Détection non supervisée de motifs sur séries temporelles

Thibaut GERMAIN Alexandre BOIS Charles TRUONG Brian TERVIL Laurent OUDRE

Université Paris Saclay, Université Paris Cité, ENS Paris Saclay, CNRS, SSA, INSERM, Centre Borelli, F-91190, Gif-sur-Yvette, France

Résumé – Nous présentons un nouvel algorithme de détection de motifs dans des séries temporelles. Il utilise un graphe pour quantifier la similarité entre chaque pair de sous-séquences d'un signal, puis des outils d'analyse topologique de données permettant de reconstruire les motifs à partir du graphe sans en connaître le nombre a priori.

Abstract – We present a new algorithm for pattern detection in time series. It uses a graph that quantifies the similarity of pairs of subsequences of a given signal, and topological data analysis tools to detect and reconstruct the patterns from the graph without any information a priori about the number of patterns.

1 Introduction

La détection non-supervisée de motifs consiste à trouver dans un signal des formes qui se répètent et leurs occurrences, sans information a priori sur les motifs. Elle permet de résumer un long signal en un nombre limité de motifs, qui peuvent être utilisés dans des analyses postérieures pour réduire la complexité en temps, améliorer les performances ou interpréter des résultats. Cette problématique trouve son intérêt dans plusieurs domaines tels que l'industrie ou la médecine. Par exemple, l'électrocardiogramme d'un patient souffrant d'extrasystole ventriculaire (ESV), c'est à dire des contractions prématurées du ventricule cardiaque, inclut un motif pour la contraction normale et un autre pour la contraction prématurée, figure 1. Quantifier la fréquence des ESV permet d'évaluer un risque de tachycardie.

Les algorithmes de détection de motifs s'inspirent des méthodes de clustering [9], ils incorporent la structure temporelle d'un signal pour améliorer les performances [2] et la complexité en temps [5]. Le premier algorithme proposé [4] apprend les motifs dont le nombre d'occurrences est maximal compte tenu d'un seuil de similarité entre occurrences. Néanmoins, cet algorithme suppose que la durée des motifs est connue et identique. Plusieurs alternatives ont été proposées pour s'affranchir de cette contrainte [3], mais tous ces algorithmes supposent que le nombre de motifs est connu à l'avance.

Dans cette étude, nous présentons un algorithme capable de détecter des motifs de tailles variables et dont le choix du nombre de motifs se fait à posteriori grâce à une représentation graphique adaptée. Il s'appuie sur un graphe qui quantifie la proximité entre différentes sous-séquences du signal, et sur des outils d'analyse topologique de données [1] pour les regrouper par similarité.

2 Présentation de la méthode

La détection de motifs se décompose en deux tâches :

- Séparer les zones d'occurrence des motifs du reste du signal.
- Identifier les occurrences qui correspondent à un même motif.

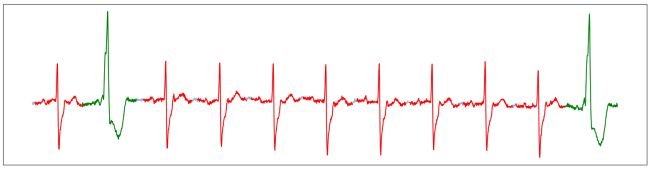

FIGURE 1 : ECG d'un patient souffrant d'extrasystole ventriculaire (ESV). Les ESV sont en vert et les contractions normales du ventricule en rouges.

La résolution des deux tâches peut se faire en définissant une distance permettant de comparer des sous-séquences du signal deux à deux. L'idée est qu'un motif se répétant plusieurs fois dans le signal donnera plusieurs sous-séquences proches entre elles, alors qu'une sous-séquence unique ou composée uniquement de bruit sera plus éloignée de toutes les autres. Une structure de graphe est adaptée à de telles comparaisons car elle permet de quantifier la proximité entre chaque paire de sous-séquences et de les regrouper de proche en proche. Notre méthode se décompose en trois étapes :

1. Transformer le signal en un graphe dont chaque noeud représente une sous-séquence et dont les arrêtes sont pondérées à l'aide d'une distance entre sous-séquences.
2. Sur le graphe, identifier des clusters (sous-séquences d'une occurence d'un motif) et des noeuds isolés (sous-séquences du reste du signal).
3. Reconstruire les motifs et déterminer leurs occurrences à partir des groupes.

2.1 Représentation sous forme de graphe

Dans cette partie nous présentons notre construction d'un graphe à partir d'un signal.

On rappelle qu'un graphe pondéré non-orienté est un triplet $G = (V, E, d)$ où V est l'ensemble des noeuds, $E \subset V \times V$ est l'ensemble de ses arêtes et $d : E \mapsto \mathbb{R}_+$ est une fonction de poids (on considère ici que le poids est d'autant plus faible que les noeuds sont proches).

On commence par définir une distance entre sous-séquences :

Definition 1 *Soient $x, y \in \mathbb{R}^w$. La distance euclidienne normalisée entre x et y est définie par :*

$$d_Z(x,y) = \left\| \frac{x-\bar{x}}{\sigma_x} - \frac{y-\bar{y}}{\sigma_y} \right\|_2 \quad (1)$$

où $\bar{x} = \frac{1}{w}\sum_{i=1}^{w} x[i]$, $\sigma_x^2 = \frac{1}{w-1}\sum_{i=1}^{w}(x[i]-\bar{x})^2$ et $\|.\|_2$ désigne la norme euclidienne.

Pour un signal $\mathbf{s} \in \mathbb{R}^L$ on définit le graphe suivant :

- V est l'ensemble des sous-séquences de taille w du signal $\mathbf{s}$: $V = \{\mathbf{s}[i : i+w] \mid i \in [1, L-w+1]\}$. La taille de la fenêtre w est un paramètre de la méthode.
- E est l'ensemble des paires de sous-séquences qui ne se chevauchent pas : $E = \{(\mathbf{s}[i : i+w], \mathbf{s}[j : j+w]) \in V \times V \mid |i-j| > w\}$.
- d est la distance euclidienne normalisée : $d = d_Z$.

Par la suite, on considérera la taille de fenêtre w fixée, et plus petite que la longueur de chaque motif. Si w est plus grande que la longueur d'un motif, sa détection sera imprécise. En effet, une sous-séquence correspondant à ce motif inclura une partie du reste du signal et/ou du motif suivant, donc elle ne se répétera pas lors des autres occurrences du motif.

Les raisons du choix de la distance euclidienne normalisée sont les suivantes :

- La normalisation permet d'avoir de grandes distances entre deux sous-séquences composées uniquement de bruit centré en 0. En revanche, comme l'amplitude du bruit est négligeable devant celle du signal, la distance entre deux sous-séquences similaires est faible [7].
- Cette distance est invariante par changement d'amplitude et translation d'une des séquences, c'est à dire que pour tout $x, y \in \mathbb{R}^w$, $a > 0$ et $b \in \mathbb{R}$, $d(x,y) = d(x, a \times y + b)$. Deux sous-séquences translatées et/ou amplifiées l'une par rapport à l'autre seront donc détectées comme occurrences du même motif (on choisit de se concentrer sur la forme du signal et d'ignorer les translations/amplifications).
- Elle permet un calcul efficace de toutes les distances : en suivant l'algorithme [10], la complexité en temps est en $\mathcal{O}(L^2)$ au lieu de $\mathcal{O}(CL^2)$ pour la méthode classique consistant à calculer toutes les distances entre paires de sous-séquences, où C est le temps de calcul de la distance entre deux sous-séquences. Le calcul parallèle est aussi possible.

2.2 Clustering de graphe par homologie persistante

Cette partie décrit l'algorithme de clustering de graphe utilisé. Il repose sur la notion de persistance, centrale en analyse topologique de données.

Definition 2 *Soit $G = (V, E, d)$ un graphe. $G' = (V', E', d)$ est un sous-graphe de G si $V' \subset V$ et $E' \subset E$.*

Definition 3 *Une filtration (croissante[1]) sur un graphe $G = (V, E, d)$ est une suite croissante $(G_\alpha)_{\alpha \in I} = (V_\alpha, E_\alpha)_{\alpha \in I}$ de sous-graphes de G (i.e si $\alpha \leq \beta$ alors G_α est un sous-graphe de G_β), où I est un sous-ensemble quelconque de $\mathbb{R}$.*

Une filtration construit donc un graphe en y ajoutant successivement des noeuds et des arêtes. On utilisera ici la filtration suivante, appelée filtration NNVR (pour *Nearest Neighbor Vietoris-Rips*, cette filtration est une variante de la très classique filtration de Vietoris-Rips [1]). Elle ajoute les arêtes par poids croissant (et donc, dans notre application, par distance croissante entre sous-séquences) et ajoute les noeuds en fonction de leur distance au plus proche voisin, ce qui permet de détecter les noeuds isolés.

Definition 4 *Soit $G = (V, E, d)$ un graphe. La filtration $(NNVR(G, \alpha))_{\alpha \in \mathbb{R}_+}$ est définie pour tout α positif de la manière suivante :*

- *Un noeud $x \in V$ est dans $NNVR(G, \alpha)$ si et seulement si $\inf_{y \in V, (x,y) \in E} d(x,y) \leq \alpha$.*
- *Une arête $(x, y) \in E$ est dans $NNVR(G, \alpha)$ si et seulement si $d(x, y) \leq \alpha$.*

La figure 2-A ((a)-(g)) montre un exemple de filtration NNVR sur un graphe. Lorsque $\alpha = 1$, les cinq noeuds à une distance 1 de leur plus proche voisin sont ajoutés, ainsi que les arêtes de poids 1. Lorsque $\alpha = 2$, les noeuds 6 et 2 sont ajoutés avec les arêtes de poids 2, et ainsi de suite jusqu'à reconstruction complète du graphe à $\alpha = 15$.

Lorsque l'on fait augmenter la valeur du paramètre α, on ajoute des noeuds ou des arêtes à G_α. Lorsqu'un noeud est ajouté à une valeur $\alpha = \alpha_0$, une nouvelle composante connexe est créée dans G_{α_0}. On dit que la date de naissance de la composante (ou du noeud) est α_0. Lorsqu'une arête est ajoutée à une valeur $\alpha = \alpha_1$, si elle relie deux composantes connexes distinctes, alors on dit que l'une des deux composantes (la plus jeune par convention) meurt à la date α_1. On regroupe ses informations dans un *diagramme de persistance*.

Definition 5 *Le diagramme de persistance correspondant à une filtration est l'ensemble des couples (α_n, α_m) des dates de naissance α_n et de mort α_m des composantes connexes obtenues en faisant augmenter le paramètre de filtration. Les points sont comptés avec multiplicité (i.e. un même point peut apparaître plusieurs fois). Les éventuelles composantes qui ne meurent pas ont une date de mort infinie. On appelle persistance d'une composante la quantité $\alpha_m - \alpha_n$.*

La figure 2-A montre un exemple de diagramme de persistance (2-A (h)) et la filtration correspondante (2-A (a)-(g)). Lorsque $\alpha = 1$, les composantes connexes des noeuds 0 et 3 tuent respectivement celles des noeuds 4-5 et 1 (qui ont donc une persistance de 0). Les noeuds 6, 7 et 8 naissent et sont immédiatement tués pas les composantes de leur plus proche voisin. La composante du noeud 3 est tuée par celle du noeud 0 à $\alpha = 15$.

[1] Par la suite, le terme "filtration" désignera toujours des filtrations croissantes

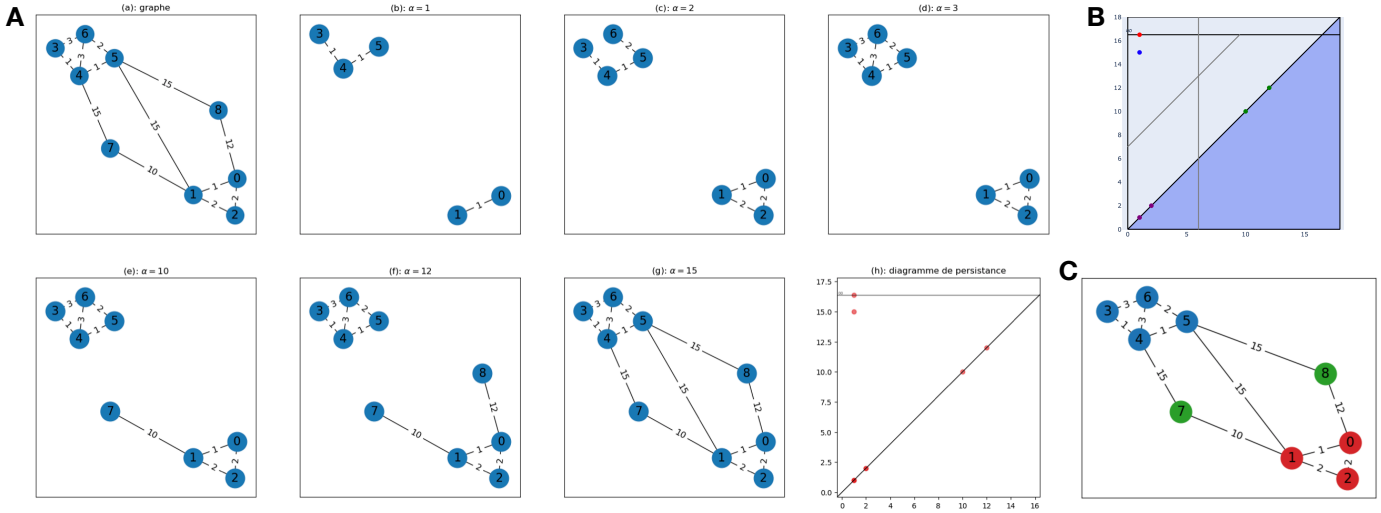

FIGURE 2 : **(A)** Exemple d'une filtration sur graphe et de son diagramme de persistance. **(B)** Diagramme de persistance avec un seuil sur les dates de naissance et sur la persistance. La ligne grise verticale représente le seuil de date de naissance (6). L'autre ligne grise représente le seuil de persistance (7). La persistance d'un point est proportionnelle à sa distance à la diagonale. **(C)** Résultats du clustering. En bleu et rouge : les clusters. En vert : les noeuds isolés.

Les composantes les plus persistantes sont généralement considérées comme importantes. C'est l'idée derrière notre algorithme de clustering : les composantes les plus persistantes sont les clusters, et les noeuds nés les plus tard sont les noeuds isolés. L'algorithme est décrit en détail ci-dessous.

- **Entrée :** un graphe $G = (V, E, d)$.
- Construire la filtration $(NNVR(G, \alpha))_{\alpha \in \mathbb{R}_+}$ et son diagramme de persistance.
- Choisir un seuil minimal de persistance P et un seuil maximal de date de naissance N.
- Placer les noeuds nés après N dans l'ensemble des noeuds isolés. Soit G' le sous-graphe de G obtenu en excluant les noeuds isolés et les arêtes dont ils sont une extrémité.
- Construire la filtration $(NNVR(G', \alpha))_{\alpha \in \mathbb{R}_+}$, sans ajouter les arêtes qui feraient mourir une composante ayant une persistance supérieure à P.
- Les composantes connexes résultantes sont les clusters.
- **Sortie :** une partition de V en différents clusters et un ensemble de noeuds isolés.

La figure 2-B illustre le choix de seuil sur l'exemple de la figure 2-A. Les composantes nées à $\alpha = 10$ et $\alpha = 12$ sont considérées isolées (points verts, noeuds 7 et 8), et deux points persistants sont identifiés (points bleu et rouge), il y aura donc deux clusters. La figure 2-C montre le résultat de l'algorithme sur le graphe de la figure 2-A. Lors de la seconde filtration, on a exclu les noeuds 7 et 8, et on n'a pas ajouté l'arête (1,5) car elle aurait tué une composante née à $\alpha = 1$, qui avait donc une persistance de 14, alors que le seuil est de 7.

2.3 Reconstruction des motifs

Cette partie décrit comment résoudre le problème initial (retrouver les zones d'occurrence des motifs et savoir quelles occurrences correspondent au même motif) après avoir transformé le signal en graphe et obtenu les clusters.

L'idée est de considérer les plages d'indices recouvertes par les sous-séquences de chaque cluster, puis de les regrouper de proche en proche en considérant que deux sous-séquences qui s'intersectent ou qui sont consécutives correspondent à la même occurence du motif.

Formellement : soient $M_1, ..., M_k$ les k clusters obtenus. Chacun est un ensemble de noeuds et donc de sous-séquences de taille w du signal $\mathbf{s}$. Il y a k clusters, chacun correspond à un motif. Un cluster M_i est un ensemble de sous-séquences de $\mathbf{s} \in \mathbb{R}^L$, leur union couvre donc une partie de $\mathbf{s}$ dont les indices sont un ensemble $Z_i \subset [1, L]$. Les zones d'occurrence du motif M_i sont les ensembles maximaux d'indices consécutifs de Z_i. On a donc a la fois les zones d'occurence et le regroupement par motif.

3 Expériences

Nous comparons les performances de notre méthode avec deux algorithmes concurrents sur des signaux synthétiques. La comparaison se fait selon deux critères :

1. **Localisation** : Capacité à retrouver toutes les occurrences. Les zones d'occurrences prédites sont comparées aux vrais zones indépendemment du clustering.
2. **Regroupement** : Capacité à regrouper les occurrences par motifs. Les zones d'occurrences prédites sont comparées aux vrais zones en tenant compte du clustering.

Les performances sont évaluées en termes de précision, rappel et f1-score. Ces métriques sont calculées pour chacun des signaux et nous prenons le score moyen sur l'ensemble des signaux. Le calcul de ces métriques est décrit dans le papier [8] et nous prenons un seuil d'intersection sur union de 40%.

Nous avons créé une base de 50 signaux synthétiques. Chaque signal est composé de 2 motifs dont la taille varie aléatoirement de 200 à 250 points, le nombre d'occurrences varie de 2 à 5 et un intervalle de bruit est laissé entre chaque

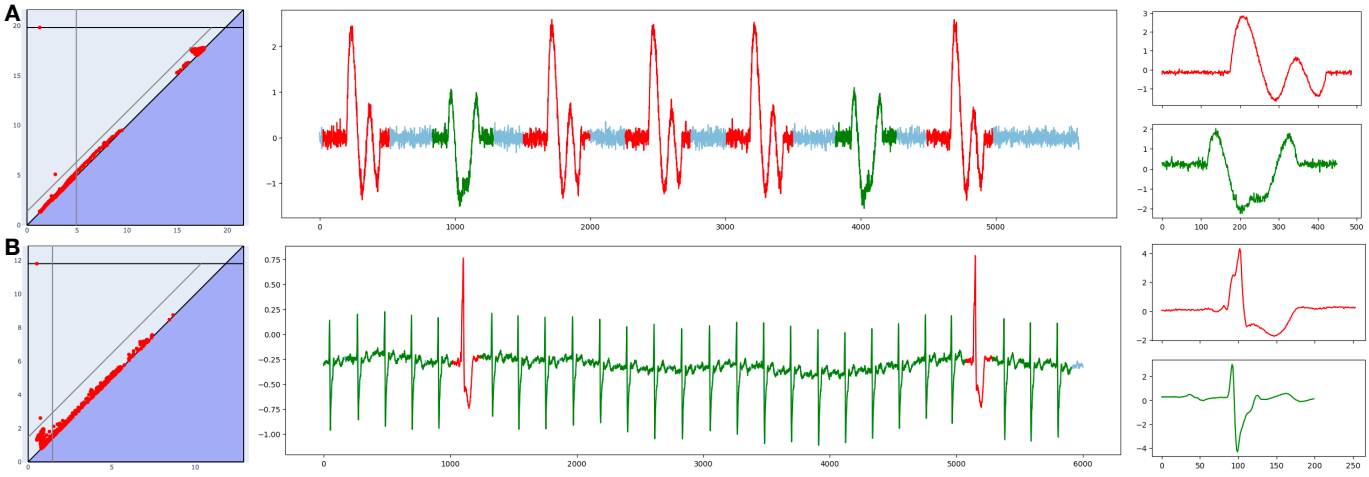

FIGURE 3 : **(A, signal synthétique)** & **(B, extrait d'ECG)**. Dans les deux cas à gauche le diagramme de persistance, au centre une illustration du signal et des occurrences détectées, à droite les motifs appris centrés et réduits.

		Précision	Rappel	F1 score
Localisation	Référence	0.70	0.81	0.73
	Valmod	0.85	0.86	0.84
	PersPa	**0.94**	**0.95**	**0.92**
Regroupement	Référence	0.40	0.69	0.49
	Valmod	0.75	0.81	0.75
	PersPa	**0.87**	**0.94**	**0.88**

TABLE 1 : Résultats de l'expérience sur signaux synthétiques.

occurrence. Chaque motif est créé par interpolation cubic-spline de 7 points consécutifs simulés selon une loi gaussienne centrée réduite et dont le premier et le dernier sont nuls. La figure 3-A est un exemple de signal.

Nous prenons une seule configuration pour notre algorithme et plusieurs pour les algorithmes concurrents. Pour chaque signal et algorithme nous gardons les performances de la meilleure configuration. Les algorithmes et leurs configurations sont les suivants :

- Référence [4]. Les motifs détectés sont tous de la même taille. La taille de fenêtre varie de 200 à 250 et 10 seuils de distance possibles.
- Valmod [3]. Les motifs détectés sont de tailles variables, entre 200 et 250 points, et il y a 10 seuils de distances.
- PersPa, notre méthode. La taille de fenêtre est de 200, nous prenons les 2 points les plus persistants et le seuil sur les distances est obtenu par l'heuristique Otsu [6].

Les résultats de l'expérience sont présentés dans la table 3. Notre méthode montre de meilleures performances selon les deux critères. Les scores de l'algorithme de référence sont les plus faibles, cela suggère que la formulation initial du problème n'est pas appropriée pour le types de signaux étudiés.

A titre qualitatif nous illustrons les résultats de notre algorithme sur un signal synthétique issu de l'expérience précédente, figure 3-A et un extrait de l'ECG d'un patient souffrant d'extrasystole ventriculaire, figure 3-B. Dans les deux cas, le diagramme de persistance suggère le bon nombre de motifs, l'algorithme se concentre sur les occurrences et elles sont bien regroupées par motif. Pour l'ECG, les extrasystoles ventriculaires sont séparées du comportement normal. On remarque que les motifs du signal synthétique commencent et finissent par du bruit, ce sont des effets de bords dus au changement d'amplitude.

4 Conclusion

Nous avons présenté un algorithme de détection non-supervisée de motifs sans connaissance a priori du nombre de motifs. Il représente un signal par un graphe dont sont extraits, selon un critère de persistance, des sous-graphes qui correspondent aux motifs et desquels les occurrences sont déduites. Les expériences menées montrent de bonnes performances par rapport à des méthodes existantes, malgré des effets de bord dus aux changements d'amplitude au début et à la fin des motifs.

Références

[1] J. BOISSONNAT, F. CHAZAL et M. YVINEC : Geometric and topological inference, volume 57. Cambridge University Press, 2018.

[2] E. Keoghand J. LIN : Clustering of time-series subsequences is meaningless : implications for previous and future research. Knowledge and information systems, 8:154–177, 2005.

[3] M. LINARDI, Y. ZHU, T. PALPANAS et E. KEOGH : Matrix profile x : Valmod-scalable discovery of variable-length motifs in data series. In Proceedings of the 2018 International Conference on Management of Data, pages 1053–1066, 2018.

[4] J. LONARDI et P. PATEL : Finding motifs in time series. In Proc. of the 2nd Workshop on Temporal Data Mining, pages 53–68, 2002.

[5] A. MUEEN, E. KEOGH, Q. ZHU, S. CASH et B. WESTOVER : Exact discovery of time series motifs. In Proceedings of the 2009 SIAM international conference on data mining, pages 473–484. SIAM, 2009.

[6] N. OTSU : A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics, 9(1):62–66, 1979.

[7] D. De PAEPE, D. AVENDANO et S. Van HOECKE : Implications of z-normalization in the matrix profile. In Pattern Recognition Applications and Methods : 8th International Conference, ICPRAM 2019, Prague, Czech Republic, February 19-21, 2019, Revised Selected Papers, pages 95–118. Springer, 2020.

[8] N. TATBUL, T. LEE, S. ZDONIK, M. ALAM et J. GOTTSCHLICH : Precision and recall for time series. Advances in neural information processing systems, 31, 2018.

[9] S. TORKAMANI et V. LOHWEG : Survey on time series motif discovery. Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery, 7(2):e1199, 2017.

[10] Y. ZHU, Z. ZIMMERMAN, S. SENOBARI, M. YEH, G. FUNNING, A. MUEEN, P. BRISK et E. KEOGH : 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), pages 739–748. IEEE, 2016.