Manuscrit de thèse*

Mathieu Carrière

9 avril 2018

1 Analyse de donnée

La génération et l'accumulation de données dans des secteurs d'activités variés, autant industriels qu'académiques, ont pris beaucoup d'importance au cours des dernières années, et sont maintenant omniprésents dans de nombreux domaines scientifiques, financiers et industriels. A titre d'exemple, en science du numérique, le développement rapide des processus d'acquisition et de traitement d'images ont permis la mise à disposition publique en ligne d'importantes bases de données [32, 30, 38, 40, 44]. De la même manière, en biologie, la nouvelle génération de séquenceurs ont permis à la plupart des laboratoires d'aisément déterminer l'ADN de différents organismes [7, 25, 29, 36]. Ainsi, la synthétisation et l'extraction d'informations utiles à partir de ces bases de données massives sont devenus des problèmes d'intérêt majeur.

L'apprentissage automatique est un domaine de la science des données dont le but est de fournir des algorithmes ("automatique") pouvant réaliser des prédictions sur de nouvelles données à partir seulement de l'information déjà présente dans des données préalablement collectées ("apprentissage"). Ces techniques permettent de répondre à de multiples problèmes de l'analyse de données, tels que la *classification*, où l'on cherche à prédire des labels, le *clustering*, où l'on cherche à regrouper les données en différents groupes, ou la *régression*, où l'on cherche à approcher une fonction à partir de sa valeur sur les points de données. Nous orientons le lecteur désireux de trouver plus de détails vers [23] pour une introduction complète de ces problématiques. Par exemple, un problème typique de classification est la prédiction de la présence ou non d'effets d'un médicament sur un patient P. Il s'agit d'un

*Ce résumé est repris du premier chapitre du manuscrit de thèse, qui résume déjà les principaux challenges ainsi que les solutions proposées.

problème de classification binaire en cela que les labels à prédire sont au nombre de deux, à savoir "effet" ou "sans effet". En supposant qu'une base de données est disponible, dans laquelle sont enregistrés les effets ou non du médicament sur plusieurs patients, une des manières les plus simples de procéder est de chercher le patient le plus proche de P dans la base de données, et d'attribuer à P le label de ce patient. Cette méthode, simple quoique très efficace, s'appelle la prédiction par le plus proche voisin, et a déjà été étudiée en détail. Plus généralement, la prédiction par le plus proche voisin n'est qu'une méthode parmi de nombreuses autres en apprentissage automatique, qui peuvent traiter de problèmes aussi variés que la classification d'images, la prédiction du genre musical ou le diagnostic médical, pour ne citer que quelques exemples. D'autres exemples d'applications sont présentés dans [23].

Descripteurs. En général, les données prennent la forme de nuage de points dans $\mathbb{R}^D$, où $D \in \mathbb{N}^*$. Chaque point de donnée représente une *observation*, et chaque dimension, ou coordonnée, représente une *mesure*. Par exemple, les observations peuvent être des patients, des images ou des séquences d'ADN, dont les mesures correspondantes seraient des caractéristiques physiques (la taille, le poids, l'âge...), le niveau de gris des pixels, ou des bases azotées A, C, T ou G composant l'ADN. Très souvent, le nombre de mesures est élevé, fournissant ainsi beaucoup d'informations, mais rendant dans le même temps les données impossibles à visualiser.

Ainsi, une grande partie de l'analyse de données se consacre à la synthétisation de l'information contenue dans les données en des *descripteurs* simples et interprétables, qui dépendent en général de l'application. Par exemple, on peut touver, parmi les descripteurs usuels : le modèle sac-de-mots [47] pour les données textuelles, les descripteurs SIFT [33] et HoG [21] pour les images, la courbure et les images de spin [28] pour les formes 3D, les descripteurs en ondelettes [35] pour le traitement du signal, et, plus généralement, le résultat d'une technique de réduction de dimension, comme l'ACP, MDS ou Isomap [48]. L'efficacité des descripteurs est souvent corrélée aux propriétés dont ils bénéficient. En fonction de l'application, il peut être pertinent d'exiger d'un descripteur qu'il soit invariant par translation ou rotation, intrinsèque ou extrinsèque, un vecteur Euclidien, etc. Trouver des descripteurs avec de telles propriétés est une question importante car permettant d'améliorer grandement l'interprétation et la visualisation des données, comme mentionné plus haut, mais aussi le résultat des algorithmes d'apprentissage, qui sont susceptibles de produire de mauvaises performances si alimentés avec des données brutes. Le but de cette thèse est d'étudier une classe spécifique

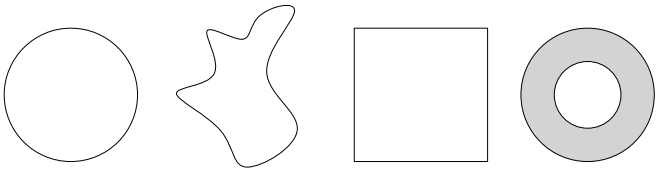

FIGURE 1 – Déformations du cercle.

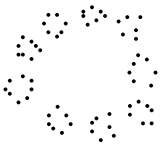

FIGURE 2 – Ce nuage de points semble échantillonné sur neuf cercle à petite échelle, et sur un seul cercle à plus grande échelle.

de descripteurs appelés *topologiques*, et qui sont connus pour être invariants aux déformations continues des données qui n'impliquent pas de déchirement ou de recollement [10].

1.1 Descripteurs topologiques

L'idée derrière les descripteurs topologiques est de synthétiser *l'information topologique* présente dans les données [10]. Intuitivement, la topologie des données englobe toutes les propriétés qui sont préservées par des déformations continues, comme l'étirement, le rétrécissement ou l'épaississement, sans déchirure ni recollement. Par exemple, si un cercle est continument déformé sans déchirement ou recollement, un trou va toujours subsister dans l'objet résultant, quelle qu'ait été la transformation. C'est ce qu'on appelle un *attribut topologique*. Voir la Figure 1, où la présence d'un trou est attestée dans différentes déformations du cercle.

De manière similaire, les composantes connexes, cavités, et trous de dimension supérieure sont des attributs topologiques. Dans l'optique de formaliser la présence de tels attributs (en toute dimension), *la théorie de l'homologie*, a été développée au 19e et au début du 20e siècle. Elle se présente comme un encodage algébrique de l'information topologique. L'homologie d'un espace est une famille de groupes abéliens (un pour chaque dimension), dont les éléments sont des combinaisons linéaires des trous de l'espace.

Cependant, les groupes d'homologie ne sont pas des descripteurs topologiques très performants en tant que tels, la raison principale étant que les données prennent souvent la forme de nuages de points, dont les groupes d'homologie ne sont pas informatifs : chaque point du nuage est un générateur du groupe d'homologie en dimension 0, puisque l'homologie en dimension 0 compte les composantes connexes, et tous les groupes d'homologie de dimension supérieure sont triviaux puisque le nuage n'a aucun trou. Evidemment, le nuage de points peut tout de même refléter de l'information topologique -

par exemple s'il est échantillonné sur un objet géométrique comme un cercle, une sphère ou un tore. La question devient ainsi celle de l'échelle avec laquelle observer les données, comme illustré dans la Figure 2.

L'analyse de données topologiques fournit deux constructions : les diagramme de persistance, qui synthétisent l'information topologique à toutes les échelles, et les Mappers, qui encodent plus d'information géométrique à échelle fixée.

Diagrammes de persistance. Puisque chaque échelle fournit des informations topologiques pertinentes, l'idée de l'homologie persistante est d'encoder l'homologie du nuage de points à toutes les échelles. Considérons la base de données de la Figure 3, contenant des images à 128×128 pixels, vus comme des vecteurs en dimension 16 384, où chaque coordonnée est le niveau de gris d'un pixel. Puisque la caméra a tourné autour de l'objet, il s'ensuit qu'à petite échelle, les données semblent être réparties en petits groupes, tandis qu'à échelle plus grande, elles semblent échantillonnées sur un cercle (plongé dans $\mathbb{R}^{16384}$).

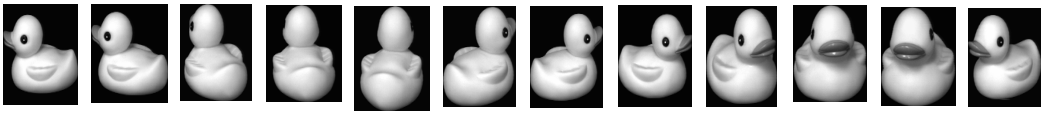

FIGURE 3 – Une base de données d'images.

Pour synthétiser cette information, on peut faire grossir des boules centrées sur les points de données. Considérons trois rayons différents pour ces boules : un petit α, un légèrement plus grand β et un beaucoup plus grand γ, comme montré dans la Figure 4.

Quand le rayon des boules vaut α, l'union des boules est simplement l'union de dix composantes connexes, dont l'homologie en dimension 1 et supérieure est triviale. Cependant, quand le rayon devient β, l'union des boules a l'homologie d'un cercle, dont le trou en dimension 1 devient rempli quand le rayon devient γ. On dit que les composantes connexes sont nées à la valeur α, et neuf sont mortes, c'est-à-dire se sont fait relier à la dixième, à la valeur β. De la même manière, le trou en dimension 1 est apparu au rayon β, et a disparu au rayon γ. Enfin, la dixième composante connexe est apparue au rayon α et a persisté jusqu'au rayon γ. Cette information est encodée dans le *diagramme de persistance*, qui est un multi-ensemble [1] de points, chacun représentant un attribut topologique, et ayant les rayons de naissance et de

1. Un multi-ensemble est une généralisation d'un ensemble, dans laquelle les points ont des multiplicités.

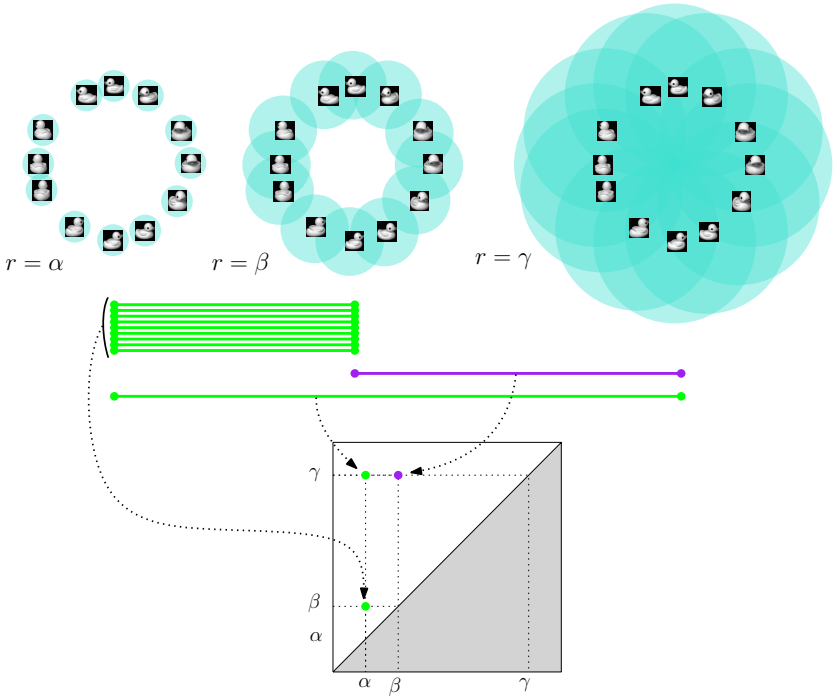

FIGURE 4 – Trois différentes unions de boules centrées sur des images vus comme des vecteurs dans un espace Euclidien de grande dimension. L'apparition et la disparition d'attributs topologiques, comme des composantes connexes ou des trous, est enregistrée dans un *diagramme de persistance*, dans lequel les points représentant des attributs en dimension 0 sont en vert, et ceux représentant des attributs en dimension 1 sont en violet.

mort comme coordonnées. La distance à la diagonale fournit une quantité utile et interprétable dans les diagrammes de persistance. En effet, si un point est loin de la diagonale, alors son ordonnée est largement supérieur à son abscisse, ce qui signifie que l'attribut topologique correspondant était présent dans l'union des boules pour une large gamme de rayons différents, indiquant ainsi que l'attribut topologique a des chances d'être présent dans l'objet sous-jacent, et d'être une information pertinente. Au contraire, les points proches de la diagonale représentent des attributs qui ont disparu rapidement après être apparus. Ces attributs éphemères correspondent plutôt à du bruit ou des attributs de l'objet sous-jacent qui ne sont pas pertinents. C'est le cas par exemple des neuf composantes connexes de l'union des boules au rayon α dans la Figure 4, qui ont disparu au rayon β, proche de α. Il est à noter que nous avons éxpliqué la construction dans le cas où il n'y a que trois unions de boules, mais il est bien sûr possible de construire un diagramme de persistance quand le rayon des boules augmente continument de 0 à $+\infty$. Dans ce cas, le trou de dimension 1 a une abscisse située entre α et β (car

il n'est pas encore présent pour le rayon α et est déjà là au rayon β), et une ordonnée située entre β et γ (car il a déjà disparu au rayon γ). De même, toutes les composantes connexes ont pour abscisse 0. Neuf d'entre elles [2] ont une ordonnée comprise entre α et β et l'ordonnée de la dixième est $+\infty$ puisqu'elle est toujours présente, quelque soit le rayon des boules.

Les diagrammes de persistance peuvent en faire être définis beaucoup plus généralement. - même si l'interprétation en terme d'échelle n'est plus forcément pertinente. Tout ce qui est requis est une famille d'espaces intriqués les uns dans les autres, appelée *filtration*, c'est-à-dire une famille $\{X_\alpha\}_{\alpha\in A}$, où A est un ensemble d'indices totalement ordonnés, telle que $\alpha \leq \beta \Rightarrow X_\alpha \subseteq X_\beta$. La construction du diagramme de persistance est alors la même, c'est-à-dire l'enregistrement de l'apparition et de la disparition d'attributs topologiques quand on parcourt A par ordre croissant. Dans l'exemple précédent, la filtration contient trois espaces, qui sont les trois différentes unions de boules, chaque union étant indicée par le rayon de ses boules. Il est clair dans ce cas que ces trois espaces sont intriqués car une boule est toujours incluse dans la boule de même centre avec un rayon supérieur.

Une manière pratique de construire une filtration est d'utiliser les *sous-niveaux* d'une fonction continue à valeurs réelles f, c'est-à-dire les espaces de la forme $f^{-1}((-\infty, \alpha])$. En effet, il est évident que $f^{-1}((-\infty, \alpha]) \subseteq f^{-1}((-\infty, \beta])$ pour tous $\alpha \leq \beta \in \mathbb{R}$. Par exemple, l'union des boules de rayon r centrées sur les points d'un nuage P est égale au sous-niveau de la fontion distance au nuage P : $d_P^{-1}((-\infty, r])$, où $d_P(x) = \min_{p\in P} d(x, p)$. Ainsi, dès qu'une fonction continue à valeurs réelles est à disposition, un diagramme de persistance peut être construit, ce qui explique pourquoi le diagramme de persistance est un descripteur prolifique. Prenons par exemple l'image floue d'un zéro, affichée dans le coin inférieur droit de la Figure 5, pour laquelle le niveau de gris des pixels est utilisé comme fonction continue pour calculer un diagramme de persistance. De nouveau, on trouve deux points se distinguant des autres dans le diagramme de persistance, l'un représentant la composant connexe du zéro, et l'autre son trou de dimension 1. Le reste des points est engendré par le bruit présent dans l'image.

Une des raisons pour lesquelles les diagrammes de persistance sont des descripteurs appréciés est qu'en plus d'être invariant par déformation continue (sans déchirement ou recollement), ils sont *stables* [18, 20]. En effet, si des diagrammes de persistance sont calculés avec les sous-niveaux de fonctions similaires, alors la distance entre eux est bornée supérieurement par la différence entre les fonctions en norme infinie :

$$d_{\mathrm{b}}(\mathrm{Dg}(f), \mathrm{Dg}(g)) \leq \|f - g\|_\infty,$$

2. En fait, chaque point est une composante connexe au rayon 0.

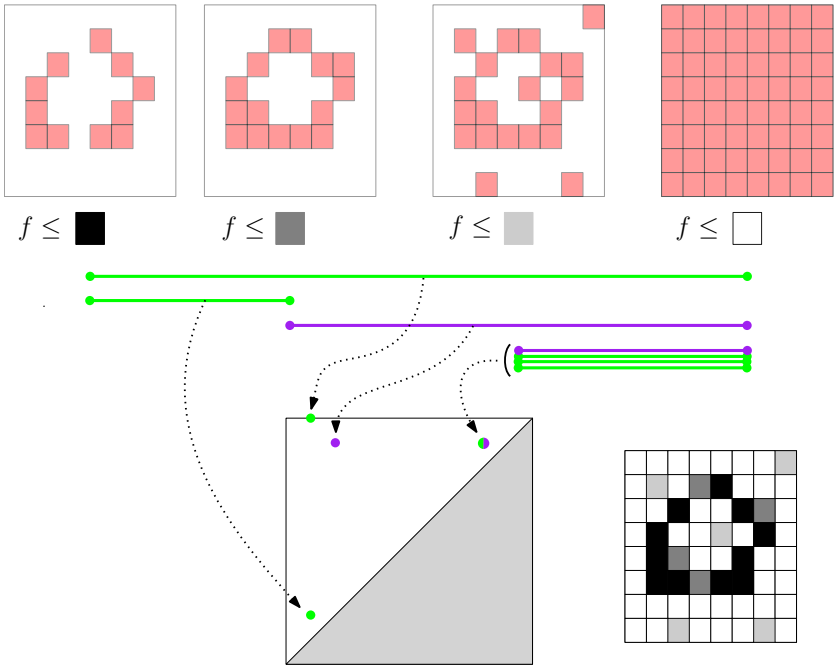

FIGURE 5 – Autre exemple d'une construction de diagramme de persistance, avec les sous-niveaux du niveau de gris des pixels d'une image floue d'un zéro.

où d_b désigne la distance bottleneck entre diagrammes de persistance, qui est le coût de la meilleure correspondance partielle entre les points de chaque diagramme. Cela signifie que, par exemple, si les positions des images de la Figure 4 sont légèrement perturbées, ou si l'image floue du zéro de la Figure 5 est légèrement modifiée, les diagrammes de persistance correspondant seront très proches des originaux avec la distance bottleneck.

Les diagrammes de persistance ont aidé à améliorer l'analyse des données dans de nombreuses applications, allant de l'analyse de forme 3D [16, 19] à la transition de phase de matériaux [24, 27] et la génomique [9, 17] pour n'en citer que quelques-unes.

Mapper. Comme expliqué plus haut, les diagrammes de persistance synthétisent l'information de nature topologique contenue dans les données. Cependant, ils perdent beaucoup d'information géométrique dans le processus : ils est aisé de construire des espaces différents ayant les mêmes diagrammes de persistance. Le *Mapper*[3], introduit par [46], est une approximation directe de l'objet sous-jacent, qui contient non seulement les attributs topologiques,

3. Dans cette thèse, on appelle *Mapper* l'objet mathématique, et pas l'algorithme utilisé pour le construire.

mais aussi de l'information additionnelle, concernant le positionnement des attributs les uns par rapport aux autres par exemple. Comme pour les diagrammes de persistance, une fonction réelle continue, appelée parfois *filtre*, est requise, ainsi qu'une couverture de son image par des intervalles ouverts qui se chevauchent. L'idée est de calculer les antécédents par f de tous les intervalles de la couverture, de les raffiner en leurs composantes connexes via des techniques de clustering, et de finalement lier les composantes connexes entre elles si elles contiennent des points de données en commun.

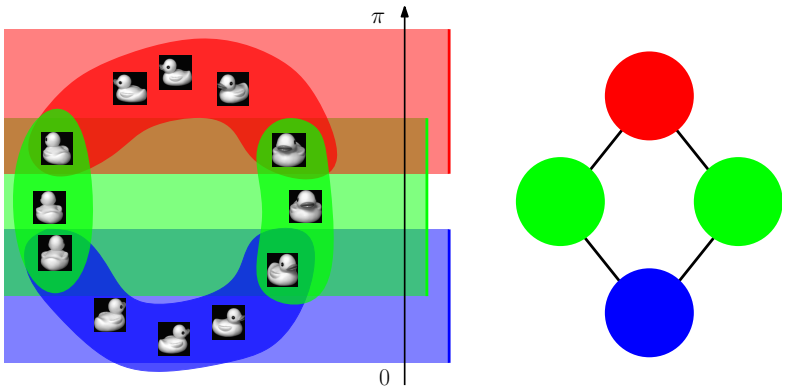

FIGURE 6 – Exemple de Mapper calculé sur le nuage d'images, avec la fonction d'angle et une couverture de trois intervalles.

Nous fournissons un exemple dans la Figure 6, où nous considérons de nouveau le nuage d'images. La fonction réelle continue est la valeur absolue de l'angle à partir duquel l'image a été prise, et son image $[0, \pi]$ est couverte par trois intervalles (bleu, rouge et vert). Dans les antécédents des intervalles rouge et bleu, il y a une seule composant connexe, tandis qu'il y en a deux dans l'antécédent de l'intervalle vert. Le Mapper est obtenu en ajoutant des arètes entre les composantes connexes, en fonction de la présence ou non de points de données en commun à l'intérieur de ces composantes ; par exemple, les composantes connexes vertes et bleues, ou vertes et rouges, sont reliées, mais pas celles qui sont rouges et bleues. Le Mapper a l'homologie d'un cercle, est constitue une approximation directe du support sous-jacent au nuage d'images.

Il est bon de remarquer que les longueurs des intervalles contrôlent directement l'échelle à partir de laquelle on observe le nuage : si les intervalles sont petits, le Mapper va avoir beaucoup de composantes déconnectées puisque les antécédents contiendront au plus un point de donnée. A l'opposé, si les intervalles sont larges, le Mapper aura peu de composantes puisque les antécédents vont contenir beaucoup de points de données.

En pratique, le Mapper a deux domaines d'applications majeures. Le

premier est la visualisation et le clustering. En effet, le Mapper fournit une visualisation des données sous forme de graphe dont la topologie reflète celle des données. Il apporte ainsi une information complémentaire à celle des algorithmes de clustering usuels concernant la structure interne des clusters par l'identification de *branches* et de *boucles* qui mettent en lumière des attributs topologiques potentiellement remarquables dans les groupes identifiés par clustering. Voir par exemple [50, 34, 45, 26] pour des exemples d'applications. La deuxième application est la sélection d'attributs. En effet, chaque attribut des données peut être évalué en regard de sa capacité à différencier les attributs topologiques mentionnés plus haut (branches et boucles) du reste des données, via l'utilisation de tests statistiques, comme celui de Kolmogorov-Smirnov. Voir par exemple [34, 39, 43] pour des exemples d'applications.

2 Limitations

Même si le Mapper et les diagrammes de persistance bénéficient de propriétés désirables, plusieurs limitations refrènent leur usage pratique, à savoir la *la difficulté de la sélection de paramètres pour Mapper* et la *non linéarité* de l'espace des diagrammes de persistance.

Distance et stabilité pour les Mappers et les graphes de Reeb Un problème du Mapper est que, contrairement aux diagrammes de persistance, il a un paramètre, la couverture, dont la sélection à priori est difficile. A cause de cela, le Mapper apparaît comme une construction très *instable* : il arrive que des Mappers calculés sur des nuages de points similaires, comme dans la Figure 7, ou avec des couvertures proches, comme dans la Figure 8, soient très différents.

Ce problème majeur est un obstacle important à son utilisation en exploration de données. La seule réponse dans l'état-de-l'art consiste à sélectionner des paramètres dans une grille de valeurs pour lesquels le Mapper semble stable - voir [39] par exemple.

Ainsi, prouver un résultat de stabilité pour les Mappers nécessite de les comparer avec une distance qui dépend au moins de la couverture utilisée. Malheureusement, même si des distances théoriques peuvent être définies [37], la définition d'une distance calculable et interprétable entre Mappers manque dans l'état-de-l'art. Pour gérer ce problème, on peut prendre inspiration d'une classe de descripteurs très semblables aux Mappers, les *graphes de Reeb*.

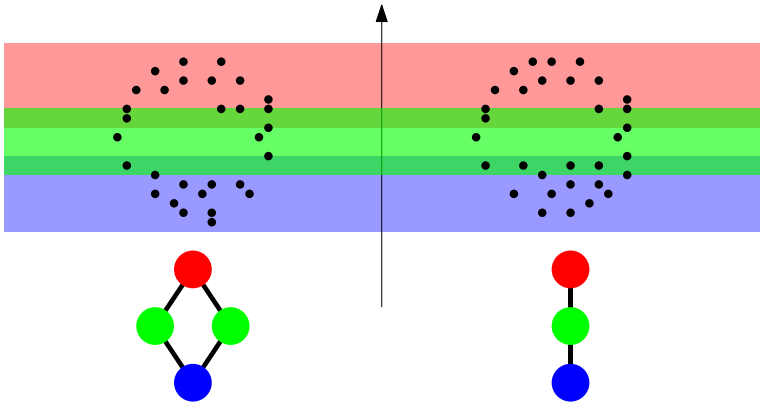

FIGURE 7 – Mappers calculés sur des échantillonnages similaires du cercle, avec la fonction hauteur et une couverture composée de trois intervalles.

Graphes de Reeb. Même si les Mappers sont définis pour des nuages de points, leur extension à des espaces non discrets est évidente, la différence étant que des techniques de clustering ne sont pas nécessaires pour calculer les composantes connexes des antécédents puisqu'elles sont bien définies. Dans ce cas, faire tendre la longueur des intervalles vers zéro définit le *graphe de Reeb*. Ainsi, les Mappers (calculés sur des espaces non discrets) ne sont que des *approximations*, ou des *versions pixelisées* des graphes de Reeb, comme illustré dans la Figure 9.

Cette observation est cruciale car plusieurs distances, ainsi que des résultats de stabilité, ont été obtenus pour les graphes de Reeb [4, 5, 22] et peuvent être étendus aux Mappers. Cependant, ces distances ne sont pas calculables et ne peuvent pas être utilisées en tant que telles en pratique [2]. La question de savoir s'il est possible de définir des distances stables et calculables pour les Mappers reste ainsi ouverte.

Non linéarité de l'espace des diagrammes de persistance. Même si les diagrammes de persistance sont stables, ils ne peuvent pas être utilisés systématiquement par des algorithmes d'apprentissage automatique. En effet, une classe très large de ces algorithmes nécessitent que les données soient soit des vecteurs d'un espace Euclidien (comme les forêts aléatoires), ou d'un espace de Hilbert (comme les SVM). L'espace des diagrammes de persistance, équipé avec la distance bottleneck, n'est malheureusement ni l'un ni l'autre. Même les moyennes de Fréchet ne sont pas bien définies [49]. L'*astuce du noyau* permet cependant de traiter ce genre de données. En supposant que les points de données vivent dans un espace métrique (X, d_X), l'astuce du noyau nécessite seulement une fonction semi-définie positive, appelée *noyau*,

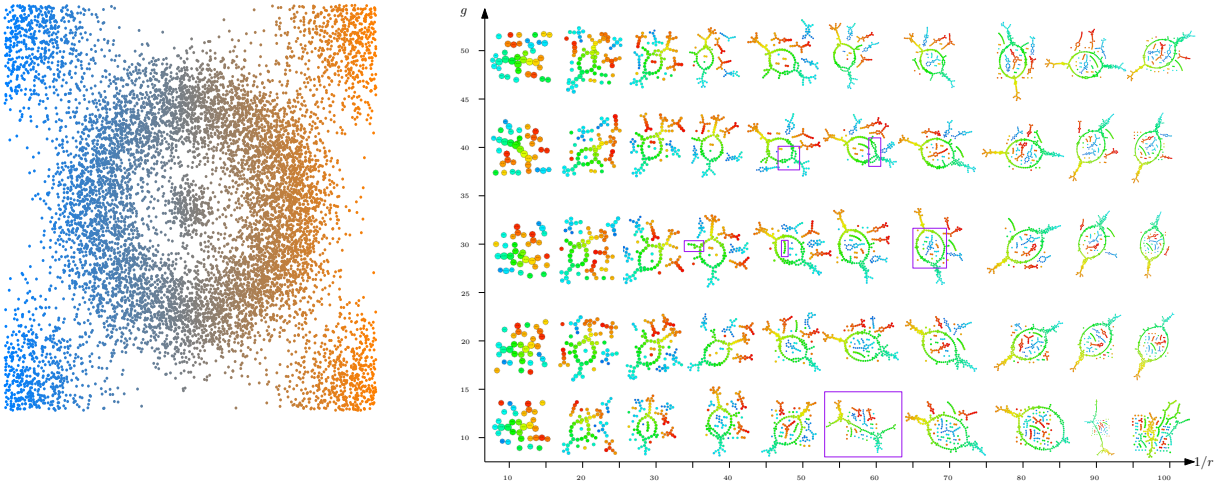

FIGURE 8 – Un ensemble de Mappers calculés sur le jeu de données du cratère avec des couvertures différentes (r est la longueur des intervalles et g est le pourcentage de chevauchement) et la coordonnée horizontale. Gauche : jeu de données du cratère coloré par les valeurs de fonction, allant de bleu à orange. Droite : Mappers calculés avec des paramètres différents. Les rectangles violets indiquent les attributs topologiques qui apparaissent ou disparaissent soudainement dans les Mappers.

c'est-à-dire une fonction $k : X \times X \to \mathbb{R}$ telle que, pour tous $a_1, \cdots, a_n \in \mathbb{R}$ et $x_1, \cdots, x_n \in X$, on ait :

$$\sum_{i,j} a_i a_j k(x_i, x_j) \geq 0.$$

Gràce au théorême de Moore-Aronszajn [3], les valeurs du noyau calculées sur des points de données peuvent être démontrées égales à l'évaluation d'un produit scalaire entre les images des points de données par un plongement dans un espace de Hilbert spécifique qui dépend uniquement de k et qui est en général inconnu. Plus formellement, il existe un espace de Hilbert $\mathcal{H}_k$ tel que, pour tous $x, y \in X$, on ait :

$$k(x, y) = \langle \Phi_k(x), \Phi_k(y) \rangle_{\mathcal{H}_k},$$

pour un certain plongement Φ_k. Les valeurs du noyau peuvent donc être considérées comme des produits scalaires généralisés entre les points de données, et peuvent être directement utilisés par les algorithmes d'apprentissage. Dans le cas qui nous intéresse, la question est ainsi de trouver de tels noyaux pour les diagrammes de persistance.

Une manière standard de procéder pour définir un noyau pour des points

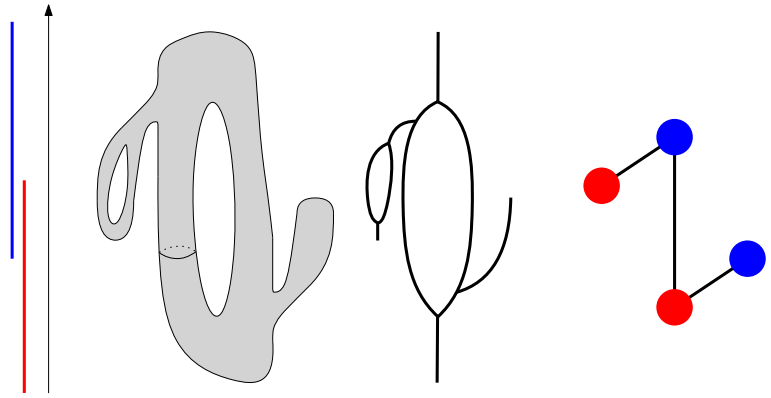

FIGURE 9 – Une surface plongée dans $\mathbb{R}^3$ (gauche), son graphe de Reeb calculé avec la fonction hauteur (milieu) et son Mapper calculé avec la fonction hauteur et une couverture à deux intervalles (droite).

d'un espace métrique (X, d_X) est d'utiliser des fonctions *Gaussiennes* :

$$k_\sigma(x, y) = \exp\left(-\frac{d_X(x, y)}{2\sigma^2}\right),$$

où $\sigma > 0$ est un paramètre d'échelle. Un théorême de Berg et al. [6] stipule que k_σ est un noyau, c'est-à-dire une fonction semi-définie positive, pour tous $\sigma > 0$ si et seulement si d_X est *conditionnellement semi-définite négative*, c'est-à-dire est telle qu'on ait $\sum_{i,j} a_i a_j d_X(x_i, x_j) \leq 0$ pour tous $x_1, \cdots, x_n \in X$ et $a_1, \cdots, a_n \in \mathbb{R}$ tels que $\sum_{i=1}^n a_i = 0$. Malheureusement, comme montré par Reininghaus et al. [41], la distance bottleneck d_b pour les diagrammes de persistance n'est pas conditionnellement semi-définite négative. Il est même possible de trouver des contre-exemples pour les distances de Wasserstein, une autre classe de distance pour diagrammes. L'utilisation de noyaux Gaussiens pour les diagrammes de persistance est donc impossible avec leurs métriques canoniques.

Néanmoins, plusieurs noyaux ont été proposés au cours des dernières années [1, 8, 31, 42], bénéficiant tous de résultats de stabilité bornant supérieurement la distance entre les plongements des diagrammes par les distances bottleneck ou de Wasserstein entre les diagrammes eux-mêmes. En d'autres termes, la distorsion métrique

$$\mathrm{dist}(\mathrm{Dg}, \mathrm{Dg}') = \frac{\|\Phi_k(\mathrm{Dg}) - \Phi_k(\mathrm{Dg}')\|_{\mathcal{H}_k}}{d_\mathrm{b}(\mathrm{Dg}, \mathrm{Dg}')}$$

est bornée supérieurement. Cependant, le calcul d'une borne inférieure non triviale reste ouvert : il se pourrait que les plongements de diagrammes différents soient en fait très proches l'un de l'autre, ce qui n'est pas désirable

en pratique pour la discriminativité d'un noyau. Par exemple, le plongement constant, qui envoie tous les diagrammes sur un même point d'un espace de Hilbert spécifique, est stable (les distances entre images dans l'espace de Hilbert étant toujours nulles), mais les résultats du noyau correspondant seront évidemment très faibles. Plus généralement, le comportement et les propriétés des distances dans les espaces de Hilbert induits par des noyaux sont flous, et la question de savoir s'il existe des noyaux avec des propriétés théoriques de discriminativité est ouverte.

3 Contributions

Dans cette thèse, nous nous penchons sur trois problèmes : l'interprétation des attributs topologiques) du Mapper (par exemple avec des régions de confiance), le réglage de ses paramètres, et l'intégration globale des descripteurs topologiques en apprentissage automatique.

Distance entre graphes de Reeb. Dans le Chapitre 3, nous définissons une pseudodistance calculable entre graphes de Reeb, qui revient à comparer leurs diagrammes de persistance. Nous montrons aussi que cette pseudodistance est en fait *localement équivalente* aux autres distances existantes pour les graphes de Reeb. Cette équivalence locale est alors utilisée pour étudier les propriétés de l'espace métrique des graphes de Reeb, équipé des distances *intrinsèques*. Nous montrons que toutes ces distances intrinsèques sont *fortement équivalentes*, ce qui nous permet d'englober toutes les techniques pour comparer des graphes de Reeb en une seule approche. Ce travail a été publié dans les proceedings du Symposium on Computational Geometry 2017 [15].

Structure du Mapper. Dans le Chapitre 4, nous fournissons un lien entre les diagrammes de persistance du graphe de Reeb et ceux du Mapper (calculé sur le même espace topologique). Plus spécifiquement, nous montrons que le diagramme de persistance du Mapper est obtenu à partir de celui du graphe de Reeb en supprimant des points spécifiques, à savoir ceux qui appartiennent à des régions du plan qui dépendent uniquement de la couverture utilisée pour calculer le Mapper. Cette relation explicite nous permet alors d'étendre la pseudodistance entre graphes de Reeb aux Mappers. Nous montrons finalement que cette pseudodistance *stabilise* les Mappers : nous fournissons un théorême de stabilité pour des Mappers comparés avec cette pseudodistance. Ce travail a été publié dans les proceedings du Symposium on Computational Geometry 2016 [14] et une version longue a été soumise et acceptée au Journal of Foundations of Computational Mathematics [13].

Cas discret. Dans le Chapitre 5, nous étendons les résultats précédents au cas où les Mappers sont calculés sur des espaces discrets, c'est-à-dire des nuages de points, et les composantes connexes sont calculées avec du single-linkage clustering. En particulier, nous fournissons des conditions suffisantes pour lesquelles le Mapper calculé sur un nuage de points coincide avec celui calculé sur le support. De plus, nous montrons que le Mapper converge vers le graphe de Reeb avec une vitesse de convergence *optimale*, au sens où aucun estimateur du graphe de Reeb ne peut converger plus vite. Les paramètres utilisés pour démontrer l'optimalité fournissent en plus des *heuristiques pour le réglage automatique* de ces paramètres. Ces heuristiques se basent sur des techniques de sous-échantillonnage et dépendent uniquement de la cardinalité du nuage de points de données. Finalement, nous proposons un moyen de calculer des *régions de confiance* pour les différents attributs topologiques du Mapper. Ce travail a été soumis et accepté au Journal of Machine Learning Research [12].

Méthodes à noyaux. Dans le Chapitre 6, nous appliquons des techniques d'apprentissage aux diagrammes de persistance, via des méthodes à noyaux.

Nous définissons d'abord un *noyau Gaussien* en utilisant une modification de la distance de Wasserstein, appelée distance de *Sliced Wasserstein*. Nous montrons en effet que cette distance, à l'inverse de la distance de Wasserstein, est bien conditionnellement semi-définie négative, et permet donc de définir un noyau Gaussien. De plus, nous montrons que la distance induite dans l'espace de Hilbert associé est *équivalente* à la distance de Wasserstein de départ. Ainsi, ce noyau, en plus d'être stable et Gaussien, est aussi théoriquement discriminant. Nous en fournissons aussi une preuve empirique en obtenant de nettes améliorations par rapport aux autres noyaux de l'état-de-l'art dans plusieurs applications. Ce travail a été publié dans les proceedings de l'International Conference on Machine Learning 2017 [11].

Enfin, nous définissons aussi une *méthode de vectorisation* pour envoyer les diagrammes de persistance dans $\mathbb{R}^D$, où $D \in \mathbb{N}^*$. Ce plongement stable, même si non injectif, permet l'usage des diagrammes de persistance pour des problèmes et algorithmes où des vecteurs Euclidiens sont nécessaires. Nous détaillons alors une application où une telle structure est requise, à savoir le traitement de formes 3D, pour laquelle nous démontrons que les diagrammes de persistance apportent une information complémentaire aux descripteurs traditionnels. Ce travail a été publié dans les proceedings du Symposium on Geometry Processing 2015 [16].

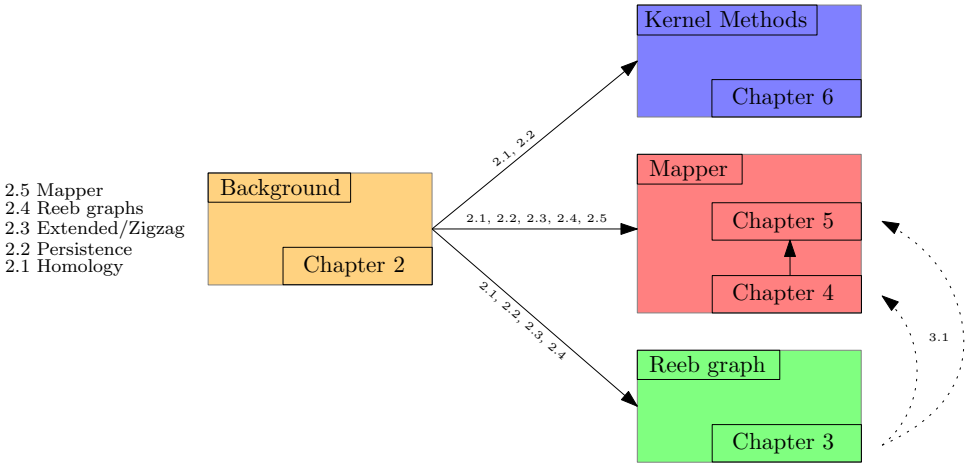

FIGURE 10 – Les flèches indiquent des dépendances entre chapitres, et les flèches en pointillés indiquent des dépendances partielles, c'est-à-dire que seule une petite et non essentielle partie du chapitre dépend de l'autre.

Comment lire cette thèse ? Cette thèse est composée de quatre parties différentes :

- La première est le Chapitre 2, dans lequel nous détaillons les fondations théoriques de l'homologie, la persistance, les graphes de Reeb et les Mappers. Nous expliquons aussi la *persistance étendue* et la *persistance en zigzag*.
- La deuxième partie est le Chapitre 3, qui traite des graphes de Reeb et de leurs distances.
- La troisième partie est composée des Chapitres 4 et 5, qui traitent de Mapper.
- La quatrième partie est le Chapitre 6. Il traite des noyaux pour les diagrammes de persistance, dans des espaces de Hilbert en dimension finie et infinie.

Voir la Figure 10. Le Chapitre 2 rappelle essentiellement les fondamentaux en topologie. Les autres chapitres contiennent en revanche les contributions de cette thèse. Les Chapitres 3 et 4 sont très orientés topologie, tandis que le Chapitre 5 utilise plutôt des notions de statistiques, et que le Chapitre 6 se concentre davantage sur l'apprentissage automatique. Ces chapitres ne sont pas indépendants, comme illustré par la Figure 10, mais les contributions de chaque chapitre sont énoncées dans les introductions correspondantes. Ainsi, pour chacun de ces chapitres, le lecteur, en fonction de ses goûts ou connaissances personnelles, peut soit se limiter à l'introduction, soit lire le chapitre dans son intégralité.

Références

[1] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence Images : A Stable Vector Representation of Persistent Homology. *Journal of Machine Learning Research*, 18(8) :1–35, 2017.

[2] Pankaj Agarwal, Kyle Fox, Abhinandan Nath, Anastasios Sidiropoulos, and Yusu Wang. Computing the Gromov-Hausdorff Distance for Metric Trees. In *Proceedings of the 26th International Symposium on Algorithms and Computation*, 2015.

[3] Nachman Aronszajn. Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, 68 :337–404, 1950.

[4] Ulrich Bauer, Barbara Di Fabio, and Claudia Landi. An Edit Distance for Reeb Graphs. In *Proceedings of the Eurographics 2016 Workshop on 3D Object Retrieval*, pages 27–34, 2016.

[5] Ulrich Bauer, Xiaoyin Ge, and Yusu Wang. Measuring Distance Between Reeb Graphs. In *Proceedings of the 30th Symposium on Computational Geometry*, pages 464–473, 2014.

[6] Christian Berg, Jens Christensen, and Paul Ressel. *Harmonic Analysis on Semigroups : Theory of Positive Definite and Related Functions*. Springer, 1984.

[7] Eckart Bindewald and Bruce Shapiro. Rna secondary structure prediction from sequence alignments using a network of k-nearest neighbor classifiers. *RNA*, 12(3) :342–352, 2006.

[8] Peter Bubenik. Statistical Topological Data Analysis using Persistence Landscapes. *Journal of Machine Learning Research*, 16 :77–102, 2015.

[9] Pablo Camara, Arnold Levine, and Raul Rabadan. Inference of Ancestral Recombination Graphs through Topological Data Analysis. *PLoS Computational Biology*, 12(8) :1–25, 2016.

[10] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46 :255–308, 2009.

[11] Mathieu Carrière, Marco Cuturi, and Steve Oudot. Sliced Wasserstein Kernel for Persistence Diagrams. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[12] Mathieu Carrière, Bertrand Michel, and Steve Oudot. Statistical Analysis and Parameter Selection for Mapper. *CoRR*, abs/1706.00204, 2017.

[13] Mathieu Carrière and Steve Oudot. Structure and Stability of the 1-Dimensional Mapper. *CoRR*, abs/1511.05823, 2015.

[14] Mathieu Carrière and Steve Oudot. Structure and Stability of the 1-Dimensional Mapper. In *Proceedings of the 32nd Symposium on Computational Geometry*, volume 51, pages 25 :1–25 :16, 2016.

[15] Mathieu Carrière and Steve Oudot. Local Equivalence and Induced Metrics for Reeb Graphs. In *Proceedings of the 33rd Symposium on Computational Geometry*, 2017.

[16] Mathieu Carrière, Steve Oudot, and Maks Ovsjanikov. Stable Topological Signatures for Points on 3D Shapes. *Computer Graphics Forum*, 34, 2015.

[17] Joseph Chan, Gunnar Carlsson, and Raul Rabadan. Topology of viral evolution. *Proceedings of the National Academy of Science*, 110(46) :18556–18571, 2013.

[18] Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas Guibas, and Steve Oudot. Proximity of Persistence Modules and their Diagrams. In *Proceedings of the 25th Symposium on Computational Geometry*, pages 237–246, 2009.

[19] Frédéric Chazal, David Cohen-Steiner, Leonidas Guibas, Facundo Mémoli, and Steve Oudot. Gromov-Hausdorff Stable Signatures for Shapes using Persistence. *Computer Graphics Forum*, pages 1393–1403, 2009.

[20] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of Persistence Diagrams. *Discrete and Computational Geometry*, 37(1) :103–120, 2007.

[21] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[22] Vin de Silva, Elizabeth Munch, and Amit Patel. Categorified Reeb Graphs. *Discrete and Computational Geometry*, 55 :854–906, 2016.

[23] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning*. Springer series in statistics Springer, 2001.

[24] Marcio Gameiro, Yasuaki Hiraoka, and Ippei Obayashi. Continuation of Point Clouds via Persistence Diagrams. *Physica D : Nonlinear Phenomena*, 334 :118–132, 2016.

[25] Sara Goodwin, John McPherson, and Richard McCombie. Coming of age : ten years of next-generation sequencing technologies. *Nature Review Genetics*, 17(6) :333–351, 2016.

[26] TS. Hinks, X. Zhou, KJ. Staples, BD. Dimitrov, A. Manta, T. Petrossian, P. Lum, CG. Smith, JA. Ward, PH Howarth, AF. Walls, SD. Gadola, and R. Djukanovic. Innate and adaptive t cells in asthmatic patients : Relationship to severity and disease mechanisms. *Journal of Allergy and Clinical Immunology*, 136(2) :323–333, 2015.

[27] Yasuaki Hiraoka, Takenobu Nakamura, Akihiko Hirata, Emerson Escolar, Kaname Matsue, and Yasumasa Nishiura. Hierarchical structures of amorphous solids characterized by persistent homology. In *Proceedings of the National Academy of Science*, volume 26, 2016.

[28] Andrew Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5) :433–449, 1999.

[29] Min-su Kim, Benjamin Hur, and Sun Kim. Rddpred : a condition-specific rna-editing prediction model from rna-seq data. *BMC Genomics*, 17(1), 2016.

[30] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[31] Genki Kusano, Kenji Fukumizu, and Yasuaki Hiraoka. Persistence Weighted Gaussian Kernel for Topological Data Analysis. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2004–2013, 2016.

[32] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[33] David Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2) :91–110, 2004.

[34] P. Lum, G. Singh, A. Lehman, T. Ishkanov, M. Vejdemo-Johansson, M. Alagappan, J. Carlsson, and G. Carlsson. Extracting insights from the shape of complex data using topology. *Scientific Reports*, 3, 2013.

[35] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 2008.

[36] Michael Metzker. Sequencing technologies - the next generation. *Nature Review Genetics*, 11(1) :31–46, 2010.

[37] Elizabeth Munch and Bei Wang. Convergence between Categorical Representations of Reeb Space and Mapper. In *Proceedings of the 32nd Symposium on Computational Geometry*, volume 51, pages 53 :1–53 :16, 2016.

[38] Sameer Nene, Shree Nayar, and Hiroshi Murase. Columbia Object Image Library (COIL-100). Technical Report CUCS-006-96, 1996.

[39] Jessica Nielson, Jesse Paquette, Aiwen Liu, Cristian Guandique, Amy Tovar, Tomoo Inoue, Karen-Amanda Irvine, John Gensel, Jennifer Kloke, Tanya Petrossian, Pek Lum, Gunnar Carlsson, Geoffrey Manley, Wise Young, Michael Beattie, Jacqueline Bresnahan, and Adam Ferguson. Topological data analysis for discovery in preclinical spinal cord injury and traumatic brain injury. *Nature Communications*, 6, 2015.

[40] Timo Ojala, Topi Mäenpää, Matti Pietikäinen, Jaakko Viertola, Juha Kyllönen, and Sami Huovinen. Outex - new framework for empirical evaluation of texture analysis algorithms. In *Proceedings of the 16th International Conference on Pattern Recognition*, pages 701–706, 2002.

[41] Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A Stable Multi-Scale Kernel for Topological Machine Learning. *CoRR*, abs/1412.6821, 2014.

[42] Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A Stable Multi-Scale Kernel for Topological Machine Learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[43] Matteo Rucco, Emanuela Merelli, Damir Herman, Devi Ramanan, Tanya Petrossian, Lorenzo Falsetti, Cinzia Nitti, and Aldo Salvi. Using topological data analysis for diagnosis pulmonary embolism. *Journal of Theoretical and Applied Computer Science*, 9(1) :41–55, 2015.

[44] Bryan Russell, Antonio Torralba, Kevin Murphy, and William Freeman. LabelMe : A Database and Web-Based Tool for Image Annotation. *International Journal of Computer Vision*, 77(1-3) :157–173, 2008.

[45] G. Sarikonda, J. Pettus, S. Phatak, S. Sachithanantham, JF. Miller, JD. Wesley, E. Cadag, J. Chae, L. Ganesan, R. Mallios, S. Edelman, B. Peters, and M. von Herrath. Cd8 t-cell reactivity to islet antigens is unique to type 1 while cd4 t-cell reactivity exists in both type 1 and type 2 diabetes. *Journal of Autoimmunity*, 50 :77–82, 2014.

[46] Gurjeet Singh, Facundo Mémoli, and Gunnar Carlsson. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In *Symposium on Point Based Graphics*, pages 91–100, 2007.

[47] George Soumya and Joseph Shibily. Text Classification by Augmenting Bag of Words (BOW) Representation with Co-occurrence Feature. *IOSR Journal of Computer Engineering*, 16 :34–38, 2014.

[48] Joshua Tenenbaum, Vin de Silva, and John Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500), 2000.

[49] Katharine Turner, Yuriy Mileyko, Sayan Mukherjee, and John Harer. Fréchet Means for Distributions of Persistence Diagrams. *Discrete and Computational Geometry*, 52(1) :44–70, 2014.

[50] Yuan Yao, Jian Sun, Xuhui Huang, Greg Bowman, Gurjeet Singh, Michael Lesnick, Leonidas Guibas, Vijay Pande, and Gunnar Carlsson. Topological methods for exploring low-density states in biomolecular folding pathways. *Journal of Chemical Physics*, 130(14), 2009.