *Respiration physiology* 10.3 (1970), pp. 384–395.

[BI03] J HT Bates and C G Irvin. "Measuring lung function in mice: the phenotyping uncertainty principle". In: *Journal of applied physiology* 94.4 (2003), pp. 1297–1306.

[Bau+21] Martin Bauer et al. "Intrinsic riemannian metrics on spaces of curves: Theory and computation". In: *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging: Mathematical Imaging and Vision* (2021), pp. 1–35.

[Bec+90] Norbert Beckmann et al. "The R*-tree: An efficient and robust access method for points and rectangles". In: *Proceedings of the 1990 ACM SIGMOD international conference on Management of data*. 1990, pp. 322–331.

[Beg+05] M Faisal Beg et al. "Computing large deformation metric mappings via geodesic flows of diffeomorphisms". In: *International journal of computer vision* 61 (2005), pp. 139–157.

[Ben+20] Andrew Van Benschoten et al. "MPA: a novel cross-language API for time series analysis". In: *Journal of Open Source Software* 5.49 (2020), p. 2179. DOI: 10.21105/joss.02179. URL: https://doi.org/10.21105/joss.02179.

[BT11] Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[Ber+14] Gordon J Berman et al. "Mapping the stereotyped behaviour of freely moving fruit flies". In: *Journal of The Royal Society Interface* 11.99 (2014), p. 20140672.

[Ber+11] Véronique Bernard et al. "Distinct localization of collagen Q and PRiMA forms of acetylcholinesterase at the neuromuscular junction". In: *Molecular and Cellular Neuroscience* 46.1 (2011), pp. 272–281.

[BC94] D. J. Berndt and J. Clifford. "Using dynamic time warping to find patterns in time series". In: *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*. AAAIWS'94. Seattle, WA: AAAI Press, 1994, pp. 359–370.

[Bha+18] Karthik Bharath et al. "Radiologic image-based statistical shape analysis of brain tumours". In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 67.5 (2018), pp. 1357–1378.

[BAM23] Munish Bhatia, Tariq Ahamed Ahanger, and Ankush Manocha. "Artificial intelligence based real-time earthquake prediction". In: *Engineering Applications of Artificial Intelligence* 120 (2023), p. 105856.

[Boe+20] Wolfgang Boedeker et al. "The global distribution of acute unintentional pesticide poisoning: estimations based on a systematic review". In: *BMC public health* 20 (2020), pp. 1–19.

[BTO24] Alexandre Bois, Brian Tervil, and Laurent Oudre. "Persistence-based clustering with outlier-removing filtration". In: *Frontiers in Applied Mathematics and Statistics* 10 (2024), p. 1260828.

[BCY18] Jean-Daniel Boissonnat, Frédéric Chazal, and Mariette Yvinec. *Geometric and topological inference*. Vol. 57. Cambridge University Press, 2018.

[Bor+06] Karsten M Borgwardt et al. "Integrating structured biological data by kernel maximum mean discrepancy". In: *Bioinformatics* 22.14 (2006), e49–e57.

[Bou+09] E. Boudinot et al. "Influence of differential expression of acetylcholinesterase in brain and muscle on respiration". In: *Respiratory physiology & neurobiology* 165.1 (2009), pp. 40–48.

[Bru+22] S. Bruggink et al. "A leak-free head-out plethysmography system to accurately assess lung function in mice". In: *Journal of Applied Physiology* 133.1 (2022), pp. 104–118.

[Cam+08] Shelley Camp et al. "Acetylcholinesterase expression in muscle is specifically controlled by a promoter-selective enhancesome in the first intron". In: *Journal of Neuroscience* 28.10 (2008), pp. 2459–2470.

[Can06] Kevin Cannard. "The acute treatment of nerve agent exposure". In: *Journal of the neurological sciences* 249.1 (2006), pp. 86–94.

[Car+10] Claudio Carmeli et al. "Vector valued reproducing kernel Hilbert spaces and universality". In: *Analysis and Applications* 8.01 (2010), pp. 19–61.

[Cas+15] Maurizio Casarrubea et al. "T-pattern analysis for the study of temporal structure of animal and human behavior: a comprehensive review". In: *Journal of neuroscience methods* 239 (2015), pp. 34–46.

[Cha+22] Sujana Chandrasekar et al. "Similarity analysis for thermal signature comparison in metal additive manufacturing". In: *Materials & Design* 224 (2022), p. 111261.

[CCT17] Benjamin Charlier, Nicolas Charon, and Alain Trouvé. "The fshape framework for the variability analysis of functional shapes". In: *Foundations of Computational Mathematics* 17 (2017), pp. 287–357.

[Cha13] Nicolas Charon. "Analysis of geometric and functional shapes with extensions of currents: applications to registration and atlas estimation". PhD thesis. École normale supérieure de Cachan-ENS Cachan, 2013.

[CT13] Nicolas Charon and Alain Trouvé. "The varifold representation of nonoriented shapes for diffeomorphic registration". In: *SIAM journal on Imaging Sciences* 6.4 (2013), pp. 2547–2580.

[CT14] Nicolas Charon and Alain Trouvé. "Functional currents: a new mathematical tool to model and analyse functional shapes". In: *Journal of mathematical imaging and vision* 48 (2014), pp. 413–431.

[Cha+03] Fabrice Chatonnet et al. "Respiratory survival mechanisms in acetylcholinesterase knockout mouse". In: *European Journal of Neuroscience* 18.6 (2003), pp. 1419–1427.

[Che+18] Ricky TQ Chen et al. "Neural ordinary differential equations". In: *Advances in neural information processing systems* 31 (2018).

[CCN10] Yueguo Chen, Ke Chen, and Mario A Nascimento. "Effective and efficient shape-based pattern detection over streaming time series". In: *IEEE Transactions on Knowledge and Data Engineering* 24.2 (2010), pp. 265–278.

[Chu+14] Junyoung Chung et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014).

[Chu+09] Marie Chupin et al. "Fully automatic hippocampus segmentation and classification in Alzheimer's disease and mild cognitive impairment applied on data from ADNI". In: *Hippocampus* 19.6 (2009), pp. 579–587.

[Cle+90] Robert B Cleveland et al. "STL: A seasonal-trend decomposition". In: *J. Off. Stat* 6.1 (1990), pp. 3–73.

[Coh+21] Samuel Cohen et al. "Aligning time series on incomparable spaces". In: *International conference on artificial intelligence and statistics*. PMLR. 2021, pp. 1036–1044.

[Com24] Sylvain Combettes. "Symbolic representations of time series". PhD thesis. Université Paris-Saclay, 2024.

[Com+24] Sylvain W Combettes et al. "Arm-CODA: A Data Set of Upper-limb Human Movement During Routine Examination". In: *Image Processing On Line* 14 (2024), pp. 1–13.

[CB17] Marco Cuturi and Mathieu Blondel. "Soft-dtw: a differentiable loss function for time-series". In: *International conference on machine learning*. PMLR. 2017, pp. 894–903.

[DB07] John A Dani and Daniel Bertrand. "Nicotinic acetylcholine receptors and nicotinic cholinergic mechanisms of the central nervous system". In: *Annu. Rev. Pharmacol. Toxicol.* 47.1 (2007), pp. 699–729.

[Dau+19] Hoang Anh Dau et al. "The UCR time series archive". In: *IEEE/CAA Journal of Automatica Sinica* 6.6 (2019), pp. 1293–1305.

[DAV19] Dieter De Paepe, Diego Nieves Avendano, and Sofie Van Hoecke. "Implications of z-normalization in the matrix profile". In: *International Conference on Pattern Recognition Applications and Methods*. Springer. 2019, pp. 95–118.

[Dem06] Janez Demšar. "Statistical comparisons of classifiers over multiple data sets". In: *The Journal of Machine learning research* 7 (2006), pp. 1–30.

[Din+08] Hui Ding et al. "Querying and mining of time series data: experimental comparison of representations and distance measures". In: *Proceedings of the VLDB Endowment* 1.2 (2008), pp. 1542–1552.

[Dob+09] Alexandre Dobbertin et al. "Targeting of acetylcholinesterase in neurons in vivo: a dual processing function for the proline-rich membrane anchor subunit and the attachment domain on the catalytic subunit". In: *Journal of Neuroscience* 29.14 (2009), pp. 4519–4530.

[DM16] Ian L Dryden and Kanti V Mardia. *Statistical shape analysis: with applications in R*. John Wiley & Sons, 2016.

[Dub+18] Bruno Dubois et al. "Cognitive and neuroimaging features and brain β-amyloidosis in individuals at risk of Alzheimer's disease (INSIGHT-preAD): a longitudinal observational study". In: *The Lancet Neurology* 17.4 (2018), pp. 335–346.

[DAJ13] Stanley Durrleman, Stéphanie Allassonnière, and Sarang Joshi. "Sparse adaptive parameterization of variability in image ensembles". In: *International Journal of Computer Vision* 101 (2013), pp. 161–183.

[Dut+14] M. Dutschmann et al. "The physiological significance of postinspiration in respiratory control". In: *Progress in brain research* 212 (2014), pp. 113–130.

[Duy+01] Ellen G Duysen et al. "Evidence for nonacetylcholinesterase targets of organophosphorus nerve agent: supersensitivity of acetylcholinesterase knockout mouse to VX lethality". In: *Journal of Pharmacology and Experimental Therapeutics* 299.2 (2001), pp. 528–535.

[DRG15] Gintare Karolina Dziugaite, Daniel M Roy, and Zoubin Ghahramani. "Training generative neural networks via maximum mean discrepancy optimization". In: *arXiv preprint arXiv:1505.03906* (2015).

[EH+08] Herbert Edelsbrunner, John Harer, et al. "Persistent homology-a survey". In: *Contemporary mathematics* 453.26 (2008), pp. 257–282.

[EB16] SE Roian Egnor and Kristin Branson. "Computational analysis of behavior". In: *Annual review of neuroscience* 39.1 (2016), pp. 217–236.

[Elh+13] Ehsan Elhamifar et al. "A convex optimization framework for active learning". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 209–216.

[EA12a] P. Esling and C. Agon. "Time-series data mining". In: *ACM Computing Surveys* 45 (1 2012), pp. 1–34. DOI: 10.1145/2379776.2379788Ãŕ. URL: https://hal.archives-ouvertes.fr/hal-01577883.

[EA12b] Philippe Esling and Carlos Agon. "Time-series data mining". In: *ACM Computing Surveys (CSUR)* 45.1 (2012), pp. 1–34.

[FRM94] Christos Faloutsos, Mudumbai Ranganathan, and Yannis Manolopoulos. "Fast subsequence matching in time-series databases". In: *ACM Sigmod Record* 23.2 (1994), pp. 419–429.

[Fen+99] Guoping Feng et al. "Genetic analysis of collagen Q: roles in acetylcholinesterase and butyrylcholinesterase assembly and in synaptic structure and function". In: *The Journal of cell biology* 144.6 (1999), pp. 1349–1360.

[FN19] Tiantian Feng and Shrikanth S Narayanan. "Discovering optimal variable-length time series motifs in large-scale wearable recordings of human bio-behavioral signals". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 7615–7619.

[Fer17] Tamás Ferenti. "Biomedical applications of time series analysis". In: *2017 ieee 30th neumann colloquium (nc)*. IEEE. 2017, pp. 000083–000084.

[Fey20] Jean Feydy. "Geometric data analysis, beyond convolutions". In: *Applied Mathematics* 3 (2020).

[Fu11] Tak-chung Fu. "A review on time series data mining". In: *Engineering Applications of Artificial Intelligence* 24.1 (2011), pp. 164–181.

[GL17] Yifeng Gao and Jessica Lin. "Efficient discovery of time series motifs with large length range in million scale time series". In: *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2017, pp. 1213–1222.

[GL19] Yifeng Gao and Jessica Lin. "HIME: discovering variable-length motifs in large-scale time series". In: *Knowledge and Information Systems* 61 (2019), pp. 513–542.

[Gas+22] Christian Gaser et al. "CAT–A computational anatomy toolbox for the analysis of structural MRI data". In: *biorxiv* (2022), pp. 2022–06.

[GBA21] Fleur Gaudfernau, Eléonore Blondiaux, and Stéphanie Allassonière. "Analysis of the anatomical variability of Fetal brains with Corpus callosum agenesis". In: *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Perinatal Imaging, Placental and Preterm Image Analysis: 3rd International Workshop, UNSURE 2021, and 6th International Workshop, PIPPI 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, October 1, 2021, Proceedings 3*. Springer. 2021, pp. 274–283.

[GTO24a] Thibaut Germain, Charles Truong, and Laurent Oudre. "Interactive motif discovery in time series with persistent homology". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2024, pp. 383–387.

[GTO24b] Thibaut Germain, Charles Truong, and Laurent Oudre. "Linear-trend normalization for multivariate subsequence similarity search". In: *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*. IEEE. 2024, pp. 167–175.

[GTO24c] Thibaut Germain, Charles Truong, and Laurent Oudre. "Persistence-based motif discovery in time series". In: *IEEE Transactions on Knowledge and Data Engineering* (2024).

[Ger+22] Thibaut Germain et al. "Unsupervised study of plethysmography signals through DTW clustering". In: *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE. 2022, pp. 3396–3400.

[Ger+23] Thibaut Germain et al. "Unsupervised classification of plethysmography signals with advanced visual representations". In: *Frontiers in Physiology* 14 (2023), p. 781.

[Ger+24] Thibaut Germain et al. "Shape analysis for time series". In: *Advances in neural information processing systems* (2024).

[GB21] T. Glaab and A. Braun. "Noninvasive measurement of pulmonary function in experimental mouse models of airway disease". In: *Lung* 199.3 (2021), pp. 255–261.

[Gla05] Joan Glaunes. "Transport par difféomorphismes de points, de mesures et de courants pour la comparaison de formes et l'anatomie numérique". In: *These de sciences, Université Paris* 13 (2005).

[Gla+08] Joan Glaunes et al. "Large deformation diffeomorphic metric curve mapping". In: *International journal of computer vision* 80 (2008), pp. 317–336.

[God+21] Rakshitha Godahewa et al. "Monash time series forecasting archive". In: *arXiv preprint arXiv:2105.06643* (2021).

[Gol+00] Ary L Goldberger et al. "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals". In: *circulation* 101.23 (2000), e215–e220.

[GK95] Dina Q Goldin and Paris C Kanellakis. "On similarity queries for time-series data: constraint specification and implementation". In: *International conference on principles and practice of constraint programming*. Springer. 1995, pp. 137–153.

[GZ15] Koosha Golmohammadi and Osmar R Zaiane. "Time series contextual anomaly detection for detecting market manipulation in stock market". In: *2015 IEEE international conference on data science and advanced analytics (DSAA)*. IEEE. 2015, pp. 1–10.

[Gom+14] Alex Gomez-Marin et al. "Big behavioral data: psychology, ethology and the foundations of neuroscience". In: *Nature neuroscience* 17.11 (2014), pp. 1455–1462.

[GCS20] Tiago Silveira Gontijo, Marcelo Azevedo Costa, and Rodrigo Barbosa de Santis. "Similarity search in electricity prices: An ultra-fast method for finding analogs". In: *Journal of Renewable and Sustainable Energy* 12.5 (2020).

[GSS16] Josif Grabocka, Nicolas Schilling, and Lars Schmidt-Thieme. "Latent time-series motifs". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11.1 (2016), pp. 1–20.

[GPM90] Scott David Greenwald, Ramesh S Patil, and Roger G Mark. *Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information*. IEEE, 1990.

[Gre94] Ulf Grenander. *General pattern theory: A mathematical study of regular structures*. Oxford University Press, 1994.

[GM98] Ulf Grenander and Michael I Miller. "Computational anatomy: An emerging discipline". In: *Quarterly of applied mathematics* 56.4 (1998), pp. 617–694.

[Gua+24] Debao Guan et al. "Using LDDMM and a kinematic cardiac growth model to quantify growth and remodelling in rat hearts under PAH". In: *Computers in Biology and Medicine* 171 (2024), p. 108218.

[GKK20] Harshit Gujral, Ajay Kumar Kushwaha, and Sukant Khurana. "Utilization of time series tools in life-sciences and neuroscience". In: *Neuroscience Insights* 15 (2020), p. 2633105520963045.

[Hel+05] Patrick Helm et al. "Measuring and mapping cardiac fiber and laminar architecture using diffusion tensor MR imaging". In: *Annals of the New York Academy of Sciences* 1047.1 (2005), pp. 296–307.

[Heo+24] Tae-Young Heo et al. "Logistic regression models for elastic shape of curves based on tangent representations". In: *Journal of the Korean Statistical Society* (2024), pp. 1–19.

[HW21] Matthieu Herrmann and Geoffrey I Webb. "Early abandoning and pruning for elastic distances including dynamic time warping". In: *Data Mining and Knowledge Discovery* 35.6 (2021), pp. 2577–2601.

[Hip+19] Michael Hippke et al. "Wōtan: Comprehensive time-series detrending in Python". In: *The Astronomical Journal* 158.4 (2019), p. 143.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[HMB24] Christopher Holder, Matthew Middlehurst, and Anthony Bagnall. "A review and evaluation of elastic distance functions for time series clustering". In: *Knowledge and Information Systems* 66.2 (2024), pp. 765–809.

[Hoy12] H. Hoymann. "Lung function measurements in rodents in safety pharmacology studies". In: *Frontiers in Pharmacology* 3 (2012), p. 156. ISSN: 1663-9812. DOI: 10.3389/fphar.2012.00156. URL: https://www.frontiersin.org/article/10.3389/fphar.2012.00156.

[HC62] John R Hurley and Raymond B Cattell. "The Procrustes program: Producing direct rotation to test a hypothesized factor structure". In: *Behavioral science* 7.2 (1962), p. 258.

[JPJ24] Aryan Jadon, Avinash Patil, and Shruti Jadon. "A Comprehensive Survey of Regression-Based Loss Functions for Time Series Forecasting". In: *International Conference on Data Management, Analytics & Innovation*. Springer. 2024, pp. 117–147.

[Jai19] B. Jain. "Revisiting inaccuracies of time series averaging under dynamic time warping". In: *Pattern Recognition Letters* 125 (2019), pp. 418–424.

[JB20] Sara Jamal and Joshua S Bloom. "On neural architectures for astronomical time-series classification with application to variable stars". In: *The Astrophysical Journal Supplement Series* 250.2 (2020), p. 30.

[JCG20] Hicham Janati, Marco Cuturi, and Alexandre Gramfort. "Spatio-temporal alignments: Optimal transport through space and time". In: *International conference on artificial intelligence and statistics*. PMLR. 2020, pp. 1695–1704.

[JB19] Junteng Jia and Austin R Benson. "Neural jump stochastic differential equations". In: *Advances in Neural Information Processing Systems* 32 (2019).

[KCC17] Irene Kaltenmark, Benjamin Charlier, and Nicolas Charon. "A general framework for curve and surface comparison and registration with oriented varifolds". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3346–3355.

[KHP11] Kittipat Kampa, Erion Hasanbelliu, and Jose C Principe. "Closed-form Cauchy-Schwarz PDF divergence for mixture of Gaussians". In: *The 2011 International Joint Conference on Neural Networks*. IEEE. 2011, pp. 2578–2585.

[KR09] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. Vol. 344. John Wiley & Sons, 2009.

[KR05] Eamonn Keogh and Chotirat Ann Ratanamahatana. "Exact indexing of dynamic time warping". In: *Knowledge and information systems* 7 (2005), pp. 358–386.

[Keo+01] Eamonn Keogh et al. "Dimensionality reduction for fast similarity search in large time series databases". In: *Knowledge and information Systems* 3 (2001), pp. 263–286.

[KG22] Samira Khodabandehlou and Seyyed Alireza Hashemi Golpayegani. "Market manipulation detection: A systematic literature review". In: *Expert Systems with Applications* 210 (2022), p. 118330.

[Kid+20] Patrick Kidger et al. "Neural controlled differential equations for irregular time series". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6696–6707.

[KPC01] Sang-Wook Kim, Sanghyun Park, and Wesley W Chu. "An index-based approach for similarity search supporting time warping in large sequence databases". In: *Proceedings 17th international conference on data engineering.* IEEE. 2001, pp. 607–614.

[Kli15] Christian Peter Klingenberg. "Analyzing fluctuating asymmetry with geometric morphometrics: concepts, methods, and applications". In: *Symmetry* 7.2 (2015), pp. 843–934.

[KF09] Bryan Kolb and Bryan D Fantie. "Development of the child's brain and behavior". In: *Handbook of clinical child neuropsychology* (2009), pp. 19–46.

[Kra+11] Gabriela Krasteva et al. "Cholinergic chemosensory cells in the trachea regulate breathing". In: *Proceedings of the National Academy of Sciences* 108.23 (2011), pp. 9478–9483.

[Kuh55] Harold W Kuhn. "The Hungarian method for the assignment problem". In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[KG20] Punit Kumar and Atul Gupta. "Active learning query strategies for classification, regression, and clustering: A survey". In: *Journal of Computer Science and Technology* 35 (2020), pp. 913–945.

[Lau+22] Jessy Lauer et al. "Multi-animal pose estimation, identification and tracking with DeepLabCut". In: *Nature Methods* 19.4 (2022), pp. 496–504.

[LH20] Mathias Lechner and Ramin Hasani. "Learning long-term dependencies in irregularly-sampled time series". In: *arXiv preprint arXiv:2006.04418* (2020).

[Lee+18] Nung Kion Lee et al. "DeepFinder: An integration of feature-based and deep learning approach for DNA motif discovery". In: *Biotechnology & Biotechnological Equipment* 32.3 (2018), pp. 759–768.

[Lee+22] Yurim Lee et al. "Multi-view integrative attention-based deep representation learning for irregular clinical time-series data". In: *IEEE Journal of Biomedical and Health Informatics* 26.8 (2022), pp. 4270–4280.

[Li+17] Jundong Li et al. "Feature selection: A data perspective". In: *ACM computing surveys (CSUR)* 50.6 (2017), pp. 1–45.

[LZ21] Bryan Lim and Stefan Zohren. "Time-series forecasting with deep learning: a survey". In: *Philosophical Transactions of the Royal Society A* 379.2194 (2021), p. 20200209.

[Lin+02] Jessica Lin et al. "Finding motifs in time series". In: *Proceedings of the 2nd Workshop on Temporal Data Mining, at the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 2002, pp. 53–68.

[Lin+07] Jessica Lin et al. "Experiencing SAX: a novel symbolic representation of time series". In: *Data Mining and knowledge discovery* 15 (2007), pp. 107–144.

[Lin+12] Jessica Lin et al. "Pattern recognition in time series". In: *Advances in machine learning and data mining for astronomy* 1.617-645 (2012), p. 3.

[Lin+18] Michele Linardi et al. "Matrix profile X: VALMOD-scalable discovery of variable-length motifs in data series". In: *Proceedings of the 2018 International Conference on Management of Data*. 2018, pp. 1053–1066.

[Liu+15] Bo Liu et al. "Efficient motif discovery for large-scale time series in healthcare". In: *IEEE Transactions on Industrial Informatics* 11.3 (2015), pp. 583–590.

[LR13] Xiao-Ying Liu and Chuan-Lun Ren. "Fast subsequence matching under time warping in time-series databases". In: *2013 International Conference on Machine Learning and Cybernetics*. Vol. 4. IEEE. 2013, pp. 1584–1590.

[Liu+19] Xuanqing Liu et al. "Neural sde: Stabilizing neural ode networks with stochastic noise". In: *arXiv preprint arXiv:1906.02355* (2019).

[Loc15] Oksana Lockridge. "Review of human butyrylcholinesterase structure, function, genetic variants, history of use in the clinic, and potential therapeutic uses". In: *Pharmacology & therapeutics* 148 (2015), pp. 34–46.

[Lou+17] Maxime Louis et al. "Parallel transport in shape analysis: a scalable numerical scheme". In: *Geometric Science of Information: Third International Conference, GSI 2017, Paris, France, November 7-9, 2017, Proceedings 3*. Springer. 2017, pp. 29–37.

[Mag20] Magnus S Magnusson. "T-pattern detection and analysis (TPA) with THEMETM: a mixed methods approach". In: *Frontiers in Psychology* 10 (2020), p. 2663.

[Mai+18] S. Mailhot-Larouche et al. "Assessment of respiratory function in conscious mice by double-chamber plethysmography". In: *Journal of visualized experiments: JoVE* 137 (2018).

[Mal12] Stéphane Mallat. "Group invariant scattering". In: *Communications on Pure and Applied Mathematics* 65.10 (2012), pp. 1331–1398.

[Man+11] Tommaso Mansi et al. "A statistical model for quantification and prediction of cardiac remodelling: Application to tetralogy of fallot". In: *IEEE transactions on medical imaging* 30.9 (2011), pp. 1605–1616.

[Mar+17] Lucie Maršánová et al. "ECG features and methods for automatic classification of ventricular premature and ischemic heartbeats: A comprehensive experimental study". In: *Scientific reports* 7.1 (2017), p. 11239.

[Mat+18] Alexander Mathis et al. "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning". In: *Nature neuroscience* 21.9 (2018), pp. 1281–1289.

[MM20] Mackenzie Weygandt Mathis and Alexander Mathis. "Deep learning tools for the measurement of animal behavior in neuroscience". In: *Current opinion in neurobiology* 60 (2020), pp. 1–11.

[MJ99] Larry Medsker and Lakhmi C Jain. *Recurrent neural networks: design and applications*. CRC press, 1999.

[Meh+22] Philip Mehrgardt et al. "Pulse Transit Time PPG Dataset". In: *PhysioNet* 10 (2022), e215–e220.

[MQ09] Michael I Miller and Anqi Qiu. "The emerging discipline of computational functional anatomy". In: *Neuroimage* 45.1 (2009), S16–S39.

[MTY02] Michael I Miller, Alain Trouvé, and Laurent Younes. "On the metrics and Euler-Lagrange equations of computational anatomy". In: *Annual review of biomedical engineering* 4.1 (2002), pp. 375–405.

[MTY06] Michael I Miller, Alain Trouvé, and Laurent Younes. "Geodesic shooting for computational anatomy". In: *Journal of mathematical imaging and vision* 24 (2006), pp. 209–228.

[Min+02] Jasmina Minic et al. "Regulation of acetylcholine release by muscarinic receptors at the mouse neuromuscular junction depends on the activity of acetylcholinesterase". In: *European Journal of Neuroscience* 15.3 (2002), pp. 439–448.

[Min+07] David Minnen et al. "Detecting subdimensional motifs: An efficient algorithm for generalized multivariate pattern discovery". In: *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. IEEE. 2007, pp. 601–606.

[MLA23] Jahangir Moini, Anthony LoGalbo, and Raheleh Ahangari. *Foundations of the Mind, Brain, and Behavioral Relationships: Understanding Physiological Psychology*. Elsevier, 2023.

[MAM23] Tanmoy Mondal, Reza Akbarinia, and Florent Masseglia. "kNN matrix profile for knowledge discovery from time series". In: *Data Mining and Knowledge Discovery* 37.3 (2023), pp. 1055–1089.

[MM01] George B Moody and Roger G Mark. "The impact of the MIT-BIH arrhythmia database". In: *IEEE engineering in medicine and biology magazine* 20.3 (2001), pp. 45–50.

[Mor+18] M. Morel et al. "Time-series averaging using constrained dynamic time warping with tolerance". In: *Pattern Recognition* 74 (2018), pp. 77–89.

[Mor+08] Susumu Mori et al. "Stereotaxic white matter atlas based on diffusion tensor imaging in an ICBM template". In: *Neuroimage* 40.2 (2008), pp. 570–582.

[Mos+23] Eduardo Mosqueira-Rey et al. "Human-in-the-loop machine learning: a state of the art". In: *Artificial Intelligence Review* 56.4 (2023), pp. 3005–3054.

[Mue+09] Abdullah Mueen et al. "Exact discovery of time series motifs". In: *Proceedings of the 2009 SIAM international conference on data mining*. SIAM. 2009, pp. 473–484.

[Mur02] D. J. Murphy. "Assessment of respiratory function in safety pharmacology". In: *Fundamental & clinical pharmacology* 16.3 (2002), pp. 183–196.

[MSS17] David Murray, Lina Stankovic, and Vladimir Stankovic. "An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study". In: *Scientific data* 4.1 (2017), pp. 1–12.

[Ner+19] A. Nervo et al. "Respiratory failure triggered by cholinesterase inhibitors may involve activation of a reflex sensory pathway by acetylcholine spillover". In: *Toxicology* 424 (2019), p. 152232.

[Ner18] Aurélie Nervo. "Physiopathologie de la ventilation après modulation génétique et pharmacologique du système cholinergique: mise en place d'un modèle d'analyse de la ventilation par système de pléthysmographie double chambre". PhD thesis. Université Sorbonne Paris Cité, 2018.

[NW97] Craig G Nevill-Manning and Ian H Witten. "Identifying hierarchical structure in sequences: A linear-time algorithm". In: *Journal of Artificial Intelligence Research* 7 (1997), pp. 67–82.

[NR07] V. Niennattrakul and C. A. Ratanamahatana. "Inaccuracies of shape averaging method using dynamic time warping for time series data". In: *Proceedings of the International conference on computational science*. Springer. 2007, pp. 513–520.

[OLK24] YongKyung Oh, Dongyoung Lim, and Sungil Kim. "Stable Neural Stochastic Differential Equations in Analyzing Irregular Time Series Data". In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=4VIgNuQ1pY.

[Ots79] N. Otsu. "A threshold selection method from gray-level histograms". In: *IEEE transactions on systems, man, and cybernetics* 9.1 (1979), pp. 62–66.

[Pai+23] François Painblanc et al. "Match-and-deform: Time series domain adaptation through optimal transport and temporal alignment". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2023, pp. 341–356.

[Pal20] Themis Palpanas. "Evolution of a Data Series Index: The iSAX Family of Data Series Indexes: iSAX, iSAX2. 0, iSAX2+, ADS, ADS+, ADS-Full, ParIS, ParIS+, MESSI, DPiSAX, ULISSE, Coconut-Trie/Tree, Coconut-LSM". In: *Information Search, Integration, and Personalization: 13th International Workshop, ISIP 2019, Heraklion, Greece, May 9–10, 2019, Revised Selected Papers 13*. Springer. 2020, pp. 68–83.

[Pap+20] John Paparrizos et al. "Debunking four long-standing misconceptions of time-series distance measures". In: *Proceedings of the 2020 ACM SIGMOD international conference on management of data*. 2020, pp. 1887–1905.

[Pap+22a] John Paparrizos et al. "TSB-UAD: an end-to-end benchmark suite for univariate time-series anomaly detection". In: *Proceedings of the VLDB Endowment* 15.8 (2022), pp. 1697–1711.

[Pap+22b] John Paparrizos et al. "Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection". In: *Proceedings of the VLDB Endowment* 15.11 (2022), pp. 2774–2787.

[Pap+23] John Paparrizos et al. "Accelerating similarity search for elastic measures: A study and new generalization of lower bounding distances". In: *Proceedings of the VLDB Endowment* 16.8 (2023), pp. 2019–2032.

[Pat+02] Pranav Patel et al. “Mining motifs in massive time series databases”. In: *2002 IEEE International Conference on Data Mining, 2002. Proceedings.* IEEE. 2002, pp. 370–377.

[PKG11] F. Petitjean, A. Ketterlin, and P. Gançarski. “A global averaging method for dynamic time warping, with applications to clustering”. In: *Pattern recognition* 44.3 (2011), pp. 678–693.

[PPK21] Konstantin A Petrov, Svetlana E Proskurina, and Eric Krejci. “Cholinesterases in tripartite neuromuscular synapse”. In: *Frontiers in Molecular Neuroscience* 14 (2021), p. 811220.

[Pir+21] Paolo Piras et al. “Transporting deformations of face emotions in the shape spaces: A comparison of different approaches”. In: *Journal of Mathematical Imaging and Vision* 63.7 (2021), pp. 875–893.

[PLX22] Chi Seng Pun, Si Xian Lee, and Kelin Xia. “Persistent-homology-based machine learning: a survey and a comparative study”. In: *Artificial Intelligence Review* 55.7 (2022), pp. 5169–5213.

[Qiu+09] Anqi Qiu et al. “Time sequence diffeomorphic metric mapping and parallel transport track time-dependent shape changes”. In: *NeuroImage* 45.1 (2009), S51–S60.

[RBP11] Julien Rabatel, Sandra Bringay, and Pascal Poncelet. “Anomaly detection in monitoring sensor data for preventive maintenance”. In: *Expert Systems with Applications* 38.6 (2011), pp. 7003–7015.

[RLM21] A Raiyani, A Lathigara, and H Mehta. “Usage of time series forecasting model in Supply chain sales prediction”. In: *IOP Conference Series: Materials Science and Engineering*. Vol. 1042. 1. IOP Publishing. 2021, p. 012022.

[Rak+12a] Thanawin Rakthanmanon et al. “MDL-based time series clustering”. In: *Knowledge and information systems* 33 (2012), pp. 371–399.

[Rak+12b] Thanawin Rakthanmanon et al. “Searching and mining trillions of time series subsequences under dynamic time warping”. In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2012, pp. 262–270.

[Sac+22] Lucie Saclova et al. “Reliable P wave detection in pathological ECG signals”. In: *Scientific Reports* 12.1 (2022), p. 6589.

[SC78] Hiroaki Sakoe and Seibi Chiba. “Dynamic programming algorithm optimization for spoken word recognition”. In: *IEEE transactions on acoustics, speech, and signal processing* 26.1 (1978), pp. 43–49.

[Sal+10] Albert Ali Salah et al. “T-patterns revisited: mining for temporal patterns in sensor data”. In: *Sensors* 10.8 (2010), pp. 7496–7513.

[Sar+21] Saquib Sarfraz et al. “Temporally-weighted hierarchical clustering for unsupervised action segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11225–11234.

[SBH74] B Mca Savers, HA Beagley, and WR Henshall. "The mechanism of auditory evoked EEG responses". In: *Nature* 247.5441 (1974), pp. 481–483.

[SL22] Patrick Schäfer and Ulf Leser. "Motiflets: Simple and Accurate Detection of Motifs in Time Series". In: *Proceedings of the VLDB Endowment* 16.4 (2022), pp. 725–737.

[SWP22] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. "Anomaly detection in time series: a comprehensive evaluation". In: *Proceedings of the VLDB Endowment* 15.9 (2022), pp. 1779–1797.

[SJ18] D. Schultz and B. Jain. "Nonsmooth analysis and subgradient methods for averaging in dynamic time warping spaces". In: *Pattern Recognition* 74 (2018), pp. 340–358.

[SP97] Mike Schuster and Kuldip K Paliwal. "Bidirectional recurrent neural networks". In: *IEEE transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

[Sco+24] Ben A Scott et al. "Matrix Profile data mining for BGP anomaly detection". In: *Computer Networks* 242 (2024), p. 110257.

[Sen+14] Pavel Senin et al. "Grammarviz 2.0: a tool for grammar-based pattern discovery in time series". In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part III 14*. Springer. 2014, pp. 468–472.

[Sen+18] Pavel Senin et al. "Grammarviz 3.0: Interactive discovery of variable-length time series patterns". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 12.1 (2018), pp. 1–28.

[SMR13] Huijuan Shao, Manish Marwah, and Naren Ramakrishnan. "A temporal motif mining approach to unsupervised energy disaggregation: Applications to residential and commercial buildings". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 27. 1. 2013, pp. 1327–1333.

[SM21] Satya Narayan Shukla and Benjamin M Marlin. "Multi-time attention networks for irregularly sampled time series". In: *arXiv preprint arXiv:2101.10318* (2021).

[Sil+18] Diego F Silva et al. "Fast similarity matrix profile for music analysis and exploration". In: *IEEE Transactions on Multimedia* 21.1 (2018), pp. 29–38.

[Sin+20] Amit Singhal et al. "An efficient removal of power-line interference and baseline wander from ECG signals by employing Fourier decomposition technique". In: *Biomedical Signal Processing and Control* 57 (2020), p. 101741.

[SR+15] Haemwaan Sivaraks, Chotirat Ann Ratanamahatana, et al. "Robust and accurate anomaly detection in ECG artifacts using time series motif discovery". In: *Computational and mathematical methods in medicine* 2015 (2015).

[Sla15] Clarke R Slater. "The functional organization of motor nerve terminals". In: *Progress in neurobiology* 134 (2015), pp. 55–103.

[SR24] Sondre Sørbø and Massimiliano Ruocco. "Navigating the metric maze: A taxonomy of evaluation metrics for anomaly detection in time series". In: *Data Mining and Knowledge Discovery* 38.3 (2024), pp. 1027–1068.

[Sri+10] Anuj Srivastava et al. "Shape analysis of elastic curves in euclidean spaces". In: *IEEE transactions on pattern analysis and machine intelligence* 33.7 (2010), pp. 1415–1428.

[Sto+24] Kaitlin M Stouffer et al. "Cross-modality mapping using image varifolds to align tissue-scale atlases to molecular-scale measures with application to 2D brain sections". In: *Nature Communications* 15.1 (2024), p. 3530.

[SF21] M. D. Sunshine and D. D. Fuller. "Automated Classification of Whole Body Plethysmography Waveforms to Quantify Breathing Patterns". In: *Frontiers in Physiology* (2021), p. 1347.

[TR87] John Tafuri and James Roberts. "Organophosphate poisoning". In: *Annals of emergency medicine* 16.2 (1987), pp. 193–202.

[TPW19] Chang Wei Tan, François Petitjean, and Geoffrey I Webb. "Elastic bands across the path: A new framework and method to lower bound DTW". In: *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM. 2019, pp. 522–530.

[Tat+18] Nesime Tatbul et al. "Precision and recall for time series". In: *Advances in neural information processing systems* 31 (2018).

[TSP08] Romain Tavenard, Albert A Salah, and Eric J Pauwels. "Searching for temporal patterns in ami sensor data". In: *Constructing Ambient Intelligence: AmI 2007 Workshops Darmstadt, Germany, November 7-10, 2007 Revised Papers*. Springer. 2008, pp. 53–62.

[Tho17] J Arthur Thomson. *On growth and form*. 1917.

[TL17] Sahar Torkamani and Volker Lohweg. "Survey on time series motif discovery". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7.2 (2017), e1199.

[TOV20] Charles Truong, Laurent Oudre, and Nicolas Vayatis. "Selective review of offline change point detection methods". In: *Signal Processing* 167 (2020), p. 107299.

[TR19] Belinda Tzen and Maxim Raginsky. "Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit". In: *arXiv preprint arXiv:1905.09883* (2019).

[Vai+04] Marc Vaillant et al. "Statistics on diffeomorphisms via tangent space representations". In: *NeuroImage* 23 (2004), S161–S169.

[Vay+20] Titouan Vayer et al. "Time series alignment with global invariances". In: *arXiv preprint arXiv:2002.03848* (2020).

[Vic+19] Jose Vicente et al. "Assessment of Multi-Ion Channel Block in a Phase I Randomized Study Design: Results of the Ci PA Phase I ECG Biomarker Validation Study". In: *Clinical Pharmacology & Therapeutics* 105.4 (2019), pp. 943–953.

[Vij+93] R. Vijayaraghavan et al. "Characteristic modifications of the breathing pattern of mice to evaluate the effects of airborne chemicals on the respiratory tract". In: *Archives of toxicology* 67.7 (1993), pp. 478–490.

[WM18] Jana Wäldchen and Patrick Mäder. "Plant species identification using computer vision techniques: a systematic literature review". In: *Archives of computational methods in engineering* 25 (2018), pp. 507–543.

[Wan+07] Lei Wang et al. "Large deformation diffeomorphism and momentum based hippocampal shape discrimination in dementia of the Alzheimer type". In: *IEEE transactions on medical imaging* 26.4 (2007), pp. 462–470.

[Wan+13] Xiaoyue Wang et al. "Experimental comparison of representation methods and distance measures for time series data". In: *Data Mining and Knowledge Discovery* 26 (2013), pp. 275–309.

[Wan+24] Zeyu Wang et al. "DumpyOS: A data-adaptive multi-ary index for scalable data series similarity search". In: *The VLDB Journal* (2024), pp. 1–25.

[WJ21] Rutuja Wankhedkar and Sanjay Kumar Jain. "Motif discovery and anomaly detection in an ECG using matrix profile". In: *Progress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2019, Volume 1*. Springer. 2021, pp. 88–95.

[WP21] Geoffrey I Webb and François Petitjean. "Tight lower bounds for dynamic time warping". In: *Pattern Recognition* 115 (2021), p. 107895.

[Wei+24] Caleb Weinreb et al. "Keypoint-MoSeq: parsing behavior by linking point tracking to pose dynamics". In: *Nature Methods* 21.7 (2024), pp. 1329–1339.

[Wei18] Ben G Weinstein. "A computer vision for animal ecology". In: *Journal of Animal Ecology* 87.3 (2018), pp. 533–545.

[Wen+20] Junhao Wen et al. "Convolutional neural networks for classification of Alzheimer's disease: Overview and reproducible evaluation". In: *Medical image analysis* 63 (2020), p. 101694.

[Wil+17] R. Willmann et al. "Improving reproducibility of phenotypic assessments in the dyw mouse model of laminin-α2 related congenital muscular dystrophy". In: *Journal of neuromuscular diseases* 4.2 (2017), pp. 115–126.

[Wil17] Seunghye J Wilson. "Data representation for time series data mining: time domain approaches". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 9.1 (2017), e1392.

[Wil+20] Alexander B Wiltschko et al. "Revealing the structure of pharmacobehavioral space through motion sequencing". In: *Nature neuroscience* 23.11 (2020), pp. 1433–1443.

[WHS24] Yuexuan Wu, Chao Huang, and Anuj Srivastava. "Shape-based functional data analysis". In: *TEST* 33.1 (2024), pp. 1–47.

[Ye+24] Shaokai Ye et al. "SuperAnimal pretrained pose estimation models for behavioral analysis". In: *Nature Communications* 15.1 (2024), p. 5165.

[YKK17] Chin-Chia Michael Yeh, Nickolas Kavantzas, and Eamonn Keogh. "Matrix profile VI: Meaningful multidimensional motif discovery". In: *2017 IEEE international conference on data mining (ICDM)*. IEEE. 2017, pp. 565–574.

[Yeh+16] Chin-Chia Michael Yeh et al. "Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets". In: *2016 IEEE 16th international conference on data mining (ICDM)*. Ieee. 2016, pp. 1317–1322.

[You10] Laurent Younes. *Shapes and diffeomorphisms*. Vol. 171. Springer, 2010.

[You12] Laurent Younes. "Spaces and manifolds of shapes in computer vision: An overview". In: *Image and Vision Computing* 30.6-7 (2012), pp. 389–397.

[ZK16] Sergey Zagoruyko and Nikos Komodakis. "Wide residual networks". In: *arXiv preprint arXiv:1605.07146* (2016).

[ZGC20] Yadong Zhang, Fuhang Gan, and Xin Chen. "Motif difference field: An effective image-based time series classification and applications in machine malfunction detection". In: *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)*. IEEE. 2020, pp. 3079–3083.

[ZI18] Jiaping Zhao and Laurent Itti. "shapeDTW: Shape dynamic time warping". In: *Pattern Recognition* 74 (2018), pp. 171–184.

[Zhe+23] Ce Zheng et al. "Deep learning-based human pose estimation: A survey". In: *ACM Computing Surveys* 56.1 (2023), pp. 1–37.

[ZM24] Sheng Zhong and Abdullah Mueen. "MASS: distance profile of a query over a time series". In: *Data Mining and Knowledge Discovery* (2024), pp. 1–27.

[ZMK19] Yan Zhu, Abdullah Mueen, and Eamonn Keogh. "Matrix profile IX: Admissible time series motif discovery with missing data". In: *IEEE Transactions on Knowledge and Data Engineering* 33.6 (2019), pp. 2616–2626.

[Zhu+16] Yan Zhu et al. "Matrix profile ii: Exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins". In: *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE. 2016, pp. 739–748.

[Zhu+18] Yan Zhu et al. "Matrix profile XI: SCRIMP++: time series motif discovery at interactive speeds". In: *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2018, pp. 837–846.

[Zhu+20] Juntang Zhuang et al. "Adabelief optimizer: Adapting stepsizes by the belief in observed gradients". In: *Advances in neural information processing systems* 33 (2020), pp. 18795–18806.

[Zin+24] Ekaterina Zinkovskaia et al. "Temporally aligned segmentation and clustering (TASC) framework for behavior time series analysis". In: *Scientific Reports* 14.1 (2024), p. 14952.